An Argumentative Characterization of Disjunctive Logic Programming

Jesse Heyninck[1*] and Ofer Arieli[2]

[1] Institute of Philosophy II, Ruhr University Bochum
[2] School of Computer Science, The Academic College of Tel-Aviv

Abstract. This paper extends the result of Caminada and Schulz [6, 7] by showing that assumption-based argumentation can represent not only normal logic programs, but also disjunctive logic programs. For this, we incorporate the setting of Heyninck and Arieli (see [19, 20]), in which reasoning with assumption-based argumentation frameworks is based on certain core *logics* and that the strict/defeasible assumptions may be arbitrary formulas in those logics. In our case, the core logic respects some inference rules for disjunction, which allows disjunctions in the heads of the programs' rules to be handled properly.

Keywords: Knowledge representation and reasoning, Non-monotonic reasoning, Computational models of argument, Disjunctive logic programming, Structured argumentation.

1 Introduction

Logic programming (LP) and assumption-based argumentation (ABA) are two primary methods for knowledge representation and non-monotonic reasoning, originated from similar intuitions and applied in different contexts. In fact, to a large extent, the introduction of ABA systems was motivated by an argumentative interpretation of LP semantics [4]. The former provides a Dung-style representation [11] of coherent sets of formulas that admits other sets of formulas as their contraries, and the latter combines reasoning based on strict inference rules with an interpretation of negation as 'failure to prove the converse' [21].

The similar ground of LP and ABA calls upon translation methods for revealing the exact relations between them, and for importing reasoning methods from one formalism to the other. For example, argumentative characterizations of logic programming have been proven useful for explanation [27, 28] and visualization [26] of inferences in logic programming. Among the works that translate LP and ABA we recall the one of Schulz and Toni [29, 30] that provides a

* This work is supported by the Israel Science Foundation (grant number 817/15). The first author is also partially supported by the Sofja Kovalevskaja award of the Alexander von Humboldt Foundation, funded by the German Ministry for Education and Research, the FCT projects RIVER (PTDC/CCI-COM/30952/2017) and NOVA LINCS (UID/CEC/04516/2013).

one-to-one correspondence between the 3-valued stable models for normal logic programs [23] and complete labelings for ABA frameworks. Recent works of Caminada and Schulz [6, 7] show the equivalence between ABA semantics and normal logic programs. Their research is restricted to normal logic programs, where negations may occur in the bodies of the rules but only atoms are allowed in the head of the rules. Yet, a faithful modeling of real-world problems often requires to cope with *incomplete knowledge*, which is not possible in the scope of normal logic programs. This is a primary motivation in the introduction of *disjunctive logic programming*, where disjunctions are allowed in the heads of the rules and negation may occur in their bodies.

Under usual complexity assumptions, disjunctive logic programs were shown to be strictly more expressive than normal logic programs [14, 18]. Moreover, in the last decades they have been efficiently implemented and widely applied, thus became a key technology in knowledge representation (see, e.g., [31]). Therefore, in this paper we set out to generalize the argumentative characterization of logic programming for disjunctive logic programs. For this, we incorporate the ideas of [19, 20], extending ABA frameworks to propositional formulas (as the defeasible or strict assumption at hand), expressed in Tarskian logics. This allows to augment the core logic of the ABA frameworks with inference rules for handling disjunctive assertions, and so associate the stable extensions of such frameworks with the stable models of the corresponding disjunctive logic programs.

2 Preliminaries

We denote by $\mathcal{L}$ a propositional language. Atomic formulas in $\mathcal{L}$ are denoted by p, q, r, s (possibly indexed), compound formulas are denoted by ψ, ϕ, σ, and sets of formulas in $\mathcal{L}$ are denoted by $\Gamma, \Delta, \Theta, \Lambda$. We shall assume that $\mathcal{L}$ contains a conjunction (denoted as usual in logic programing by a comma), disjunction $\vee$, implication $\rightarrow$, a negation operator $\sim$, and a propositional constant T for truth. Also, we denote $\sim\Gamma = \{\sim\gamma \mid \gamma \in \Gamma\}$. The powerset of $\mathcal{L}$ is denoted $\wp(\mathcal{L})$.

2.1 Assumption-Based Argumentation

Definition 1. A (propositional) *logic* for a language $\mathcal{L}$ is a pair $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$, where $\vdash$ is a (Tarskian) consequence relation for $\mathcal{L}$, that is, a binary relation between sets of formulas and formulas in $\mathcal{L}$, which is reflexive (if $\psi \in \Gamma$ then $\Gamma \vdash \psi$), monotonic (if $\Gamma \vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \psi$), and transitive (if $\Gamma \vdash \psi$ and $\Gamma', \psi \vdash \phi$, then $\Gamma, \Gamma' \vdash \phi$).

The next definition, adapted from [19], generalizes the definition in [4] of assumption-based frameworks.

Definition 2. An *assumption-based framework* is a tuple $\mathbf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$, where:

- $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ is a propositional Tarskian logic

– Γ (the *strict assumptions*) and Λ (the *candidate or defeasible assumptions*) are distinct countable sets of $\mathcal{L}$-formulas where Λ is assumed to be nonempty.
– $- : \Lambda \to \wp(\mathcal{L})$ is a contrariness operator, assigning a finite set of $\mathcal{L}$-formulas to every defeasible assumption in Λ.

Note 1. Unlike the setting of [4], an ABF may be based on *any* Tarskian logic $\mathfrak{L}$. Also, the strict as well as the candidate assumptions are formulas that may not be just atomic. Concerning the contrariness operator, note that it is not a connective of the language $\mathcal{L}$, as it is restricted only to the candidate assumptions.

Defeasible assertions in an ABF may be attacked by counterarguments.

Definition 3. Let $\mathbf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ be an assumption-based framework, $\Delta, \Theta \subseteq \Lambda$, and $\psi \in \Lambda$. We say that Δ *attacks* ψ iff $\Gamma, \Delta \vdash \phi$ for some $\phi \in -\psi$. Accordingly, Δ attacks Θ if Δ attacks some $\psi \in \Theta$.

The last definition gives rise to the following adaptation to ABFs of the usual semantics for abstract argumentation frameworks [11].

Definition 4. ([4]) Let $\mathbf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ be an assumption-based framework, and let $\Delta \subseteq \Lambda$. Then Δ is *conflict-free* iff there is no $\Delta' \subseteq \Delta$ that attacks some $\psi \in \Delta$. We say that Δ is *stable* iff it is conflict-free and attacks every $\psi \in \Lambda \setminus \Delta$. The set of stable extensions of $\mathbf{ABF}$ is denoted by $\mathsf{Stb}(\mathbf{ABF})$. [3]

2.2 Disjunctive Logic Programs

Definition 5. A *disjunctive logic program* π is a finite set of expressions (rules) of the form $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \to p_1 \vee \ldots \vee p_n$, where $p_1 \vee \ldots \vee p_n$ is called the *head* of the rule, and $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k$ is its *body*. When each head of a rule in π is either empty or consists of an atomic formula (i.e., $n \leq 1$), we say that π is a *normal logic program*. A logic program π is *positive* if $k = 0$ for every rule in π. We denote by $\mathcal{A}(\pi)$ the set of atomic formulas that appear in π.

In what follows, unless otherwise stated, when referring to a logic program we shall mean that it is disjunctive. The semantics of a logic program π is defined as follows:

Definition 6. A set $M \subseteq \mathcal{A}(\pi)$ *satisfies* a rule $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \to p_1 \vee \ldots \vee p_n$ in π iff either $q_i \notin M$ for some $1 \leq i \leq m$, or $r_i \in M$ for some $1 \leq i \leq k$, or $p_i \in M$ for some $1 \leq i \leq n$. We say that M is a *model* of π if it satisfies every rule in π.

Definition 7. Let π be a disjunctive logic program and let $M \subseteq \mathcal{A}(\pi)$.

[3] In many presentations of assumption-based argumentation, extensions are required to be closed, i.e. they should contain any assumption they imply. Since the translation below will always give rise to the so-called *flat* ABFs (that is, ABFs for which a set of assumptions can never imply assumptions outside the set; See Note 3 below), closure of extensions is trivially satisfied.

– The *Gelfond-Lifschitz reduct* [16] of π with respect to M is the disjunctive logic program π^M, where $q_1, \ldots, q_m \to p_1 \vee \ldots \vee p_n \in \pi^M$ iff there is a rule $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \to p_1 \vee \ldots \vee p_n \in \pi$ and $r_i \notin M$ for every $1 \leqslant i \leqslant k$.
– M is a *stable model* of π iff it is a $\subseteq$-minimal model of π^M.

3 From DLP to ABA

Given a disjunctive logic program π, we show a one-to-one correspondence between the stable models of π and the the stable extensions of an ABA framework that is induced from π. First, we describe and motivate the translation, then we prove its correctness.

3.1 The Translation

All the ABA frameworks that are induced from disjunctive logic programs will be based on the same core logic, which is constructed by three inference rules: Modus Ponens (MP), Resolution (Res) and Reasoning by Cases (RBC):

$$[\mathsf{MP}]\quad \frac{\phi_1, \ldots, \phi_n \to \psi \qquad \phi_1 \quad \phi_2 \quad \cdots \quad \phi_n}{\psi}$$

$$[\mathsf{Res}]\quad \frac{\psi'_1 \vee \ldots \vee \psi'_m \vee \phi_1 \vee \ldots \vee \phi_n \vee \psi''_1 \vee \ldots \vee \psi''_k \qquad \sim\phi_1 \ \cdots \ \sim\phi_n}{\psi'_1 \vee \ldots \vee \psi'_m \vee \ldots \vee \psi''_1 \vee \ldots \vee \psi''_k}$$

$$[\mathsf{RBC}]\quad \frac{\begin{array}{cccccc}\phi_1 & \phi_2 & & \phi_n & \\ \vdots & \vdots & & \vdots & \\ \psi & \psi & \cdots & \psi & \phi_1 \vee \ldots \vee \phi_n\end{array}}{\psi}$$

In what follows we denote by $\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ the logic based on the language $\mathcal{L}$ which consists of disjunctions of atoms ($p_1 \vee \ldots \vee p_n$ for $n \geq 1$), negated atoms ($\sim p$), or formulas of the forms of the program rules in Definition 5. Accordingly, we shall use only fragments of the inference rules above, in which in [Res] and [RBC] the formulas ψ, ψ_i are disjunctions of atomic formulas and ϕ_i are atomic formulas. In [MP], ψ is a disjunction of atomic formulas and $\phi_i \in \{p_i, \sim p_i\}$ are literals. Now, $\Delta \vdash \phi$ iff ϕ is either in Δ or is derivable from Δ using the inference rules above. In other words, $\Delta \vdash \phi$ iff $\phi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$, where $\mathsf{Cn}_{\mathfrak{L}}(\Delta)$ is the $\mathfrak{L}$-based transitive closure of Δ (namely, the $\subseteq$-smallest set that contains Δ and is closed under [MP], [Res] and [RBC]). Notice that for any $\phi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$, if ϕ is not of the form $p_1 \vee \ldots \vee p_n$ then $\phi \in \Delta$.

Note 2. Since $\to \psi$ is identified with $\mathsf{T} \to \psi$, [MP] implies Reflexivity: [Ref] $\dfrac{\to \psi}{\psi}$.

Definition 8. The ABF that is *induced* by a disjunctive logic program π is defined by: $\mathbf{ABF}(\pi) = \langle \mathfrak{L}, \pi, \sim\mathcal{A}(\pi), - \rangle$, where $-\sim p = \{p\}$ for every $p \in \mathcal{A}(\pi)$.

Example 1. Let $\pi_1 = \{\rightarrow p \vee q;\ p \rightarrow q;\ q \rightarrow p\}$. The attack diagram of the induced framework $\mathbf{ABF}(\pi_1)$ is shown in Figure 1a. Note that in this case $\{p, q\}$ is the stable model of π_1 and $\emptyset$ is the stable extension of $\mathbf{ABF}(\pi_1)$.

Fig. 1: Attack diagrams for Examples 1, 3c (left) and Example 3b (right)

Definition 9. Let π be a disjunctive logic program and $\Theta \subseteq \mathcal{A}(\pi)$. We denote:

- $\lfloor \sim \Theta \rfloor = \Theta$ (Thus, $\lfloor \cdot \rfloor$ eliminates the leading $\sim$ from the formulas).
- If $\Delta \subseteq \sim\mathcal{A}(\pi)$ then $\underline{\Delta} = \mathcal{A}(\pi) \setminus \lfloor \Delta \rfloor$.
- If $\Delta \subseteq \mathcal{A}(\pi)$ then $\overline{\Delta} = \sim(\mathcal{A}(\pi) \setminus \Delta)$.

In other words, $\underline{\Delta}$ (respectively, $\overline{\Delta}$) takes the complementary set of Δ and removes (respectively, adds) the negation-as-failure operator from (respectively, to) the prefix of its formulas.

Example 2. Consider again the program π_1 of Example 1, and let $\Delta = \sim\mathcal{A}(\pi_1) = \{\sim p, \sim q\}$. Then $\underline{\Delta} = \emptyset$ and $\overline{\emptyset} = \Delta$.

The semantic correspondence between a logic program and the induced ABF is obtained by the following result:

1. If Δ is a stable extension of $\mathbf{ABF}(\pi)$ then $\underline{\Delta}$ is a stable model of π, and
2. If Δ is a stable model of π then $\overline{\Delta}$ is a stable extension of $\mathbf{ABF}(\pi)$.

First, we provide some examples and notes concerning related results.

Example 3. Caminada and Schulz [6, 7] consider the correspondence between ABA systems and normal logic programs. In our notations, the ABF that they associate with a normal logic program π is $\mathbf{ABF}_{\mathsf{Norm}}(\pi) = \langle \mathfrak{L}_{\mathsf{MP}}, \pi, \sim\mathcal{A}(\pi), - \rangle$ constructed as in Definition 8, except that $\mathfrak{L}_{\mathsf{MP}}$ is defined by Modus Ponens only.

a) To see that $\mathbf{ABF}_{\mathsf{Norm}}$ is not adequate for disjunctive logic programs, consider the program $\pi_2 = \{\rightarrow p \vee q\}$. This program has two stable models: $\{p\}$ and $\{q\}$. However, the only stable extension of $\mathbf{ABF}_{\mathsf{Norm}}(\pi_2)$ is $\{\sim p, \sim q\}$. We can enforce $\{\sim p\}$ and $\{\sim q\}$ being stable models by requiring that $\pi_2 \cup \{\sim p\} \vdash q$ and vice versa. For this, we need resolution [Res].

b) Adding only [Res] to $\mathfrak{L}_{\mathsf{MP}}$ (i.e, without [RBC]) as the inference rules for the logic is yet not sufficient. To see this consider again the program π_1 from Example 1. Recall that $\{p, q\}$ is the sole stable model of π_1. The attack graph of the ABF based on [MP] and [Res] is shown in Figure 1b. In the notations of Definition 2, we have: $\Gamma = \pi_2$ and $\Lambda = \{\sim p, \sim q\}$, thus by [MP] on $\rightarrow p \vee q$ we conclude $\Gamma \vdash p \vee q$, and by [Res] it holds that that $\Gamma, \sim p \vdash q$. Thus, since $q \rightarrow p \in \Gamma$, by [MP] we get $\Gamma, \sim p \vdash p$. It follows that $\sim p$ attacks itself, (similarly $\sim q$ attacks itself), thus there is no stable extension in this case.

c) Example 1 shows that for the ABF that is induced from π_1 according to Definition 8 (using all the three inference rules considered at the beginning of this section) the correspondence between the stable model of π_1 and the stable extension of $\mathbf{ABF}(\pi_1)$ is preserved and thus the translation is adequate for π_1. This is not a coincidence, as we show in the next section.

Note 3. The translation in Definition 8 always gives rise to a so-called *flat* ABF, that is, an ABF for which there is no $\Delta \subseteq \Lambda$ and $\psi \in \Lambda \setminus \Delta$ such that $\Gamma, \Delta \vdash \psi$:

Proposition 1. *For every disjunctive logic program π and the induced $\mathbf{ABF}(\pi) = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle = \langle \mathfrak{L}, \pi, \sim \mathcal{A}(\pi), - \rangle$, if $\Delta \subseteq \Lambda$ and $\Gamma, \Delta \vdash \psi$, then $\psi \notin \Lambda \setminus \Delta$.*

Proof. Suppose that $\Gamma, \Delta \vdash \psi$ for some $\Delta \subseteq \Lambda$. Since Λ contains only formulas of the form $\sim p$, and since $\Gamma = \pi$, then if $\Gamma, \Delta \vdash \sim p$, necessarily $\sim p \in \Delta$. □

3.2 Proof of Correctness

The correctness of the translation follows from Propositions 2 and 3 below. First, we need some definitions and lemmas. The proofs of Lemmas 2, 3 and 5 are omitted due to space restrictions. In what follows $\mathfrak{L}$ denotes the logic defined in Section 3.1, and π is an arbitrary (disjunctive) logic program.

Definition 10. Let M be set of atomic formulas, p, p_i, q_j atomic formulas, and $l_j \in \{q_j, \sim q_j\}$ literals. We denote:

- $M \models p$ iff $p \in M$,
- $M \models \sim p$ iff $p \notin M$,
- $M \models p_1 \vee \ldots \vee p_n$ iff $M \models p_i$ for some $1 \leq i \leq n$,
- $M \models l_1, \ldots, l_m$ iff $M \models l_j$ for every $1 \leq j \leq m$,
- $M \models l_1, \ldots, l_m \rightarrow p_1 \vee \ldots \vee p_n$ iff either $M \models p_1 \vee \ldots \vee p_n$, or $M \not\models l_1, \ldots, l_m$ (the latter means that it is *not the case* that $M \models l_1, \ldots, l_m$).

Given a set $\mathcal{S}$ of $\mathcal{L}$-formulas, we denote by $M \models \mathcal{S}$ that $M \models \psi$ for every $\psi \in \mathcal{S}$.[4]

Lemma 1. *Let Δ be a set of $\mathcal{L}$-formulas that are either of the form $\sim p$ or of the form $r_1, \ldots, r_m, \sim q_1, \ldots, \sim q_k \rightarrow p_1 \vee \ldots \vee p_n$. Then:*

[4] Since the underlying language consists only of negated atoms or formulas of the form of program rules (see Definition 5), this definition indeed covers all the possible sets $\mathcal{S}$ of $\mathcal{L}$-formulas.

a) If $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$ *then* $M \models \psi$ *for every* M *such that* $M \models \Delta$.
b) If $\psi = s_1 \vee \ldots \vee s_l$ *and* $M \models \psi$ *for every* M *s.t.* $M \models \Delta$, *then* $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$.

Note 4. Part (b) of Lemma 1 does not hold for *any formula* ψ, but only for a disjunction of atoms. To see this, let $\Delta = \{\sim s,\ p \to s\}$. The only $M \subseteq \{p, s\}$ such that $M \models \Delta$ is $M = \emptyset$. Thus, for every M such that $M \models \Delta$ it holds that $M \models \sim p$. However, $\sim p$ cannot be derived from Δ using [MP], [Res] and [RBC].

Proof. We prove Part (a) of the lemma by induction on the number of applications of the inference rules in the derivation of $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$.

For the base step, no inference rule is applied in the derivation of ψ, thus $\psi \in \Delta$. Since $M \models \Delta$, we have that $M \models \psi$.

For the induction step, we consider three cases, each one corresponds to an application of a different inference rule in the last step of the derivation of ψ:

1. Suppose that the last step in the derivation of ψ is an application of Resolution. Then $\psi = p'_1 \vee \ldots \vee p'_m \vee \ldots \vee p''_1 \vee \ldots \vee p''_k$ is obtained by [Res] from $p'_1 \vee \ldots \vee p'_m \vee q_1 \vee \ldots \vee q_n \vee p''_1 \vee \ldots \vee p''_k$ and $\sim q_i$ $(i = 1, \ldots, n)$. Suppose that $M \models \Delta$. Since $\sim q_i \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$ iff $\sim q_i \in \Delta$ and since $M \models \Delta$, $M \models \sim q_i$ $(i = 1, \ldots, n)$, thus $M \not\models q_i$ $(i = 1, \ldots, n)$. By the induction hypothesis, $M \models p'_1 \vee \ldots \vee p'_m \vee q_1 \vee \ldots \vee q_n \vee p''_1 \vee \ldots \vee p''_k$. By Definition 10, then, $M \models p'_i$ for some $1 \leq i \leq m$, or $M \models p''_j$ for some $1 \leq j \leq k$. By Definition 10 again, $M \models \psi$.
2. Suppose that the last step in the derivation of ψ is an application of Reasoning by Cases, and let $M \models \Delta$. By induction hypothesis, $M \models p_1 \vee \ldots \vee p_n$, and $M \models \psi$ in case that $M \models p_j$ for some $1 \leq j \leq n$. But by Definition 10 the former assumption means that there is some $1 \leq j \leq n$ for which $M \models p_j$, therefore $M \models \psi$.
3. Suppose that the last step in the derivation of ψ is an application of Modus Ponens, and let $M \models \Delta$. By induction hypothesis $M \models l_i$, where $l_i \in \{p_i, \sim p_i\}$ for $i = 1, \ldots, n$. Thus, by Definition 10, $M \models l_1, \ldots, l_n$. On the other hand, by induction hypothesis again, $M \models l_1, \ldots, l_n \to \psi$. By Definition 10 this is possible only if $M \models \psi$.

We now turn to Part (b) of the lemma. If there is no M such that $M \models \Delta$ the claim is trivially satisfied.

Suppose then that $M \models \psi$ for every M such that $M \models \Delta$, yet $\psi \notin \mathsf{Cn}_{\mathfrak{L}}(\Delta)$ (where $\psi = r_1 \vee \ldots \vee r_m$ for some $m \geqslant 1$). We show that this leads to a contradiction by constructing an M' for which $M' \models \Delta$ but $M' \not\models \psi$. For this, we consider the following set of the minimal disjunctions of a set of formulas $\mathcal{S}$:

$$\mathsf{MD}(\mathcal{S}) = \{q_1 \vee \ldots \vee q_n \in \mathcal{S} \mid \nexists\{i_1, \ldots, i_m\} \subsetneq \{1, \ldots, n\} \text{ s.t. } q_{i_1} \vee \ldots \vee q_{i_m} \in \mathcal{S}\}.$$

We first show that if $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$ then there is an $1 \leqslant i \leqslant n$ such that $q_i \notin \{r_1, \ldots, r_m\}$ and $\sim q_i \notin \Delta$. Indeed, suppose first for a contradiction that $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$, yet for every $1 < i \leqslant n$ either $q_i \in \{r_1, \ldots, r_m\}$ or $\sim q_i \in \Delta$. In that case, by [Res], $r_1 \vee \ldots \vee r_m \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$, contradicting the

original supposition that $r_1 \vee \ldots \vee r_m \notin \mathsf{Cn}_{\mathfrak{L}}(\Delta)$. Suppose now, again towards a contradiction, that $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$, yet for every $1 \leqslant i \leqslant n$, $\sim q_i \in \Delta$. In that case, by [Res] again, $q_i \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$ (for every i), but this, together with the assumption the $\sim q_i \in \Delta$ (for every i), contradicts the assumption that there is an M such that $M \models \Delta$.

We thus showed that in any case, if $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$, then there is an $1 \leqslant i \leqslant n$ such that $q_i \notin \{r_1, \ldots, r_m\}$ and $\sim q_i \notin \Delta$.

We now construct the model M' such that $M' \models \Delta$ and $M' \not\models r_1 \vee \ldots \vee r_m$. In more detail, let M' contain exactly one q_i with $1 \leqslant i \leqslant n$ and $q_i \notin \{r_1, \ldots, r_m\}$ and $\sim q_i \notin \Delta$ for every $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$. (If there is more than one such i, take i which is minimal among $1 \leqslant i \leqslant n$.) As shown above, there is at least one such i for every formula $q_1 \vee \ldots \vee q_n \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$.

We now show that (1) $M' \models \Delta$ and (2) $M' \not\models r_1 \vee \ldots \vee r_m$.

Item (1): Suppose that $\phi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$. We have to consider two possibilities: $\phi = \sim s$ or $\phi = q_1 \vee \ldots \vee q_n$. In the first case, by construction, $s \notin M'$ and thus $M' \models \sim s$. In the second case, there is a $q_{i_1} \vee \ldots \vee q_{i_m} \in \mathsf{MD}(\mathsf{Cn}_{\mathfrak{L}}(\Delta))$ such that $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$. By construction, there is a $1 \leqslant j \leqslant m$ such that $q_{i_j} \in M'$. Thus, $M \models q_1 \vee \ldots \vee q_n$.

Item (2): By construction $r_i \notin M'$ $(i = 1, \leqslant, m)$, thus $M' \not\models r_1 \vee \ldots \vee r_m$. □

Lemma 2. *For every sets M, N of literals, if $N \setminus M \neq \emptyset$ then $N \not\models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$.*

Lemma 3. *Given a logic program π, if M is a minimal model of a logic program $\pi' \subseteq \pi^M$, then for every $N \subset M$, $N \not\models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$.*

Lemma 4. *Let M be a stable model of π. Then it is the (unique) minimal subset N of $\mathcal{A}(\pi)$ such that $N \models \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \overline{M})$.*

Proof. Let M be a stable model of π. We first show that $M \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. Let $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. Then it has an $\mathfrak{L}$-derivation $\mathsf{D}_{\mathfrak{L}}(\psi)$. We show by induction on the size of $\mathsf{D}_{\mathfrak{L}}(\psi)$ that $M \models \psi$.

For the base step, no inference rule is applied in $\mathsf{D}_{\mathfrak{L}}(\psi)$, thus $\psi = \sim\phi \in \overline{M}$. Since this means that $\phi \notin M$, we have that $M \models \psi$.

For the induction step, we consider three cases, each one corresponds to an application of a different inference rule in the last step of $\mathsf{D}_{\mathfrak{L}}(\psi)$:

1. Suppose that the last step in $\mathsf{D}_{\mathfrak{L}}(\psi)$ is an application of Resolution. Then $\psi = p'_1 \vee \ldots \vee p'_m \vee \ldots \vee p''_1 \vee \ldots \vee p''_k$ is obtained by [Res] from $p'_1 \vee \ldots \vee p'_m \vee q_1 \vee \ldots \vee q_n \vee p''_1 \vee \ldots \vee p''_k$ and $\sim q_i$ $(i = 1, \ldots, n)$. Since $\sim q_i \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$ means that $\sim q_i \in \overline{M}$, we have $q_i \notin M$ for every $1, \ldots, n$. By the inductive hypothesis, $M \models p'_1 \vee \ldots \vee p'_m \vee q_1 \vee \ldots \vee q_n \vee p''_1 \vee \ldots \vee p''_k$. Thus, by Definition 10, $M \models p'_i$ for some $1 \leq i \leq m$, or $M \models p''_j$ for some $1 \leq j \leq k$. By Definition 10 again, $M \models \psi$.
2. Suppose that the last step in $\mathsf{D}_{\mathfrak{L}}(\psi)$ is an application of Reasoning by Cases. By induction hypothesis we know that $M \models p_1 \vee \ldots \vee p_n$, and that $M \models \psi$ in case that $M \models p_j$ for some $1 \leq j \leq n$. But by Definition 10 the former assumption means that there is some $1 \leq j \leq n$ for which $M \models p_j$, therefore $M \models \psi$.

3. Suppose that the last step in $\mathsf{D}_{\mathfrak{L}}(\psi)$ is an application of Modus Ponens on the rule $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_l \in \pi$. By induction hypothesis $M \models p_i$, for every $1 \leqslant i \leqslant n$. Also, for every $1 \leqslant i \leqslant m$, $\sim q_i \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$ implies $q_i \notin M$. Thus, $p_1, \ldots, p_n \rightarrow r_1 \vee \ldots \vee r_l \in \pi^M$. Since M is a model of π^M, $M \models r_i$ for some $1 \leqslant i \leqslant l$. Thus, $M \models r_1 \vee \ldots \vee r_l$.

Thus, we have shown that $M \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. By Lemma 2, for no $N \subseteq \mathcal{A}(\pi)$ such that $N \setminus M \neq \emptyset$ it holds that $N \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. Thus, if there is some $N \subseteq \mathcal{A}(\pi)$ such that $N \models Cn_{\mathfrak{L}}(\overline{M} \cup \pi)$, then $N \subseteq M$. But if $N \subset M$, by Lemma 3 we have that $N \not\models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. Thus, M is the unique set of literals that models $\mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. □

Corollary 1. *Let M be a stable model of π. Then $p \in M$ iff $p \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$.*

Proof. Suppose first that $p \in M$. Since by Lemma 4 M is the unique model of $\mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$, by Lemma 1 it holds that $M \models p$ implies that $p \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. For the converse, suppose that $p \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. By Lemma 4, $M \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$ and thus $M \models p$. □

Lemma 5. *Let π be a disjunctive logic program, $\Delta = \{\sim p_1, \ldots, \sim p_n\}$ and $r \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$. If M is a model of $\pi^{\lfloor \Delta \rfloor}$ and $M \subseteq \lfloor \Delta \rfloor$, then $r \in M$.*

Now we can show the main results of this section.

Proposition 2. *If M is a stable model of π, then $\overline{M}$ is a stable extension of* $\mathbf{ABF}(\pi)$.

Proof. Suppose that M is a stable model of π. We show first that $\overline{M}$ is conflict-free in $\mathbf{ABF}(\pi)$. Otherwise, there is some $\sim p \in \overline{M}$ such that $\pi, \overline{M} \vdash p$. The former implies that $p \notin M$. But since M is a model of π^M, by Lemma 5, the fat that $\pi, \overline{M} \vdash p$ implies that $p \in M$, a contradiction.

We now show that $\overline{M}$ attacks every $\sim p \in \sim\mathcal{A}(\pi) \setminus \overline{M}$. This means that we have to show that $\overline{M} \vdash p$ for every $p \in M$. This follows from Corollary 1. □

Proposition 3. *If $\mathcal{E}$ is a stable extension of* $\mathbf{ABF}(\pi)$ *then $\underline{\mathcal{E}}$ is a stable model of π.*

Proof. We first show that $\underline{\mathcal{E}}$ is a model of $\pi^{\underline{\mathcal{E}}}$. Let $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k \in \pi$. If $q_j \in \underline{\mathcal{E}}$ for some $1 \leq j \leq m$ we are done. Otherwise, $q_1, \ldots, q_m \notin \underline{\mathcal{E}}$, and so $p_1, \ldots, p_n \rightarrow r_1 \vee \ldots \vee r_k \in \pi^{\underline{\mathcal{E}}}$. Again if $p_j \notin \underline{\mathcal{E}}$ for some $1 \leq j \leq n$ we are done. Thus, suppose that $p_1, \ldots, p_n \in \underline{\mathcal{E}}$. In other words, $\sim p_1, \ldots, \sim p_n \notin \mathcal{E}$ and $\sim q_1, \ldots, \sim q_m \in \mathcal{E}$. Since $\mathcal{E}$ is stable, the latter implies that $\pi, \mathcal{E} \vdash p_i$ for every $1 \leqslant i \leqslant n$. This means that $\pi, \mathcal{E} \vdash r_1 \vee \ldots \vee r_k$ (since $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k \in \pi$ and $\sim q_1, \ldots, \sim q_m \in \underline{\mathcal{E}}$). Suppose now for a contradiction that $\sim r_i \in \mathcal{E}$ for every $1 \leqslant i \leqslant k$. Then by [Res], $\pi, \mathcal{E} \vdash r_i$ for every $1 \leqslant i \leqslant k$ and thus $\mathcal{E}$ attack itself, which contradicts the fact that $\mathcal{E}$ is conflict-free. Consequently, there is at least one $1 \leqslant i \leqslant k$ such that $\sim r_i \notin \mathcal{E}$ and thus $r_i \in \underline{\mathcal{E}}$, which means that $\underline{\mathcal{E}}$ satisfies $p_1, \ldots, p_n \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k$.

To show the minimality of $\underline{\mathcal{E}}$, suppose that there is an $M \subsetneq \underline{\mathcal{E}}$ that is a model of $\pi^{\underline{\mathcal{E}}}$. Let $p \in \underline{\mathcal{E}} \setminus M$. Since $p \in \underline{\mathcal{E}}$, $\sim p \notin \mathcal{E}$. Since $\mathcal{E}$ is stable, this means that $\pi, \mathcal{E} \vdash p$. By Lemma 5, any model of $\pi^{\underline{\mathcal{E}}}$ satisfies p, a contradiction to $p \notin M$. □

4 From ABA to DLP

The main body of literature on ABA frameworks is concentrated on languages that consist solely of formulas of the form $p_1, \ldots, p_n \rightarrow p$ (where $p, p_1, \ldots, p_n$ are atomic formulas). For such assumption-based frameworks (or at least when the frameworks are flat) it has been shown that there is a straightforward translation into normal logic programs that preserve equivalence for all the commonly studied argumentation semantics (see [6, 7]). To the best of our knowledge, the more complicated classes of ABA frameworks that are considered in this paper (and which are based on a logic allowing to reason with disjunctive rules of the form $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k$) have not been investigated for other purposes other than the translation of DLPs. We thus do not see any motivation for investigating the reverse translation from these assumption-based frameworks into disjunctive logic programs. We do believe, however, that it is interesting to see if the more general class of assumption-based frameworks that are based on an arbitrary propositional logic (as defined and studied in e.g. [19, 20]) can be translated in a class of logic programs, probably more general than disjunctive ones. This is a subject for a future work.

5 Conclusion, in View of Related Work

This work generalizes translations from logic programming into assumption-based argumentation to cover also disjunctive logic programs. Somewhat surprisingly, there are only few works that investigate the representation of disjunctive defeasible reasoning by general argumentation frameworks. The most similar work to the research in this paper is probably that in [33], where a representation of DLPs by structured argumentation frameworks is proposed. In this framework, the assumptions are disjunctions of negated atoms $\sim p_1 \vee \ldots \vee \sim p_n$, instead of just negated atoms as in our translation. Furthermore, unlike [33], we define our translation in assumption-based argumentation, which means that meta-theoretical insights (e.g., complexity results [10] or results on properties of the non-monotonic consequence relations [9, 19, 20]), dialectical proof theories [12, 13], and different implementations [8, 32] can be directly used.

A representation of disjunctive logic programming by *abstract argumentation* is studied in [3]. In that translation, nodes in the argumentation framework correspond to single assumptions $\sim p$, as opposed to *sets* of such assumptions as in our translation. Because of this, the translation in [3] has to allow for attacks on *sets of nodes*, instead of just nodes, necessitating a generalization of Dung's abstract argumentation frameworks [11]. Since we work in assumption-based argumentation, where nodes in the argumentation framework correspond to sets of assumptions, the argumentation frameworks generated by our translation are normal abstract argumentation frameworks. This is important since in that way results and implementations for abstract argumentation frameworks can be straightforwardly used and applied.

Another related, but more distant line of work, is concerned with the integration of disjunctive reasoning in structured argumentation with defeasible

rules (see [1, 2]). We differ from this work both in the goal and the form of the knowledge bases.

In future work, we plan to generalize our results to other semantics for disjunctive logic programming, such as the disjunctive well-founded [5], extended well-founded [24], stationary [22], and possible world semantics [25]. Some of these semantics are based on ideas that are very similar to ideas underlying some well-known argumentation semantics. Likewise, For example, both the stationary semantics for disjunctive logic programming and the preferred semantics from abstract argumentation [11] can be characterized using three instead of two "truth values". Indeed, for normal logic programs the correspondence between the 3-valued stable models for normal logic programs [23] and complete labelings for ABA framework has been proven by [6, 7], which further supports the conjecture that the correspondence holds for disjunctive logic programs as well. Finally, we hope to extend our results to more expressive languages, such as epistemic [15] and parametrized logic programming [17].

References

1. Mathieu Beirlaen, Jesse Heyninck, and Christian Straßer. Reasoning by cases in structured argumentation. In *Proc. SAC'17*, pages 989–994. ACM, 2017.
2. Mathieu Beirlaen, Jesse Heyninck, and Christian Straßer. A critical assessment of Pollock's work on logic-based argumentation with suppositions. *Proc. NMR'18*, 20:63–72, 2018.
3. Alexander Bochman. Collective argumentation and disjunctive logic programming. *Journal of Logic and Computation*, 13(3):405–428, 2003.
4. Andrei Bondarenko, Phan Minh Dung, Robert Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63–101, 1997.
5. Stefan Brass and Jürgen Dix. Characterizations of the disjunctive well-founded semantics: confluent calculi and iterated gcwa. *Journal of Automated Reasoning*, 20(1-2):143–165, 1998.
6. Martin Caminada and Claudia Schulz. On the equivalence between assumption-based argumentation and logic programming. *Journal of Artificial Intelligence Research*, 60:779–825, 2017.
7. Martin Caminada and Claudia Schulz. On the equivalence between assumption-based argumentation and logic programming (extended abstract). In *Proc. IJCAI'18*, pages 5578–5582. ijcai.org, 2018.
8. Robert Craven, Francesca Toni, and Matthew Williams. Graph-based dispute derivations in assumption-based argumentation. In *Proc. TAFA'13*, LNCS No.8306, pages 46–62. Springer, Springer, 2013.
9. Kristijonas Čyras and Francesca Toni. Non-monotonic inference properties for assumption-based argumentation. In *International Workshop on Theory and Applications of Formal Argumentation*, pages 92–111. Springer, 2015.
10. Yannis Dimopoulos, Bernhard Nebel, and Francesca Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141(1-2):57–78, 2002.
11. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–358, 1995.

12. Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170(2):114–159, 2006.
13. Phan Minh Dung, Paolo Mancarella, and Francesca Toni. A dialectic procedure for sceptical, assumption-based argumentation. volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 145–156, 2006.
14. Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Transactions on Database Systems (TODS)*, 22(3):364–418, 1997.
15. Michael Gelfond. Logic programming and reasoning with incomplete information. *Annals of Mathematics and Artificial Intelligence*, 12(1-2):89–116, 1994.
16. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Prog. ICLP'88*, pages 1070–1080. MIT Press, 1988.
17. Ricardo Gonçalves and José Júlio Alferes. Parametrized logic programming. In *Proc. Jelia'10*, LNCS No.6341, pages 182–194. Springer, 2010.
18. Georg Gottlob. Complexity and expressive power of disjunctive logic programming (research overview). In *Proc. ICLP'94*, pages 23–42. Mit Press, 1994.
19. Jesse Heyninck and Ofer Arieli. On the semantics of simple contrapositive assumption-based argumentation frameworks. In *Proc. COMMA'18*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 9–20. IOS Press, 2018.
20. Jesse Heyninck and Ofer Arieli. Simple contrapositive assumption-based frameworks. In *Proc. LPNMR'19*, LNAI No.11481, pages 75–88. Springer, 2019.
21. John W. Lloyd. *Foundations of Logic Programming*. Springer, 1987.
22. Teodor Przymusinski. Stationary semantics for normal and disjunctive logic programs. In *DOOD*, volume 91. Citeseer, 1991.
23. Teodor C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.
24. Kenneth A Ross. A procedural semantics for well-founded negation in logic programs. *The Journal of Logic Programming*, 13(1):1–22, 1992.
25. Chiaki Sakama. Possible model semantics for disjunctive databases. In *Deductive and Object-Oriented Databases*, pages 369–383. Elsevier, 1990.
26. Claudia Schulz. Graphical representation of assumption-based argumentation. In *Proc. AAAI'15*, pages 4204–4205. AAAI Press, 2015.
27. Claudia Schulz, Ken Satoh, and Francesca Toni. Characterising and explaining inconsistency in logic programs. In *Proc. LPNMR'15*, LNCS No.9345, pages 467–479. Springer, 2015.
28. Claudia Schulz and Francesca Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59110, 2016.
29. Claudia Schulz and Francesca Toni. Complete assumption labellings. In *Proc. COMMA'2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 405–412. IOS Press, 2017.
30. Claudia Schulz and Francesca Toni. Labellings for assumption-based and abstract argumentation. *Journal of Approximate Reasoning*, 84:110–149, 2017.
31. Ezgi Su. *Extensions of equilibrium logic by modal concepts*. PhD thesis, IRIT-Institut de recherche en informatique de Toulouse, 2015.
32. Francesca Toni. A generalised framework for dispute derivations in assumption-based argumentation. *Artificial Intelligence*, 195:1–43, 2013.
33. Kewen Wang. Argumentation-based abduction in disjunctive logic programming. *The Journal of Logic Programming*, 45(1-3):105–141, 2000.