Parameterized Complexity of Logic-Based Argumentation in Schaefer's Framework

Yasir Mahmood,[1] Arne Meier,[1] Johannes Schmidt [2]

[1]Leibniz Universität Hannover, Institut für Theoretische Informatik, Germany
[2]Jönköping University, Department of Computer Science and Informatics, School of Engineering, Sweden
{mahmood,meier}@thi.uni-hannover.de, johannes.schmidt@ju.se

Abstract

Logic-based argumentation is a well-established formalism modeling nonmonotonic reasoning. It has been playing a major role in AI for decades, now. Informally, a set of formulas is the support for a given claim if it is consistent, subset-minimal, and implies the claim. In such a case, the pair of the support and the claim together is called an argument. In this paper, we study the propositional variants of the following three computational tasks studied in argumentation: ARG (exists a support for a given claim with respect to a given set of formulas), ARG-Check (is a given set a support for a given claim), and ARG-Rel (similarly as ARG plus requiring an additionally given formula to be contained in the support). ARG-Check is complete for the complexity class DP, and the other two problems are known to be complete for the second level of the polynomial hierarchy and, accordingly, are highly intractable. Analyzing the reason for this intractability, we perform a two-dimensional classification: first, we consider all possible propositional fragments of the problem within Schaefer's framework, and then study different parameterizations for each of the fragment. We identify a list of reasonable structural parameters (size of the claim, support, knowledge-base) that are connected to the aforementioned decision problems. Eventually, we thoroughly draw a fine border of parameterized intractability for each of the problems showing where the problems are fixed-parameter tractable and when this exactly stops. Surprisingly, several cases are of very high intractability (paraNP and beyond).

Introduction

Argumentation is a nonmonotonic formalism in artificial intelligence around which an active research community has evolved (Atkinson et al. 2017; Amgoud and Prade 2009; Rago, Cocarascu, and Toni 2018; Baroni et al. 2018). Essentially, there exist two branches of argumentation: the *abstract* (Dung 1995) and the *logic-based* (Besnard and Hunter 2001, 2008; Chesñevar, Maguitman, and Loui 2000; Prakken and Vreeswijk 2002) approach. The abstract setting mainly focusses on formalizing the argumentative structure in a graph-theoretic way. Arguments are nodes in a directed graph and the 'attack-relation' draws which argument eliminates which other. In the logic-based method, one looks for inclusion-minimal consistent sets of formulas Φ (the *support*) that entail a *claim* α, modelled through a formula (in the positive case one calls (Φ, α) an *argument*). In this paper, we focus on the latter formalism and, specifically, study three decision problems. The first, ARG, asks, given a set of formulas Δ (the *knowledge-base*) and a formula α, whether there exists a subset $\Phi \subseteq \Delta$ such that (Φ, α) is an argument in Δ. The two further problems of interest are $\mathrm{ARG\text{-}Check}$ (is a given set a support for a given claim), and $\mathrm{ARG\text{-}Rel}$ (ARG plus requiring an additionally given formula to be contained in the support, too).

Example 1 (Besnard and Hunter (2001))**.** *Consider the following two arguments. (A1) Support: Donald is a public person, so we can publicize details about his private life. Claim: We can publicize that Donald plays golf. (A2) Support: Donald just resigned from politics; as a result, he is no longer a public person. Claim: Donald is no longer a public person.*

Formalizing these arguments would yield $A_1 : \Phi_1 = \{x_{pd} \to x_{dg}, x_{pd}\}, \alpha_1 = \{x_{dg}\}$, $A_2 : \Phi_2 = \{x_{rd} \to \neg x_{pd}, x_{rd}\}, \alpha_2 = \{\neg x_{pd}\}$, *where* $x_{pd} \triangleq$ *"Donald is a public person",* $x_{dg} \triangleq$*"Donald plays golf",* $x_{rd} \triangleq$*"Donald resigned from politics". Each argument is supporting its claim, yet together they are conflicting, as* A_2 *attacks* A_1*.*

It is rather computationally involved to compute the support of an argument, as ARG was shown to be $\mathbf{\Sigma_2^P}$-complete by Parsons et al. (2003). Yet, there have been made efforts to improve the understanding of this high intractability by Creignou et al. (2014; 2011) in two settings: Schaefer's (1978) as well as Post's (1941) framework. Clearly, such research aims for drawing the fine intractability frontier of computationally involved problems to show for what restrictions there still is hope to reach algorithms running for practical applications. Both approaches mainly focus on restrictions on the logical part of the problem language, that is, restricting the allowed connectives or available constraints.

In this paper, Schaefer's approach is our focus, that is, the formulas we study are propositional formulas in conjunctive normalform (CNF) whose clauses are formed depending on a fixed set of relations Γ (the so-called *constraint language*, short CL). In this setting, Schaefer's framework (1978) captures well-known classes of CNF-formulas (e.g., Horn, dual-Horn, or Krom). Accordingly, one can see classifications in such a setting as a one-dimensional approach (the dimension is given rise by the considered logical fragments).

We consider a second dimension on the problem in this paper, namely, by investigating its parameterized complexity (Downey and Fellows 2013). Motivated by the claim that the input length is not the only important structural aspect of problems, one studies so-called parameterizations (or parameters) of a problem. The goal of such studies is to identify a parameter that is relevant for practice but also is slowly growing or even of constant value. If, additionally, one is able to construct an algorithm that solves the problem in time $f(k) \cdot |x|^{O(1)}$ for some computable function f and all inputs (x, k), then one calls the problem *fixed-parameter tractable*. That is why in this case one can solve the problem (for fixed parameter values) in polynomial time. As a result, this complexity class is seen to capture the idea of efficiency in the parameterized sense. While $\mathbf{NP}$-complete problems are considered intractable in the classical setting, on the parameterized level, the complexity class $\mathbf{W}[1]$ is seen to play this counterpart. Informally, this class is characterized via a special kind of satisfiability questions. Above this class an infinite $\mathbf{W}$-hierarchy is defined which culminates in the class $\mathbf{W}[\mathbf{P}]$, which in turn is contained in the class $\mathbf{para\text{-}NP}$ (problems solvable by NTMs in time $f(k) \cdot |x|^{O(1)}$).

Contributions. Our main contributions are the following.

1. We initiate a thorough study of the parameterized complexity of logic-based argumentation. We study three parameters: size of the support, of the claim, and of the knowledge-base. We show that the complexity of ARG, regarding the claim as a parameter, varies: $\mathbf{FPT}$, $\mathbf{W}[1]$, $\mathbf{W}[2]$, $\mathbf{para\text{-}NP}$-, $\mathbf{para\text{-}coNP}$-, as well as $\mathbf{para\text{-}\Sigma_2^P}$-complete cases. For the same parameter, ARG-Check is $\mathbf{FPT}$ for Schaefer, $\mathbf{para\text{-}DP}$-complete otherwise. ARG-Rel is $\mathbf{FPT}$, $\mathbf{para\text{-}NP}$-, or $\mathbf{para\text{-}\Sigma_2^P}$-complete.

   The size of the knowledge-base as the parameter yields dichotomy results for the two problems ARG and ARG-Rel: $\mathbf{FPT}$ versus membership in $\mathbf{para\text{-}coNP}$ and a lower bound that relates to the implication problem.

   Concerning the size of the support as the parameter, we prove a dichotomy: $\mathbf{FPT}$ versus $\mathbf{para\text{-}DP}$-membership and the same hardness as the implication problem.

2. As a byproduct, we advance the algebraic tools in the context of Schaefer's framework, and show a list of technical implementation results that are independent of the studied problem and might be beneficial for further research in the constraint context.

3. We classify the parameterized complexity of the implication problem (does a set of propositional formulas Φ imply a propositional formula α?) with respect to the parameter $|\alpha|$ and show that it is $\mathbf{FPT}$ if the CL is Schaefer, and $\mathbf{para\text{-}coNP}$-complete otherwise.

Related Work. Very recently, Mahmood et al. (2020) presented a parameterized classification of abductive reasoning in Schaefer's framework. Some of their cases, as well as results from Nordh and Zanuttini (2008) relate to some of our results. The studies of the implication problem in the frameworks of Schaefer (Schnoor and Schnoor 2008) as well as in the one in Post (Beyersdorff et al. 2009) prove a classical complexity landscape. Last year, Hecher et al. (2019) conducted a parameterized study of abstract argumentation. The known classical results (Creignou, Egly, and Schmidt 2014; Nordh and Zanuttini 2008; Schnoor and Schnoor 2008; Beyersdorff et al. 2009) are partially used in some of our proofs, e.g., showing some parameterized complexity lower bounds. The two mentioned parameterized complexity related papers (Fichte, Hecher, and Meier 2019; Mahmood, Meier, and Schmidt 2020) both are about different formalisms that are slightly related to our setting (the first is about abstract argumentation, the second on abduction).

Due to space limitations, for results marked with a $\star$, the proof can be found in the technical report of the paper (Mahmood, Meier, and Schmidt 2021).

Preliminaries

We assume familiarity with basic notions in complexity theory (cf. Sipser (1997)) and use the complexity classes $\mathbf{P}, \mathbf{NP}, \mathbf{coNP}, \mathbf{\Sigma_2^P}$. For a set S, we write $|S|$ for its *cardinality*. Abusing notation, we will use $|w|$, for a string w, to denote its *length*. If φ is a formula, then $\mathrm{Vars}(\varphi)$ denotes its set of variables, and $\mathrm{enc}(\varphi)$ its *encoding*. W.l.o.g., we assume a reasonable encoding computable in polynomial time that encodes variables in binary. The *weight* of an assignment σ is the number of variables mapped to 1.

Parameterized Complexity. We give a brief introduction to parameterized complexity theory. A more detailed exposition can be found in the textbook of Downey and Fellows (2013). A *parameterized problem (PP)* Π is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is an alphabet. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*. If there exists a deterministic algorithm deciding Π in time $f(k) \cdot |x|^{O(1)}$ for every input (x, k), where f is a computable function, then Π is *fixed-parameter tractable* (short: $\mathbf{FPT}$).

Definition 2. *Let Σ and Δ be two alphabets. A PP $\Pi \subseteq \Sigma^* \times \mathbb{N}$* fpt-reduces *to a PP $\Theta \subseteq \Delta^* \times \mathbb{N}$, in symbols $\Pi \leq^{\mathbf{FPT}} \Theta$, if the following is true: (i) there is an $\mathbf{FPT}$-computable function f, such that, for all $(x, k) \in \Sigma^* \times \mathbb{N}$: $(x, k) \in \Pi \Leftrightarrow f(x, k) \in \Theta$, (ii) there exists a computable function $g\colon \mathbb{N} \to \mathbb{N}$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ and $f(x, k) = (y, \ell)$: $\ell \leq g(k)$.*

The problems Π and Θ are $\mathbf{FPT}$-equivalent if both $\Pi \leq^{\mathbf{FPT}} \Theta$ and $\Theta \leq^{\mathbf{FPT}} \Pi$ is true. We also use higher classes via the concept of *precomputation on the parameter*.

Definition 3. *Let $\mathcal{C}$ be any complexity class. Then $\mathbf{para}\text{-}\mathcal{C}$ is the class of all PPs $\Pi \subseteq \Sigma^* \times \mathbb{N}$ such that there exists a computable function $\pi\colon \mathbb{N} \to \Delta^*$ and a language $L \in \mathcal{C}$ with $L \subseteq \Sigma^* \times \Delta^*$ such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ we have that $(x, k) \in \Pi \Leftrightarrow (x, \pi(k)) \in L$.*

Observe that $\mathbf{para\text{-}P} = \mathbf{FPT}$ is true. For a constant $c \in \mathbb{N}$ and a PP $\Pi \subseteq \Sigma^* \times \mathbb{N}$, the *c-slice of* Π, written as Π_c, is defined as $\Pi_c := \{\, (x, k) \in \Sigma^* \times \mathbb{N} \mid k = c \,\}$. Observe that, in our setting, showing $\Pi \in \mathbf{para}\text{-}\mathcal{C}$, it suffices to show

$\Pi_c \in \mathcal{C}$ for every $c \in \mathbb{N}$. Consider the following special subclasses of formulas:

$$\Gamma_{0,d} = \{\, \ell_1 \wedge \ldots \wedge \ell_c \mid \ell_1, \ldots, \ell_c \text{ are literals and } c \leq d \,\},$$
$$\Delta_{0,d} = \{\, \ell_1 \vee \ldots \vee \ell_c \mid \ell_1, \ldots, \ell_c \text{ are literals and } c \leq d \,\},$$
$$\Gamma_{t,d} = \left\{ \bigwedge_{i \in I} \alpha_i \,\middle|\, \alpha_i \in \Delta_{t-1,d} \text{ for } i \in I \right\},$$
$$\Delta_{t,d} = \left\{ \bigvee_{i \in I} \alpha_i \,\middle|\, \alpha_i \in \Gamma_{t-1,d},\ i \in I \right\}.$$

The parameterized weighted satisfiability problem (p-WSAT) for propositional formulas is defined as below. The problem p-WSAT$(\Gamma_{t,d})$ asks, given a $\Gamma_{t,d}$-formula α with $t, d \geq 1$ and $k \in \mathbb{N}$, parameterized by k, is there a satisfying assignment for α of weight k?

The classes of the $\mathbf{W}$-hierarchy can be defined in terms of these problems.

Proposition 4 (Downey and Fellows, 2013)**.** *The problem* p-WSAT$(\Gamma_{t,d})$ *is* $\mathbf{W}[t]$*-complete for each* $t \geq 1$ *and* $d \geq 1$*, under* $\leq^{\mathbf{FPT}}$*-reductions.*

Logic-based Argumentation. All formulas in this paper are propositional formulas. We follow the notion of Creignou et al. (2014).

Definition 5 (Besnard and Hunter (2001))**.** *Given a set of formulas* Φ *and a formula* α*, one says that* (Φ, α) *is an* argument (for α) *if (1)* Φ *is consistent, (2)* $\Phi \models \alpha$*, and (3)* Φ *is subset-minimal w.r.t. (2). In case of* $\Phi \subseteq \Delta$*,* (Φ, α) *is an* argument in Δ*. We call* α *the* claim*,* Φ *the* support *of the argument, and* Δ *the* knowledge-base.

In this paper, we consider three problems from the area of logic-based argumentation, namely ARG, ARG-Check, and ARG-Rel. The problem ARG asks, given a set of formulas Δ and a formula α, is there a set $\Phi \subseteq \Delta$ such that (Φ, α) is an argument in Δ? The problem ARG-Check asks, given a set of formulas Φ and a formula α, is (Φ, α) an argument? The problem ARG-Rel asks, given a set of formulas Δ, and formulas $\psi \in \Delta$ and α, is there a set $\Phi \subseteq \Delta$ with $\psi \in \Phi$ such that (Φ, α) is an argument in Δ?

Turning to the parameterized complexity perspective on the introduced problems, immediate parameters that we consider are $|\mathrm{enc}(\mathcal{X})|$ (size of the encoding of $\mathcal{X}$), $|\mathcal{X}|$ (number of formulas in $\mathcal{X}$), $|\mathrm{Vars}(\mathcal{X})|$ (number of variables in $\mathcal{X}$) for $\mathcal{X} \in \{\Delta, \Phi\}$, as well as $|\mathrm{enc}(\alpha)|$ and $|\mathrm{Vars}(\alpha)|$. Regarding the parameterized versions of the problems from above, e.g., p-ARG(k), where k is a parameter, then defines the version of ARG parameterized by k, accordingly.

In the following, we want to formally relate the mentioned notions of encoding length, number of variables, as well as number of formulas. We will see that bounding the encoding length, implies having limited space for encoding variables and, in turn, restricts the number of possible formulas. However, the converse is also true: if one bounds the number of variables, then one also has limited possibilities about defining *different* formulas. The following definition makes clear what 'different' means in our context.

Definition 6 (Formula redundancy)**.** *A CNF-formula* $\varphi = \bigwedge_{i=1}^{m} C_i$*, with* $C_i = (\ell_{i,1} \vee \cdots \vee \ell_{i,n_i})$ *is* redundant *if there exist* $1 \leq i \neq j \leq m$ *such that* $\{\, \ell_{i,k} \mid 1 \leq k \leq n_i \,\} = \{\, \ell_{j,k} \mid 1 \leq k \leq n_j \,\}$.

Example 7. *The formulas* $x \wedge x$ *and* $(x \vee x \vee y) \wedge (x \vee y)$ *are redundant. The formulas* $x \wedge y$ *and* $(x \vee y) \wedge x$ *are not redundant.*

Liberatore (2005) studied a stronger notion of redundancy in the context of CNF-formulas, namely, on the level of implied clauses. We do not need such a strict notion of redundancy here, as the weaker notion suffices for proving the following Lemma. As a result, in the following, we consider only formulas that are just not redundant. The redundancy (in our context) can be straightforwardly checked in time quadratic in the length of the given formula.

Lemma 8 ($\star$)**.** *For any set of CNF-formulas* Φ*, we have that*

1. $|\Phi| \leq 2^{2^{2 \cdot |\mathrm{Vars}(\Phi)|}}$,
2. $f(|\mathrm{Vars}(\Phi)|) \leq |\mathrm{enc}(\Phi)|$*, where* f *is some computable function, and*
3. $|\mathrm{enc}(\Phi)| \leq |\Phi|^3$.

Notice that due to Lemma 8, the problems ARG, ARG-Check, ARG-Rel parameterized with respect to any of the parameters for the respective three (two) variants introduced above are $\mathbf{FPT}$-equivalent. As a result, we will choose the one of the three (two) variants in our results that is technically most convenient. Notice also that the parameter $|\Phi|$ only makes sense for ARG-Check, whereas $|\Delta|$ makes sense only for the other two problems, that is, ARG and ARG-Rel.

Schaefer's Framework

For a deeper introduction into Schaefer's CSP framework, consider the article of Böhler et al. (2004).

A *logical relation* of arity $k \in \mathbb{N}$ is a relation $R \subseteq \{0,1\}^k$, and a *constraint* C is a formula $C = R(x_1, \ldots, x_k)$, where R is a k-ary logical relation, and $x_1, \ldots, x_k$ are (not necessarily distinct) variables. If V is a set of variables and u a variable, then $C[V/u]$ denotes the constraint obtained from C by replacing every occurrence of every variable of V by u. An assignment θ *satisfies* C, if $(\theta(x_1), \ldots, \theta(x_k)) \in R$. A *constraint language* (CL) Γ is a finite set of logical relations, and a Γ*-formula* is a conjunction of constraints over elements from Γ. Eventually, a Γ-formula φ is *satisfied* by an assignment θ, if θ simultaneously satisfies all constraints in it. In such a case θ is also called a *model of* φ. Whenever a Γ-formula or a constraint is logically equivalent to a single clause or term or literal, we treat it as such. We say that a k-ary relation R is represented by a formula ϕ in CNF if ϕ is a formula over k distinct variables $x_1, \ldots, x_k$ and $\phi \equiv R(x_1, \ldots, x_k)$. Moreover, we say that R is

- *Horn* (resp., *dual-Horn*) if ϕ contains at most one positive (negative) literal per each clause.
- *Bijunctive* if ϕ contains at most two literals per each clause.
- *Affine* if ϕ is a conjunction of linear equations of the form $x_1 \oplus \ldots \oplus x_n = a$ where $a \in \{0,1\}$.

- *Essentially negative* if every clause in ϕ is either negative or unit positive. R is *essentially positive* if every clause in ϕ is either positive or unit negative.
- 1-*valid* (resp., 0-*valid*) if every clause in ϕ contains at least one positive (negative) literal.

Furthermore, we say a relation is *Schaefer* if it is Horn, dual-Horn, bijunctive, or affine. We say that a relation is ε-*valid* if it is 1- or 0-valid or both. Finally, for a property *P* of a relation, we say that a CL Γ is *P* if all relations in Γ are *P*.

Definition 9. *1. The set $\langle\Gamma\rangle$ is the smallest set of relations that contains Γ, the equality constraint, $=$, and which is closed under primitive positive first order definitions, that is, if ϕ is an $\Gamma \cup \{=\}$-formula and $R(x_1, \ldots, x_n) \equiv \exists y_1 \ldots \exists y_l \phi(x_1, \ldots, x_n, y_1, \ldots, y_l)$, then $R \in \langle\Gamma\rangle$. In other words, $\langle\Gamma\rangle$ is the set of relations that can be expressed as a $\Gamma \cup \{=\}$-formula with existentially quantified variables.*

2. The set $\langle\Gamma\rangle_{\neq}$ is the set of relations that can be expressed as a Γ-formula with existentially quantified variables (no equality relation is allowed).

3. The set $\langle\Gamma\rangle_{\nexists,\neq}$ is the set of relations that can be expressed as a Γ-formula (neither the equality relation nor existentially quantified variables are allowed).

The set $\langle\Gamma\rangle$ is called a *relational clone* or a *co-clone* with *base* Γ (Böhler et al. 2005). Notice that for a co-clone C and a CL Γ the statements $\Gamma \subseteq \mathsf{C}$, $\langle\Gamma\rangle \subseteq \mathsf{C}$, $\langle\Gamma\rangle_{\neq} \subseteq \mathsf{C}$ and $\langle\Gamma\rangle_{\nexists,\neq} \subseteq \mathsf{C}$ are equivalent. Throughout the paper, we refer to different types of Boolean relations and corresponding co-clones following Schaefer's terminology (Schaefer 1978). For a tabular overview of co-clones, relational properties, and bases, we refer the reader to Mahmood, Meier, and Schmidt (2021, Table 1). Note that $\langle\Gamma\rangle_{\neq} \subseteq \langle\Gamma\rangle$ is true by definition. The other direction is not true in general. However, if $(x = y) \in \langle\Gamma\rangle_{\neq}$, then we have that $\langle\Gamma\rangle_{\neq} = \langle\Gamma\rangle$.

Example 10. *Let $R(x_1, x_2, x_3) := (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3)$. Then $(x_1 \lor x_2) \land (x_2 \oplus x_3 = 0) \equiv \exists y (R(x_1, x_2, y) \land F(y) \land (x_2 = x_3))$, where $F = \{0\}$. This implies that $(x_1 \lor x_2) \land (x_2 \oplus x_3 = 0) \in \langle\{R, F\}\rangle$.*

Technical Implementation Results

We say a Boolean relation R is *strictly essentially positive* (resp., *strictly essentially negative*) if it can be defined by a conjunction of literals and positive clauses (resp., negative clauses) only. Note that the only difference to essentially positive (resp., essentially negative) is the absence of the equality relation (see (Mahmood, Meier, and Schmidt 2021, Table 1)). We abbreviate in the following essentially positive by "ess.pos." and essentially negative by "ess.neg.".

Proposition 11 (Mahmood et al. (2020, Lem. 7))**.** *Let Γ be a CL that is neither ess.pos., nor ess.neg. Then, we have that $(x = y) \in \langle\Gamma\rangle_{\neq}$ and $\langle\Gamma\rangle = \langle\Gamma\rangle_{\neq}$.*

With the following implementation result we can strengthen this statement to Lemma 13.

Lemma 12. *Let Γ be a CL that is not ε-valid. If Γ is ess.neg. and not strictly ess.neg. or ess.pos. and not strictly ess.pos. then we have that $(x = y) \land t \land \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$.*

Proof. We prove the statement for Γ that is ess.pos. but not strictly ess.pos. The other case can be treated analogously. W.l.o.g., let $\Gamma = \{R\}$, thus R is ess.pos. but not strictly ess.pos. Furthermore, R is neither 1-valid nor 0-valid. Let R be of arity k and let $V = \{x_1, \ldots, x_k\}$ be a set of k distinct variables. By definition of ess.pos. (cf. IS_{02} in (Mahmood, Meier, and Schmidt 2021, Table 1)), R can be written as conjunction of negative literals, positive clauses and equalities.

If R can be written without any equality, then R is strictly ess.pos., a contradiction. As a result, any representation of R as conjunction of negative literals, positive clauses and equalities requires at least one equality. Suppose, w.l.o.g., that $R(x_1, \ldots, x_k) \models (x_1 = x_2)$, while $R(x_1, \ldots, x_k) \not\models x_1$ and $R(x_1, \ldots, x_k) \not\models \neg x_1$. We define the following three subsets of V: $W = \{ x_i \mid R(x_1, \ldots, x_k) \models (x_1 = x_i) \}$, $N = \{ x_i \mid R(x_1, \ldots, x_k) \models \neg x_i \}$, and $P = V \setminus (W \cup N)$

By construction the three sets provide a partition of V. Then, W is nonempty by construction, N is nonempty since R is not 1-valid and P is nonempty since R is not 0-valid. Denote by C the $\{R\}$-constraint $C = R(x_1, \ldots, x_k)$. Consider the constraint $M(x_1, x_2, t, f) = C[W/x_2, P/t, N/f]$. One verifies that $M(x_1, x_2, t, f) \equiv (x_1 = x_2) \land t \land \neg f$. ☐

Lemma 13. *Let Γ be a CL that is neither strictly ess.pos., nor strictly ess.neg. Then $(x = y) \in \langle\Gamma\rangle_{\neq}$ and $\langle\Gamma\rangle = \langle\Gamma\rangle_{\neq}$.*

Proof. If Γ is not ess.pos. and not ess.neg. the statement follows from Proposition 11. Note that this lemma's statement implies that Γ is not ε-valid. If Γ is ess.pos. or ess.neg., by Lemma 12 we have $(x = y) \land t \land \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$. Conclude by noticing that $(x = y) \equiv \exists t \exists f\, (x = y) \land t \land \neg f \in \langle\Gamma\rangle_{\neq}$. ☐

Lemma 14. *Let Γ be a CL that is neither ε-valid, nor ess.pos., nor ess.neg. Then, if Γ is*

1. *not Horn, not dual-Horn, and not complementive, then $(x \neq y) \land t \land \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$,*
2. *not Horn, not dual-Horn, and complementive, then $(x \neq y) \in \langle\Gamma\rangle_{\nexists,\neq}$, and*
3. *Horn or dual-Horn, then $(x = y) \land t \land \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$.*

Proof. This follows immediately from the proof of Proposition 11. The proof given in (Mahmood, Meier, and Schmidt 2020, Lemma 7) makes a case distinction according to whether Γ is 0-valid and/or 1-valid. In the case of non ε-valid Γ a further case distinction is made according to whether Γ is Horn and/or dualHorn. Here the statements 1., 2., and 3. are proven. ☐

Let us denote by T/F the unary relations that implement true/false. That is, $\mathrm{T} = \{(1)\}$ and $\mathrm{F} = \{(0)\}$. The following implementation results are folklore.

Proposition 15 (Creignou et al. (2001))**.** *If Γ is a CL that is*

1. *complementive and not ε-valid, then $(x \neq y) \in \langle\Gamma\rangle_{\nexists,\neq}$,*
2. *neither complementive, nor ε-valid, then $(t \land \bar{f}) \in \langle\Gamma\rangle_{\nexists,\neq}$.*
3. *1-valid and not 0-valid, then $\mathrm{T} \in \langle\Gamma\rangle_{\nexists,\neq}$,*
4. *0-valid and not 1-valid, then $\mathrm{F} \in \langle\Gamma\rangle_{\nexists,\neq}$, and*
5. *0-valid and 1-valid, then $(x = y) \in \langle\Gamma\rangle_{\nexists,\neq}$.*

Parameterized Implication Problem

In this subsection, we consider the parameterized complexity of the implication problem (IMP). The problem $\mathrm{IMP}(\Gamma)$ asks, given a set of Γ-formulas Φ and a Γ-formula α, is $\Phi \models \alpha$ true? For p-IMP, the parameterized version of IMP, we consider the parameter $k \in \{|\Phi|, |\alpha|\}$, and also write $\text{p-IMP}(\Gamma, k)$. The following corollary is due to Schnoor and Schnoor (2008, Theorem 6.5). They study a restriction of our problem IMP, where $|\Phi| = 1$.

Corollary 16. *Let* Γ *be a CL.* $\mathrm{IMP}(\Gamma)$ *is in* $\mathbf{P}$ *when* Γ *is Schaefer and* $\mathbf{coNP}$*-complete otherwise.*

Consequently, the parameterized problem $\text{p-IMP}(\Gamma, k)$ is $\mathbf{FPT}$ when Γ is Schaefer and $k \in \{|\Phi|, |\alpha|\}$. We consider the cases when Γ is not Schaefer. In the following, we differentiate the restrictions on Φ from the ones on α. That is, we introduce a technical variant, $\mathrm{IMP}(\Gamma', \Gamma)$ of the implication problem. An instance of $\mathrm{IMP}(\Gamma', \Gamma)$ is a tuple (Φ, α), where Φ is a set of Γ'-formulas and α is a Γ-formula. The following corollary also follows from the work of Schnoor and Schnoor (2008, Theorem 6.5).

Corollary 17. *Let* Γ *and* Γ' *be non-Schaefer CLs. If* $\Gamma' \subseteq \langle\Gamma\rangle$ *then* $\mathrm{IMP}(\Gamma', \Gamma) \leq_m^{\mathbf{P}} \mathrm{IMP}(\Gamma)$.

Regarding non-Schaefer CLs, it turns out that the parameter α does not make the problem any easier. One possible explanation for this hardness is that the formulas in Φ and α do not necessarily share a set of variables.

Lemma 18. *The problem* $\text{p-IMP}(\Gamma, |\alpha|)$ *is* $\mathbf{para\text{-}coNP}$*-complete when the CL* Γ *is not Schaefer.*

Proof. Membership follows because the classical problem is in $\mathbf{coNP}$. To achieve the lower bound, we reduce from the unsatisfiability problem. That is, given a formula Φ, the question is whether Φ is unsatisfiable. Moreover, checking unsatisfiability is $\mathbf{coNP}$-complete for non-Schaefer languages (follows by Schaefer's (1978) SAT classification).

We will inherently use Corollary 17 and make a case distinction as whether (Φ, α) is 1-valid, 0-valid or complementive.

Case 1. Let Γ be 1-valid and not 0-valid. We prove that for some well chosen 1-valid language Γ' and a Γ'-formula Φ, the problem $\text{p-IMP}(\Gamma', \Gamma, |\alpha|)$ is $\mathbf{para\text{-}coNP}$-hard. According to item (3.) in Proposition 15, $\mathrm{T} \in \langle\Gamma'\rangle_{\nexists,\neq}$. Let $\alpha = \mathrm{T}(x)$ and Φ be a Γ'-formula where x does not appear. Then $\Phi \models \alpha$ if and only if Φ is unsatisfiable. This is because, if Φ is satisfiable then there is an assignment s such that $s \models \psi$. This gives a contradiction because the assignment s' that extends s by $s'(x) = 0$ satisfies that $s' \models \Phi$ and $s' \not\models \alpha$.

Case 2. Let Γ be 0-valid and not 1-valid. According to item (4.) in Proposition 15, $\mathrm{F} \in \langle\Gamma'\rangle_{\nexists,\neq}$. This case is similar to Case 1, as we take $\alpha = \mathrm{F}(x)$ and Φ a Γ'-formula not containing x.

Case 3. Let Γ be complementive but not ε-valid. We prove that for some well chosen complementive language Γ' and a Γ-formula α, the problem $\text{p-IMP}(\Gamma', \Gamma, |\alpha|)$ is $\mathbf{para\text{-}coNP}$-hard. According to item (1.) in Proposition 15, $x \neq y \in \langle\Gamma'\rangle_{\nexists,\neq}$. Then, $\mathbf{coNP}$-hardness follows, as for any set of Γ'-formulas Φ, $\Phi \models (x \neq x)$ if and only if Φ is unsatisfiable.

Case 4. Let Γ be 0- and 1-valid. By Lemma 14 (1.)/(2.), we have access to '$\neq$'. We can state a reduction from the complement of SAT to $\text{p-IMP}(\Gamma, |\alpha|)$ as in Case 3. That is, Φ is unsatisfiable if and only if $\Phi \models x \neq x$ for a fresh variable x. ☐

Note that regarding the parameter $|\Phi|$, the problem $\text{p-IMP}(\Gamma, |\Phi|)$ is $\mathbf{FPT}$ if Γ is Schaefer. Otherwise, only $\mathbf{coNP}$-membership is clear.

Parameter: Size of the Claim α

In this section we discuss the complexity results regarding the parameter α, that is, the number of variables and the encoding size of α. It turns out that the computational complexity of the argumentation problems is hidden in the structure of the underlying CL. That is, in many cases, considering the claim size as a parameter does not lower the complexity. This is proved by noting that certain slices of the parameterized problems already yield hardness results.

Theorem 19. $\text{p-ARG}(\Gamma, |\alpha|)$*, for a CL* Γ*, is*

1. $\mathbf{FPT}$ *if* Γ *is Schaefer and* ε*-valid,*
2. $\mathbf{para\text{-}NP}$*-complete if* Γ *is Schaefer and neither* ε*-valid, nor strictly ess.pos., nor strictly ess.neg.,*
3. *in* $\mathbf{W}[1]$ *if* Γ *is strictly ess.neg. and strictly ess.pos.,*
4. *in* $\mathbf{W}[2]$ *if* Γ *is strictly ess.neg. or strictly ess.pos.,*
5. $\mathbf{para\text{-}coNP}$*-complete if* Γ *is not Schaefer and* ε*-valid, and*
6. $\mathbf{para\text{-}\Sigma_2^P}$*-complete if* Γ *is not Schaefer and not* ε*-valid.*

Proof. (1.) The classical problem $\mathrm{ARG}(\Gamma)$ is already in $\mathbf{P}$ for this case (Creignou, Egly, and Schmidt 2014, Thm 5.3). (2.) The upper bound follows because the unparameterized problem $\mathrm{ARG}(\Gamma)$ is in $\mathbf{NP}$ (Creignou, Egly, and Schmidt 2014, Prop 5.1). The lower bound is proven in Lemmas 21, 22 and 23. (3.) is proven in Lemma 24. (4.) is proven in Lemma 25.

For (5.) (resp., (6.)), the membership follows because the classical problem is in $\mathbf{coNP}$ (resp., $\mathbf{\Sigma_2^P}$) (Creignou, Egly, and Schmidt 2014, Thm 5.3). For hardness of $\text{p-ARG}(\Gamma, |\alpha|)$ when Γ is ε-valid, notice that, since Δ is ε-valid, an instance (Δ, α) of p-ARG admits an argument if and only if $\Delta \models \alpha$. The result follows from Lemma 18 because the implication problem is still $\mathbf{para\text{-}coNP}$-hard. Finally, when Γ is not Schaefer and not ε-valid, in the proofs of Creignou et al. (2014, Prop. 5.2) the constructed reductions define α whose length is 2 or 3. Accordingly, either the 2-slice or the 3-slice is $\mathbf{\Sigma_2^P}$-hard. This gives the desired hardness result. ☐

For technical reasons we introduce the following variant of the argumentation existence problem. The problem $\mathrm{ARG}(\Gamma, R)$ asks, given a set of Γ-formulas Δ and an R-formula α, $\exists\, \Phi \subseteq \Delta$ s.t. (Φ, α) is an argument in Δ?

Lemma 20. *Let Γ, Γ' be two CLs and R a Boolean relation. If $\Gamma' \subseteq \langle\Gamma\rangle_{\neq}$ and $R \in \langle\Gamma\rangle_{\nexists,\neq}$, then* $\mathrm{ARG}(\Gamma', R) \leq_m^{\log} \mathrm{ARG}(\Gamma)$.

Proof. Let (Δ, α) be an instance of the first problem, where $\Delta = \{\, \delta_i \mid i \in I \,\}$ and $\alpha = R(x_1, \ldots, x_k)$. We map this instance to (Δ', α'), where $\Delta' = \{\delta'_i \mid \delta_i \in \Delta\}$ and α' is a Γ-formula equivalent to $R(x_1, \ldots, x_k)$ (which exists because $R \in \langle\Gamma\rangle_{\nexists,\neq}$). For $i \in I$ we obtain δ'_i from δ_i by replacing δ_i by an equivalent Γ-formula with existential quantifiers (such a representation exists since $\Gamma' \subseteq \langle\Gamma\rangle_{\neq}$) and deleting all existential quantifiers. □

Note that the previous result is only used to show lower bounds for specific slices and, accordingly, is stated in the classical setting.

Lemma 21. *If the CL Γ is neither affine, nor ε-valid, nor ess.pos., nor ess.neg., then* $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|)$ *is* **para-NP**-*hard.*

Proof. We give a reduction from the **NP**-complete problem Pos-1-In-3-Sat such that $|\alpha|$ is constant. An instance of Pos-1-In-3-Sat is a 3CNF-formula with only positive literals, the question is to determine whether there is a satisfying assignment which maps exactly one variable in each clause to true. We make a case distinction according to the case (1.) and (3.) in Lemma 14. Case (2.) is not needed because if Γ is not affine, not horn and not dual-Horn, then Γ can not be complementive. We first treat case (3.), that is, we have that $(x = y) \wedge t \wedge \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$. We then show that the other case can be treated with minor modifications of the procedure.

Let φ be an instance of Pos-1-In-3-Sat. We first reduce φ to and instance (Δ, α) of $\mathrm{ARG}(\{\mathrm{T}, \mathrm{F}, =\}, (x = y) \wedge t \wedge \neg f)$, and then conclude with Lemmas 14 and 20. Given $\varphi = \bigwedge_{i=1}^{k}(x_i \vee y_i \vee z_i)$, an instance of Pos-1-In-3-Sat and let $t, f, c_1, \ldots, c_{k+1}$ be fresh variables. We let Δ and α as following.

$$\begin{aligned}
\Delta = &\textstyle\bigcup_{i=1}^{k}\{x_i \wedge \neg y_i \wedge \neg z_i \wedge (c_i = c_{i+1}) \wedge t \wedge \neg f\}\\
\cup &\textstyle\bigcup_{i=1}^{k}\{\neg x_i \wedge y_i \wedge \neg z_i \wedge (c_i = c_{i+1}) \wedge t \wedge \neg f\}\\
\cup &\textstyle\bigcup_{i=1}^{k}\{\neg x_i \wedge \neg y_i \wedge z_i \wedge (c_i = c_{i+1}) \wedge t \wedge \neg f\},\\
\alpha = &\ (c_1 = c_{k+1}) \wedge t \wedge \neg f.
\end{aligned}$$

Note that any formula in Δ is expressible as Γ-formula since $\{\mathrm{T}, \mathrm{F}, =\} \subseteq \mathsf{IM}_2 \subseteq \langle\Gamma\rangle$ (cf. Mahmood, Meier, and Schmidt (2021, Table 1)). Since by Lemma 13, $\langle\Gamma\rangle_{\neq} = \langle\Gamma\rangle$ and by construction $(x = y) \wedge t \wedge \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$, we have, by Lemma 20, the desired reduction to $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|)$. Note that in the reduction of Lemma 20 the size of α is always constant.

For case (1.) of Lemma 14 we have that $(x \neq y) \wedge t \wedge \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$. To cope with this change in the reduction we introduce one additional variable d and replace α by $(c_1 \neq d) \wedge (d \neq c_{k+1}) \wedge t \wedge \neg f$. □

Lemma 22. *If the CL Γ is affine, neither ε-valid, nor ess. pos., nor ess.neg., then* $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|)$ *is* **para-NP**-*hard.*

Proof. We proceed analogously to the proof of Lemma 21. We give a reduction from the **NP**-complete problem Pos-1-In-3-Sat such that $|\alpha|$ is constant. We make a case distinction according to case (1.) and (2.) in Lemma 14 (case 3. can not occur for Γ is affine and not ess.pos.). First, we treat the second case, that is, we have that $(x \neq y) \in \langle\Gamma\rangle_{\nexists,\neq}$. Then, we show that the first case can be treated with minor modifications of the procedure.

Now, we reduce Pos-1-In-3-Sat to $\mathrm{ARG}(\{=, \neq\}, \{\neq\})$, and then conclude with Lemmas 14 and 20. We give the following reduction. Let $\varphi = \bigwedge_{i=1}^{k}(x_i \vee y_i \vee z_i)$ be an instance of Pos-1-In-3-Sat and let $t, d, c_1, \ldots, c_{k+1}$ be fresh variables. We map φ to (Δ, α), where

$$\begin{aligned}
\Delta = &\textstyle\bigcup_{i=1}^{k}\{(x_i = t) \wedge (y_i \neq t) \wedge (z_i \neq t) \wedge (c_i = c_{i+1})\}\\
&\cup \textstyle\bigcup_{i=1}^{k}\{(x_i \neq t) \wedge (y_i = t) \wedge (z_i \neq t) \wedge (c_i = c_{i+1})\}\\
&\cup \textstyle\bigcup_{i=1}^{k}\{(x_i \neq t) \wedge (y_i \neq t) \wedge (z_i = t) \wedge (c_i = c_{i+1})\},\\
\alpha = &\,(c_1 \neq d) \wedge (d \neq c_{k+1}).
\end{aligned}$$

Note that any formula in Δ is expressible as Γ-formula since $\{=, \neq\} \subseteq \mathsf{ID} \subseteq \langle\Gamma\rangle$ (cf. Mahmood, Meier, and Schmidt (2021, Table 1)). Since by Lemma 13 $\langle\Gamma\rangle_{\neq} = \langle\Gamma\rangle$ and by construction $(x \neq y) \in \langle\Gamma\rangle_{\nexists,\neq}$, we have by Lemma 20 the desired reduction to $\mathrm{p\text{-}ARG}(\Gamma)$. Note that in the reduction of Lemma 20 the size of α is always constant.

For case (1.) of Lemma 14 we have that $(x \neq y) \wedge t \wedge \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$. To cope with this change in the reduction, we introduce one additional variable f and add the constraints $t \wedge \neg f$ to α as well as to every formula in Δ . □

Lemma 23. *Let Γ be a CL that is not ε-valid. If Γ is ess.pos. and not strictly ess.pos. or ess.neg. and not strictly ess.neg., then* $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|)$ *is* **para-NP**-*hard.*

Proof. We can use exactly the same reduction as in Lemma 21, except we do not require a case distinction. Note that, by Proposition 15, we have that $(t \wedge \neg f) \in \langle\Gamma\rangle_{\nexists,\neq}$. Since $\exists f\,(t \wedge \neg f) \equiv \mathrm{T}(t)$ and $\exists t\,(t \wedge \neg f) \equiv \mathrm{F}(f)$, we conclude that $\mathrm{T}, \mathrm{F} \in \langle\Gamma\rangle_{\neq}$. Further, by Lemma 13, we have that $(x = y) \in \langle\Gamma\rangle_{\neq}$. Together we have $\{\mathrm{T}, \mathrm{F}, =\} \subseteq \langle\Gamma\rangle_{\neq}$, and thus any formula in Δ is expressible as Γ-formula with existential quantifiers but without equality. By Lemma 12, it follows that $(x = y) \wedge t \wedge \neg f \in \langle\Gamma\rangle_{\nexists,\neq}$. Hence we can apply Lemma 20 to conclude. □

Lemma 24 ($\star$)**.** *If Γ is a CL that is strictly ess.neg. and strictly ess.pos., then* $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|) \in \mathbf{W}[1]$.

Lemma 25 ($\star$)**.** *If Γ is a CL that is strictly ess.neg. or strictly ess.pos., then* $\mathrm{p\text{-}ARG}(\Gamma, |\alpha|) \in \mathbf{W}[2]$.

Theorem 26. $\mathrm{p\text{-}ARG\text{-}Check}(\Gamma, |\alpha|)$*, for a CL Γ, is (1.)* **FPT** *if Γ is Schaefer, and (2.)* **para-DP**-*complete otherwise.*

Proof. 1. This follows from (Creignou, Egly, and Schmidt 2014, Theorem 6.1) as classically $\mathrm{ARG\text{-}Rel}(\Gamma) \in \mathbf{P}$.

2. Here, the membership follows as classically $\mathrm{ARG\text{-}Rel}(\Gamma) \in \mathbf{DP}$. Furthermore, the reduction in the proof of Creignou, Egly, and Schmidt (2014, Propositions 6.3 and 6.4) always uses a fixed size of the claim α.

As a consequence, certain slices of $\mathrm{ARG\text{-}Check}(\Gamma)$ are $\mathbf{DP}$-hard, giving the desired results. $\square$

Theorem 27. $\text{p-ARG-Rel}(\Gamma, |\alpha|)$*, for a CL* Γ*, is*

1. $\mathbf{FPT}$ *if* Γ *is either positive or negative.*
2. $\mathbf{para}\text{-}\mathbf{NP}$*-complete if* Γ *is Schaefer but neither strictly ess.neg. nor strictly ess.pos.*
3. $\mathbf{para}\text{-}\mathbf{\Sigma_2^P}$*-complete if* Γ *is not Schaefer.*

Proof. 1. This follows as classically $\mathrm{ARG\text{-}Rel}(\Gamma) \in \mathbf{P}$ by Creignou, Egly, and Schmidt (2014, Prop. 7.3).

2. Here, the membership follows because the classical problem is in $\mathbf{NP}$. We make a case distinction as whether Γ is ε-valid or not.

   Case 1. Let Γ be Schaefer and ε-valid, but neither positive nor negative. The hardness follows because the 2-slice of the problem is already $\mathbf{NP}$-hard (Creignou, Egly, and Schmidt 2014, Proposition 7.6).

   Case 2. Let Γ be Schaefer but neither ε-valid, nor strictly ess.neg. or strictly ess.pos The hardness follows from Theorem 19. This is due to the reason that p-ARG-Rel is always harder than p-ARG via the reduction $(\Delta, \alpha) \mapsto (\Delta \cup \{\psi\}, \psi, \alpha)$.

3. In this case, the membership is true because the classical problem is in $\mathbf{\Sigma_2^P}$. Hardness follows from a result of Creignou, Egly, and Schmidt (2014, Prop. 7.7). Notice that, while proving the hardness for each sub case, the claim α has fixed size in each reduction. This implies that certain slices in each case are $\mathbf{\Sigma_2^P}$-hard, consequently, giving the desired hardness results. $\square$

Parameters: Size of Support, Knowledge-Base

Regarding these parameters, we will always show a dichotomy: for the Schaefer cases, the problem is $\mathbf{FPT}$, otherwise we have a lower bound by the implication problem.

Recall that the collection Δ of formulas is not assumed to be consistent.

Theorem 28. $\text{p-ARG}(\Gamma, |\Delta|)$ *and* $\text{p-ARG-Rel}(\Gamma, |\Delta|)$*, for CLs* Γ*, are (1.)* $\mathbf{FPT}$ *if* Γ *is Schaefer, and (2.)* $\text{p-IMP}(\Gamma, |\Phi|)$*-hard and in* $\mathbf{para}\text{-}\mathbf{coNP}$ *otherwise.*

Proof. 1. Notice that the number of subsets of Δ is bounded by the parameter. Consequently, one simply checks each subset of Δ as a possible support Φ for α. Moreover, the size of each support Φ is also bounded by the parameter, as a result, one can determine the satisfiability and entailment in $\mathbf{FPT}$-time. This is because, the satisfiability and entailment for Schaefer languages is in $\mathbf{P}$.

2. For the lower bound, we have $\text{p-IMP}(\Gamma, |\Phi|) \leq^{\mathbf{FPT}} \mathrm{ARG}(\Gamma, |\Delta|) \leq^{\mathbf{FPT}} \text{p-ARG-Rel}(\Gamma, |\Delta|)$ by identities. For membership, we make case distinction as whether Γ is ε-valid or not.

   Case 1. Γ is ε-valid. The membership follows because the unparameterized problem $\mathrm{ARG}(\Gamma)$ is in $\mathbf{coNP}$ when Γ is ε-valid.

   Case 2. Γ is neither 0-valid nor 1-valid. The membership follows because for each candidate Φ, one needs to determine whether Φ is consistent and $\Phi \models \alpha$. The consistency can be checked in $\mathbf{FPT}$-time because $|\Phi|$ is bounded by the parameter. The entailment problem for non-Schaefer, non ε-valid languages is still in $\mathbf{para}\text{-}\mathbf{coNP}$ when $|\Phi|$ is the parameter.

   For $\text{p-ARG-Rel}(\Gamma, |\Delta|) \in \mathbf{para}\text{-}\mathbf{coNP}$, try all the subsets of Δ that contain ψ, as a candidate support. $\square$

When the support size $|\Phi|$ is considered as a parameter, the problems ARG and ARG-Rel become irrelevant. Consequently, we only consider the problem ARG-Check.

Corollary 29. $\text{p-ARG-Check}(\Gamma, |\Phi|)$*, for a CL* Γ*, is (1.)* $\mathbf{FPT}$ *if* Γ *is Schaefer, and (2.)* $\text{p-IMP}(\Gamma, |\Phi|)$*-hard and in* $\mathbf{para}\text{-}\mathbf{DP}$ *otherwise.*

Conclusion and Outlook

In this paper, we performed a two dimensional classification of reasoning in logic-based argumentation. On the one side, we studied syntactical fragments in the spirit of Schaefer's framework of co-clones. On the other side, we analysed a list of parameters and classified the parameterized complexity of three central reasoning problems accordingly.

As a take-away message we get that α as a parameter does not help to reach tractable fragments of p-ARG.

The case for $\text{p-ARG-Rel}(\Gamma, |\alpha|)$ when Γ is strictly ess.neg. or strictly ess.pos. is still open. Also, few tight complexity results have to be found and the implication problem regarding the parameter $|\Phi|$ has to be understood.

It is worth noting that for some CLs, e.g., those that are ε-valid, the problem p-ARG-Check is harder than p-ARG. This is because the problem p-ARG under consideration is the decision problem. Having the identity reduction from p-ARG-Check to p-ARG shows that the minimality is checked by solving the problem p-ARG, already. This shows that computing a minimal support is potentially harder than deciding whether such a support exists, unless the complexity classes $\mathbf{DP}$ and $\mathbf{coNP}$ coincide. We pose as an interesting open problem to classify the function version of ARG, in both, the classical and the parameterized setting.

Regarding other parameters, treewidth (Robertson and Seymour 1984) is a quite promising structural property that led to several $\mathbf{FPT}$-results in the parameterized setting: artificial intelligence (Gottlob and Szeider 2007), knowledge representation (Gottlob, Pichler, and Wei 2006), abduction in Datalog (Gottlob, Pichler, and Wei 2007), and databases (Grohe 2007). Fellows et al. (2012) show that abductive reasoning benefits from this parameter as well. Using a reduction between abduction and argumentation (Creignou, Egly, and Schmidt 2014) might yield $\mathbf{FPT}$-results in our setting. Furthermore, we plan to give a precise classification of p-IMP.

As further future work, we plan investigating the (parameterized) enumeration complexity (Fomin and Kratsch 2010; Creignou et al. 2017, 2019; Meier 2020) of reasoning in this setting.

Acknowledgements

This work was supported by the German Research Foundation (DFG) under the project number ME 4279/1-2. The authors would like to thank the anonymous reviewers for the valuable feedback they have provided.

References

Amgoud, L.; and Prade, H. 2009. Using arguments for making and explaining decisions. *Artif. Intell.* 173(3-4): 413–436.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards Artificial Argumentation. *AI Mag.* 38(3): 25–36. doi:10.1609/aimag.v38i3.2704.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds. 2018. *Handbook of Formal Argumentation*. College Publications.

Besnard, P.; and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artif. Intell.* 128(1-2): 203–235.

Besnard, P.; and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

Beyersdorff, O.; Meier, A.; Thomas, M.; and Vollmer, H. 2009. The complexity of propositional implication. *Inf. Process. Lett.* 109(18): 1071–1077. doi:10.1016/j.ipl.2009.06.015.

Böhler, E.; Reith, S.; Schnoor, H.; and Vollmer, H. 2005. Bases for Boolean co-clones. *Inf. Process. Lett.* 96(2): 59–66. doi:10.1016/j.ipl.2005.06.003.

Böhler, E.; Creignou, N.; Reith, S.; and Vollmer, H. 2004. Playing with Boolean blocks, part II: Constraint satisfaction problems. *ACM SIGACT-Newsletter* 35.

Chesñevar, C. I.; Maguitman, A. G.; and Loui, R. P. 2000. Logical models of argument. *ACM Comput. Surv.* 32(4): 337–383.

Creignou, N.; Egly, U.; and Schmidt, J. 2014. Complexity Classifications for Logic-Based Argumentation. *ACM Trans. Comput. Log.* 15(3): 19:1–19:20. doi:10.1145/2629421.

Creignou, N.; Khanna, S.; and Sudan, M. 2001. *Complexity classifications of Boolean constraint satisfaction problems*, volume 7 of *SIAM monographs on discrete mathematics and applications*. SIAM. ISBN 978-0-89871-479-1.

Creignou, N.; Ktari, R.; Meier, A.; Müller, J.; Olive, F.; and Vollmer, H. 2019. Parameterised Enumeration for Modification Problems. *Algorithms* 12(9): 189. doi:10.3390/a12090189.

Creignou, N.; Meier, A.; Müller, J.; Schmidt, J.; and Vollmer, H. 2017. Paradigms for Parameterized Enumeration. *Theory Comput. Syst.* 60(4): 737–758. doi:10.1007/s00224-016-9702-4.

Creignou, N.; Schmidt, J.; Thomas, M.; and Woltran, S. 2011. Complexity of logic-based argumentation in Post's framework. *Argument & Computation* 2(2-3): 107–129.

Downey, R. G.; and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Dung, P. M. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.* 77(2): 321–358.

Fellows, M. R.; Pfandler, A.; Rosamond, F. A.; and Rümmele, S. 2012. The Parameterized Complexity of Abduction. In Hoffmann, J.; and Selman, B., eds., *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. AAAI Press. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5048.

Fichte, J. K.; Hecher, M.; and Meier, A. 2019. Counting Complexity for Reasoning in Abstract Argumentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2827–2834. AAAI Press. doi:10.1609/aaai.v33i01.33012827.

Fomin, F. V.; and Kratsch, D. 2010. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer. ISBN 978-3-642-16532-0. doi:10.1007/978-3-642-16533-7.

Gottlob, G.; Pichler, R.; and Wei, F. 2006. Bounded Treewidth as a Key to Tractability of Knowledge Representation and Reasoning. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, 250–256. AAAI Press.

Gottlob, G.; Pichler, R.; and Wei, F. 2007. Efficient datalog abduction through bounded treewidth. In *AAAI*, 1626–1631.

Gottlob, G.; and Szeider, S. 2007. Fixed-Parameter Algorithms For Artificial Intelligence, Constraint Satisfaction and Database Problems. *The Computer Journal* 51(3): 303–325. ISSN 0010-4620. doi:10.1093/comjnl/bxm056.

Grohe, M. 2007. The Complexity of Homomorphism and Constraint Satisfaction Problems Seen from the Other Side. *J. ACM* 54(1). ISSN 0004-5411. doi:10.1145/1206035.1206036.

Liberatore, P. 2005. Redundancy in logic I: CNF propositional formulae. *Artif. Intell.* 163(2): 203–232.

Mahmood, Y.; Meier, A.; and Schmidt, J. 2020. Parameterised Complexity of Abduction in Schaefer's Framework. In *Logical Foundations of Computer Science - International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4-7, 2020, Proceedings*, 195–213. doi:10.1007/978-3-030-36755-8_13.

Mahmood, Y.; Meier, A.; and Schmidt, J. 2021. Parameterized Complexity of Logic-Based Argumentation in Schaefer's Framework. *CoRR* abs/2102.11782. URL https://arxiv.org/abs/2102.11782.

Meier, A. 2020. *Parametrised enumeration*. Habilitation thesis, Leibniz Universität Hannover. doi:10.15488/9427.

Nordh, G.; and Zanuttini, B. 2008. What makes propositional abduction tractable. *Artif. Intell.* 172(10): 1245–1284. doi:10.1016/j.artint.2008.02.001.

Parsons, S.; Wooldridge, M. J.; and Amgoud, L. 2003. Properties and Complexity of Some Formal Inter-agent Dialogues. *J. Log. Comput.* 13(3): 347–376.

Post, E. L. 1941. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies* 5: 1–122.

Prakken, H.; and Vreeswijk, G. 2002. *Logics for Defeasible Argumentation*, 219–318. Dordrecht: Springer Netherlands.

Rago, A.; Cocarascu, O.; and Toni, F. 2018. Argumentation-Based Recommendations: Fantastic Explanations and How to Find Them. In *IJCAI*, 1949–1955. ijcai.org.

Robertson, N.; and Seymour, P. D. 1984. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B* 36(1): 49–64. doi:10.1016/0095-8956(84)90013-3.

Schaefer, T. J. 1978. The Complexity of Satisfiability Problems. In Lipton, R. J.; Burkhard, W. A.; Savitch, W. J.; Friedman, E. P.; and Aho, A. V., eds., *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, 216–226. ACM. doi:10.1145/800133.804350.

Schnoor, H.; and Schnoor, I. 2008. Partial Polymorphisms and Constraint Satisfaction Problems. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, 229–254. doi:10.1007/978-3-540-92800-3_9.

Sipser, M. 1997. *Introduction to the theory of computation*. PWS Publishing Company.