TPLP **25** (5): 896–923, 2025.
doi:10.1017/S1471068425100070 First published online 26 August 2025

Argumentative Characterizations of (Extended) Disjunctive Logic Programs

JESSE HEYNINCK

Department of Computer Science, Open Universiteit, Heerlen, Netherlands
University of Cape Town, Cape Town, South Africa
(*e-mail:* `jesse.heyninck@ou.nl`)

OFER ARIELI

School of Computer Science, Tel-Aviv Academic College, Tel-Aviv, Israel
(*e-mail:* `oarieli@mta.ac.il`)

submitted 13 June 2023; revised 27 June 2025; accepted 9 July 2025

Abstract

This paper continues an established line of research about the relations between argumentation theory, particularly assumption-based argumentation, and different kinds of logic programs. In particular, we extend known result of Bondarenko, Dung, Kowalski and Toni, and of Caminada and Schulz, by showing that assumption-based argumentation can represent not only normal logic programs, but also disjunctive logic programs under the stable model semantics. For this, we consider some inference rules for disjunction that the core logic of the argumentation frameworks should respect, and show the correspondence to the handling of disjunctions in the heads of the logic programs' rules.

Keywords: knowledge representation and nonmonotonic reasoning, theory

1. Introduction

Logic programming (LP) and formal argumentation are two primary disciplines involving knowledge representation and non-monotonic reasoning. Assumption-based argumentation (ABA, for short) (Bondarenko *et al*. 1997; Cyras *et al*. 2018) is a well-established branch in argumentation theory, aimed at providing coherent sets of formulas that admit other sets of formulas as their contraries, based on assumptions, contrariness operators, and rules in corresponding deductive systems. ABA was inspired by Dung's semantics for abstract argumentation frameworks (Dung 1995) and LP with its dialectical interpretation of the acceptability of negation-as-failure assumptions based on the "failure to prove the converse" (Przymusinski 1990). Thus, ABA can be viewed as an argumentative interpretation of LP semantics.

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 LP have been proven useful for explanation (Schulz *et al*. 2015; Schulz and Toni 2016), visualization of inferences in LP (Schulz 2015), and debugging of logic programs (Thevapalan *et al*. 2021). Among the works that relate LP and ABA we recall the one of Bondarenko *et al*. (1997), who were the first to show a correspondence between stable models (respectively, the well-founded model) of a normal logic program and stable extensions (respectively, the grounded extension) of the associated ABA framework (see Theorem 3.13, respectively Theorem 6.3 by Bondarenko *et al*. (1997), and the work of Caminada and Schulz 2017, 2018) that provides a one-to-one correspondence between the 3-valued stable models (Przymusinski 1990) (respectively, the regular models (You and Yuan 1994)) for normal logic programs, and complete labellings (respectively, preferred labellings) for ABA frameworks. A recent work (Wakaki, 2020) shows that answer sets of a (non-disjunctive) extended logic programs can be captured by stable extensions of the translated ABA frameworks. These works were restricted to logic programs, where only atoms or their (classical) negations are allowed in the head of the rules. Yet, a faithful modeling of real-world problems often requires to cope with *incomplete information*, which is not possible in the scope of such logic programs. This is a primary motivation in the introduction of *disjunctive logic programs*, where (classical) disjunctions are allowed in the heads of the rules and negations (sometimes called "negation-as failure" (Przymusinski 1990; Gelfond and Lifschitz 1991)) may occur in the rules' bodies. Reasoning with uncertainty is also a principle motivation behind *extended disjunctive logic programs*, where a classical negation is also permitted, both in the rules bodies and their heads. Indeed, disjunctive logic programs have been efficiently implemented and widely applied, and so become a key technology in knowledge representation (see, e.g., Su 2015), although under the usual complexity assumptions, they were shown to be strictly more expressive than normal logic programs (Gottlob 1994; Eiter *et al*. 1997).

Despite the equivalence between ABA semantics and the semantics of normal logic programs that has already been obtained in a number of works (Caminada and Schulz 2017, 2018), it is not obvious that such a correspondence carries on to disjunctive logic programs. In fact, there are some a-priori indications to the contrary, at least in some fundamental cases. For instance, the correspondence shown in Caminada and Schulz (2017) between 3-valued stable models of normal logic programs and complete extensions of ABA, breaks down when disjunctions may appear in the rules' heads. This is simply due to the fact that there are disjunctive logic programs without 3-valued stable models (Przymusinski 1991), while (flat) ABA frameworks always have complete extensions (Cyras *et al*. 2018). Therefore, in this paper we set out to generalize the argumentative characterization of LP for disjunctive logic programs and their extended variations. Naturally, this raises the primary research question of this paper, namely: *"Can existing translations from normal logic programming into assumption-based argumentation be extended, in some 'natural way', to disjunctive logic programs?"*

This question is answered affirmatively by showing that, at least as far as two-valued stable models are involved, disjunctive logic programs (and even extended disjunctive logic programs, allowing also strong negation in the rules) *can be* faithfully represented in

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

terms of assumption-based argumentation frameworks. For this, we incorporate the ideas in Arieli and Heyninck 2021, 2025) and Heyninck *et al.* (2024), generalizing standard ABA frameworks to propositional formulas (as the defeasible or strict assumption at hand), expressed and evaluated in propositional (Tarskian) logics. This allows to augment the underlying core logic of the ABA framework (based only on Modus Ponens, MP) with the inference rules Resolution (Res) and Reasoning by Cases (RBC), for handling disjunctive assertions, and so associate the stable extensions of such frameworks with the (2-valued) stable models of the corresponding disjunctive logic programs. On the other hand, we show that even when 3-valued stable models do exist, they might not correspond to complete models in the translated argumentation theory.

The structure of the paper. In the next section we review some basic notions behind assumption-based argumentation and disjunctive LP. In Section 3, which is the main part of the paper, we show how (the stable models of) the latter can be represented by (the stable extensions of) the former. The converse is discussed in Section 4. In Section 5, we provide some negative results, showing that this correspondence does not carry on to models and extensions that are not necessarily two-valued and stable. On the other hand, as shown in Section 6, our results are easily generalized to extended disjunctive logic programs, where a classical negation is also allowed, in addition to the negation-as-failure connective. In Section 7 (referring also to the supplementary material), we discuss some related work and conclude.

Relations with previous work. This paper is an updated and extended version of the paper in Heyninck and Arieli (2019), where we first revealed the relations between ABA and disjunctive LP. In particular, we give here full proofs for the main results of the paper (in Section 3.2), including proofs that were omitted in the conference paper, and revisions of other proofs in Heyninck and Arieli (2019), add further illustrations to the main concepts, consider also 3-valued semantics (and corresponding models) for LP, discuss extended logic programs, and refer to related work in more detail. Our work in Heyninck and Arieli (2019), which is extended here, closes a gap in the investigations of the connections between LP and formal argumentation, mainly in the presence of uncertain information. For this purpose, we extend the logical setting of the work in Bondarenko *et al.* (1997) and Caminada and Schulz (2017, 2018), consisting of Modus Ponens (MP) as the sole inference rule, by two additional inference rules, (Res) and (RBC) mentioned previously, which also allow to handle disjunctive information. This extended logical setting (with some adjustments) serves as a basis for a number of other recent follow-up papers, for example the ones by Wakaki (2022, 2024), discussing disjunctive information in ABA frameworks in relation to the way it is evaluated by semantics for extended disjunctive logic programs. The work in Wakaki (2024) shows, furthermore, that (a variation[1] of) our logical setting is also useful for linking ABA frameworks and a number of other formalisms for non-monotonic reasoning, including disjunctive default logic (Gelfond *et al.* 1991; Etherington and Crawford 1996), prioritized circumscription (Lifschitz 1987), and possible model semantics for extended disjunctive logic programs (Sakama and Inoue 2000). The motivation behind our work is therefore twofold: in the theoretical level it extends, by means of a simple translation, the known links between

[1] See Section 7 for some further details on this variation.

ABA and LP, and on the more practical side it vindicates the usefulness of argumentative and LP-based frameworks in handling not only inconsistent information, but also incomplete one.[2]

2. Preliminaries

We start with a brief review of the main concepts that are related to assumption-based argumentation (Section 2.1) and disjunctive logic programs (Section 2.2).

2.1. Assumption-based argumentation

We denote by $\mathcal{L}$ a propositional language. We shall assume that $\mathcal{L}$ contains a conjunction (denoted as usual in LP by a comma), disjunction $\vee$, implication $\rightarrow$, a negation operator $\sim$, and propositional constants F, T for falsity and truth. Atomic formulas in $\mathcal{L}$ are denoted by p, q, r (possibly indexed), literals (i.e., atomic formulas or their negation) are denoted by l (possibly indexed), compound formulas are denoted by ψ, ϕ, σ, and sets of formulas in $\mathcal{L}$ are denoted by $\Gamma, \Delta, \Theta, \Lambda$. When considering extended disjunctive logic programs, we shall assume that the language also contains another kind of negation, denoted $\neg$ (the exact meaning of each connective will be defined in the sequel). In what follows, we denote by $\sim\!\Gamma$ the set $\{\sim\!\gamma \mid \gamma \in \Gamma\}$. The powerset of Γ is denoted $\wp(\Gamma)$.

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 $\mathcal{L}$, satisfying the following properties:

- *Reflexivity: if $\psi \in \Gamma$ then $\Gamma \vdash \psi$.*
- *Monotonicity: if $\Gamma \vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash \psi$.*
- *Transitivity: if $\Gamma \vdash \psi$ and $\Gamma', \psi \vdash \phi$, then $\Gamma, \Gamma' \vdash \phi$.*

The next definition, adapted from Heyninck and Arieli (2020), generalizes the definition in Bondarenko *et al.* (1997) of assumption-based argumentation frameworks.

Definition 2.
An assumption-based framework (ABF, for short) is a tuple $\mathsf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$, where:

- *$\mathfrak{L} = \langle \mathcal{L}, \vdash \rangle$ is a propositional logic.*
- *Γ (the strict assumptions) and Λ (the candidate or defeasible assumptions) are distinct countable sets of $\mathcal{L}$-formulas, where the former is assumed to be $\vdash$-consistent (i,.e, $\Gamma \nvdash \mathsf{F}$) and the latter is assumed to be nonempty.*
- *$- : \Lambda \rightarrow \wp(\mathcal{L})$ is a contrariness operator, assigning a finite set of $\mathcal{L}$-formulas to every defeasible assumption in Λ.*

Note 1.
Unlike the setting of Bondarenko et al. (1997), an ABF may be based on any propositional logic $\mathfrak{L}$. Also, the strict as well as the candidate assumptions are formulas that may not

[2] A similar motivation stands behind the proposal of Gelfond *et al.* (1991) to extend Reiter's default logic (Reiter 1980) to disjunctive default logic in order to overcome some problems of the former in handling disjunctive information.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

be just atomic. Concerning the contrariness operator, note that it is not a connective of $\mathcal{L}$*, as it is restricted only to the defeasible assumptions.*

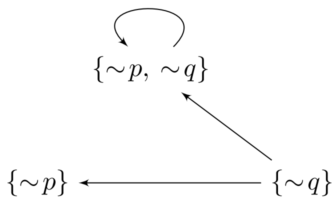
Defeasible assertions in an ABF may be attacked by counterarguments.

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

Example 1.
Let $\mathsf{ABF} = \langle \mathfrak{L}_{\mathsf{MP}}, \{\sim q \rightarrow p\}, \{\sim p, \sim q\}, - \rangle$*, where* $-\sim p = \{p\}$ *and* $-\sim q = \{q\}$*, and where* $\mathfrak{L}_{\mathsf{MP}}$ *consists of the sole inference rule Modus Ponens:*

$$[\mathsf{MP}]^{3} \; \frac{\phi_1, \ldots, \phi_n \rightarrow \psi \qquad \phi_1 \quad \phi_2 \cdots \phi_n}{\psi}$$

This gives rise to the following visual representation of ABF in terms of an attack diagram:



This diagram may be viewed as a directed graph whose nodes are sets of defeasible assumptions and where a directed arrow represents an attack of the set at the origin of the arrow on the set at the arrow's end. Note that, by MP, it hold that $\sim q, \sim q \rightarrow p \vdash p$, and so in our case every set that contains $\sim q$ attacks any set that contains $\sim p$.

The last definition gives rise to the following adaptation to ABFs of the usual semantics for abstract argumentation frameworks (Dung 1995).

Definition 4.
(*Bondarenko et al. 1997*) *Let* $\mathsf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ *be an assumption-based framework, and let* $\Delta \subseteq \Lambda$*. Then* Δ *is conflict-free if there is no* $\Delta' \subseteq \Delta$ *that attacks some* $\psi \in \Delta$*. We say that* Δ *is a stable extension of* ABF *if it is conflict-free, and attacks every* $\psi \in \Lambda \setminus \Delta$*. The set of stable extensions of* ABF *is denoted by* $\mathsf{Stb}(\mathsf{ABF})$*.*[4]

Example 2.
Consider again the assumption-based argumentation framework in Example 1. The sole stable extension of this framework is $\{\sim q\}$*.*

[3] As usual, the formulas above the fragment line are the rule's conditions and the formula below the line is the rule's conclusion.
[4] In many presentations of assumption-based argumentation, stable 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 4 below), closure of extensions is trivially satisfied in our case.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

2.2. Disjunctive logic programs

Definition 5.
(*Przymusinski 1991*) *A disjunctive logic program (DLP) π is a finite set of rules of the form*

$$(\star) q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \rightarrow p_1 \vee \ldots \vee p_n$$

where $m, k \geq 0$ and $n \geq 1$.[5] *We say that $p_1 \vee \ldots \vee p_n$ the head (conclusion) of the rule, and that $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k$ is the body (assumptions) of the rule.*

- *A logic program π is positive, if $k = 0$ for every rule in π (i.e., the negation-as-failure operator $\sim$ does not appear in π).*
- *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.*

We denote by $\mathcal{A}(\pi)$ the set of atomic formulas that appear in π.

Intuitively, the rule in $(\star)$ indicates that if q_i holds for every $1 \leq i \leq m$ and r_i is not provable for every $1 \leq i \leq k$, then either of the p_i's, for $1 \leq i \leq n$, should hold. 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.
Let M be set of atomic formulas, p, p_i, q_j atomic formulas, and l_j literals. We denote:

- $M \models p$ *if* $p \in M$,
- $M \models \sim p$ *if* $p \notin M$,
- $M \models p_1 \vee \ldots \vee p_n$ *if* $M \models p_i$ *for some* $1 \leq i \leq n$,
- $M \models l_1, \ldots, l_m$ *if* $M \models l_j$ *for every* $1 \leq j \leq m$,
- $M \models l_1, \ldots, l_m \rightarrow p_1 \vee \ldots \vee p_n$ *if 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* $M \models l_1, \ldots, l_m$*).*

Note that M may be viewed as an interpretation into $\{t, f\}$, where $M(p) = t$ if $p \in M$ (if $M \models p$). When $M \models \psi$, we say that M satisfies ψ. Given a logic program π, we denote by $M \models \pi$ that M satisfies every $\psi \in \pi$. In that case, we say that M is a model of π.

Thus, a model of a rule either falsifies at least one of the conjuncts in the rule's body, or validates at least one of the disjuncts in the rule's head. A particular family of models for disjunctive logic programs called *stable* (see Przymusinski 1991) is defined next.

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

[5] The assumption that $n \geq 1$ means that in this work we do not consider *constraints*, i.e., rules with empty heads (This is also the assumption of Caminada and Schulz in their transformation of normal logic programs to assumption-based argumentation (Caminada and Schulz 2017, 2018), which this work extends). The incorporation of constraints can be dealt with by interpreting constraints $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \rightarrow$ as rules of the form $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \rightarrow \mathsf{F}$. In turn, this requires to introduce *explosiveness assumptions*, expressing that from falsity any conclusion may be derived (see, e.g., Wakaki 2024). We leave this extension for future work.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

- *The Gelfond-Lifschitz reduct* (*Gelfond and Lifschitz 1988*) *of* π *with respect to* M *is the (positive) disjunctive logic program* π^M, *where* $q_1, \ldots, q_m \to p_1 \vee \ldots \vee p_n \in \pi^M$ *if 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 *if it is a* $\subseteq$*-minimal model of* π^M.

Example 3.
Consider the disjunctive logic program: $\pi_1 = \{{\sim}p \to q \vee r\}$. *Below are the different combinations of atoms in this case and their reducts. The two stable models of* π_1 *are marked by a gray background.*

i	M_i	$\pi_1^{M_i}$	i	M_i	$\pi_1^{M_i}$
1	$\emptyset$	$\{\to q \vee r\}$	5	$\{p, q\}$	$\emptyset$
2	$\{p\}$	$\emptyset$	6	$\{p, r\}$	$\emptyset$
3	$\{q\}$	$\{\to q \vee r\}$	7	$\{q, r\}$	$\{\to q \vee r\}$
4	$\{r\}$	$\{\to q \vee r\}$	8	$\{p, q, r\}$	$\emptyset$

Thus, $\{q\}$ *and* $\{r\}$ *are the (only) stable models of* π_1.

3. Representation of DLP by ABA

Given a disjunctive logic program π, we show a one-to-one correspondence between the stable models of π onto the stable extensions of an ABA framework that is induced from π. First, we describe the translation and then 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 the three inference rules Modus Ponens (MP), Resolution (Res) and Reasoning by Cases (RBC):

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

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

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

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

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

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, we denote $\Delta \vdash \phi$ if ϕ is either in Δ or is derivable from Δ using the inference rules above. In other words, $\Delta \vdash \phi$ if $\phi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$, 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]).

Note 2.
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 3.
Since the rule $\rightarrow \psi$ is identified with $\mathsf{T} \rightarrow \psi$, [MP] implicitly implies Reflexivity [Ref]: $\dfrac{\rightarrow \psi}{\psi}$.

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

Example 4.
Let $\pi_2 = \{\rightarrow p \vee q,\ p \rightarrow q,\ q \rightarrow p\}$. The attack diagram of the induced assumption-based argumentation framework $\mathsf{ABF}(\pi_2)$ is shown in Figure 1a.

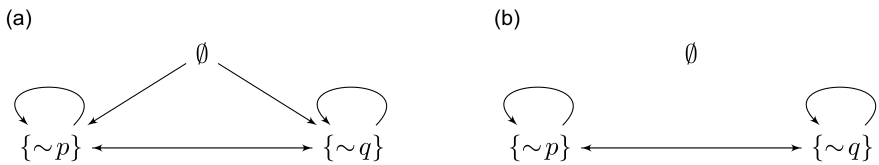


Fig. 1. Attack diagrams for Examples 4, 8c (left) and Example 8b (right).

In the notations of Definition 2, we have: $\Gamma = \pi_2$ and $\Lambda = \{{\sim}p, {\sim}q\}$. To see, for example, that the set $\{{\sim}p\}$ attacks itself, note that by [MP] on $\rightarrow p \vee q$ we conclude $\Gamma \vdash p \vee q$, and by [Res] it holds that $\Gamma, {\sim}p \vdash q$. Thus, since $q \rightarrow p \in \Gamma$, by [MP] we get $\Gamma, {\sim}p \vdash p$, namely: $\Gamma, {\sim}p \vdash -{\sim}p$. From similar reasons $\{{\sim}q\}$ attacks itself. For the attacks of $\emptyset$ on $\{{\sim}p\}$ and $\{{\sim}q\}$ we also need [RBC].

Note that in this case $\{p, q\}$ is the stable model of π_2 (Definition 7) and $\emptyset$ is the stable extension of $\mathsf{ABF}(\pi_2)$ (Definition 4).

Example 5.
Logic programs and their induced ABFs from earlier examples are the following:

a) *The assumption-based argumentation framework considered in Example 1 is induced from the logic program $\{{\sim}q \rightarrow p\}$.*[6]
b) *Figure 2 depicts (a fragment of) the graph of the assumption-based argumentation framework that is induced by the logic program $\pi_1 = \{{\sim}p \rightarrow q \vee r\}$ from Example 3. This framework has two stable extensions: $\mathcal{E}_1 = \{{\sim}p, {\sim}q, \}$ and $\mathcal{E}_2 = \{{\sim}p, {\sim}r, \}$.*

[6] In this case $\mathfrak{L}_{\mathsf{MP}}$ is used instead of the extended logic $\mathfrak{L}$, but as shown in Caminada and Schulz (2017, 2018), for normal logic programs this representation is sufficient for the correspondence between LPs and the induced ABA frameworks.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

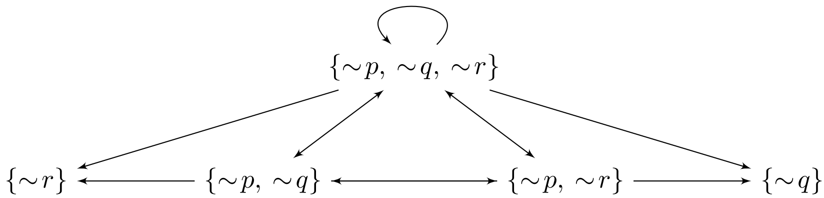


Fig. 2. Attack diagrams for Examples 3 and 5.

Note 4.
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* $\Gamma, \Delta \vdash \psi$*. This is shown in the following proposition:*

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

Proof.
Since Λ consists only of formulas of the form $\sim p$, we can restrict our attention to such formulas. 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 by Note 2, if $\Gamma, \Delta \vdash \sim p$, necessarily $\sim p \in \Delta$. $\square$

To relate the semantics of logic programs and their induced ABFs, we need the following notations:

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

- $\lfloor \sim\Theta \rfloor = \Theta$ (*Thus* $\lfloor \cdot \rfloor$ *eliminates the leading* $\sim$ *from the formulas in the set*)
- *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 6.
Consider again the program π_2 *of Example 4, and let* $\Delta = \sim\mathcal{A}(\pi_2) = \{\sim p, \sim q\}$*. Then* $\underline{\Delta} = \emptyset$ *and* $\overline{\emptyset} = \Delta$.

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

1. If Δ is a stable extension of $\mathsf{ABF}(\pi)$, 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 $\mathsf{ABF}(\pi)$.

Before showing these results, we first provide some examples and notes concerning related results.

Example 7.
Let's revisit our previous examples and see that the correspondence between the stable models of the logic programs and the stable extensions of the induced ABFs, indicated above, indeed holds in these examples:

a) *The stable models M_3 and M_4 of the logic program π_1 in Example 3 correspond to the stable extensions $\mathcal{E}_1$ and $\mathcal{E}_2$ of the assumption-based framework $\mathsf{ABF}(\pi_1)$ that is induced from π_1 according to Definition 8, which is considered in Example 5(b) and Figure 2. Indeed, $M_3 = \underline{\mathcal{E}_2}$ and $M_4 = \underline{\mathcal{E}_1}$ (Alternatively, $\mathcal{E}_1 = \overline{M_4}$ and $\mathcal{E}_2 = \overline{M_3}$).*
b) *As shown in Example 4, the correspondence between the stable model of π_2 and the stable extension of the framework $\mathsf{ABF}(\pi_2)$ that is induced from π_2 (see Figure 1a), is preserved.*

As indicated before this example, and as we show in Section 3.2 (Propositions 2 and 3), the two items above are not a coincidence.

Example 8.
As noted in the introduction, Caminada and Schulz (2017, 2018) consider the correspondence between ABA frameworks and normal logic programs. In our notations, the ABF that they associate with a normal logic program π is $\mathsf{ABF}_{\mathsf{Norm}}(\pi) = \langle \mathfrak{L}_{\mathsf{MP}}, \pi, \sim\mathcal{A}(\pi), - \rangle$ constructed as in Definition 8, except that $\mathfrak{L}_{\mathsf{MP}}$ is defined by Modus Ponens only.

a) *To see that $\mathsf{ABF}_{\mathsf{Norm}}$ is not adequate for disjunctive logic programs, consider the program $\pi_3 = \{\to p \vee q\}$. This program has two stable models: $\{p\}$ and $\{q\}$. However, the only stable extension of $\mathsf{ABF}_{\mathsf{Norm}}(\pi_3)$ is $\{\sim p, \sim q\}$. We can enforce $\{\sim p\}$ and $\{\sim q\}$ being stable models by requiring that $\pi_3 \cup \{\sim p\} \vdash q$ and $\pi_3 \cup \{\sim q\} \vdash p$. For this, we need the resolution rule* [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_2 from Example 4. Recall that $\{p, q\}$ is the sole stable model of π_2. The attack diagram of the ABF based on [MP] and [Res] is shown in Figure 1b, thus there is no stable extension in this case (However, if* [RBC] *is also available, we get the ABF depicted in Figure 1a, which, as indicated in Example 7b, is a faithful translation of π_2).*

3.2. Proof of correctness

The correctness of the translation follows from Propositions 2 and 3 below. First, we need some definitions and lemmas. In what follows $\mathfrak{L}$ denotes the logic defined in Section 3.1, and π is an arbitrary disjunctive logic program.

We start with the following soundness and completeness result.

Lemma 1.
Let Δ be a set of $\mathcal{L}$-formulas of the form $\sim p$ or $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \to p_1 \vee \ldots \vee p_n$. Then:

a) *If $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$ then $M \models \psi$ for every M such that $M \models \Delta$.*
b) *If $\psi = r_1 \vee \ldots \vee r_m$ and $M \models \psi$ for every M such that $M \models \Delta$, then $\psi \in \mathsf{Cn}_{\mathfrak{L}}(\Delta)$.*

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

Note 5.
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 $M \models \Delta$, we have $M \models \sim q_i$ $(i = 1, \ldots, n)$, 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 6, 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 6 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 6 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 6, $M \models l_1, \ldots, l_n$. On the other hand, by induction hypothesis again, $M \models l_1, \ldots, l_n \to \psi$. By Definition 6, $M \models \psi$.

We now turn to Part (b) of the lemma. Suppose 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 $\sim q_i \notin \Delta$.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

We now construct the model M' such that $M' \models \Delta$ and $M' \not\models r_1 \vee \ldots \vee r_m$: 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 $r_i \notin M'$ $(i = 1, \ldots, m)$, thus $M' \not\models r_1 \vee \ldots \vee r_m$. □

Lemma 2.
Let π be a logic program. For every $M, N \subseteq \mathcal{A}(\pi)$, if $N \setminus M \neq \emptyset$ then $N \not\models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$.

Proof.
Suppose that $N \setminus M \neq \emptyset$ and let $p \in N \setminus M$, Then $\sim p \in \overline{M}$ and thus $\sim p \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. Since $N \models p$, $N \not\models \sim p$, and so $N \not\models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. □

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

Proof.
Suppose that M is a minimal model of a logic program $\pi' \subseteq \pi^M$, and suppose towards a contradiction that there is some $N \subsetneq M$ such that $N \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. We show that N is a model of π' by showing that N is a model of π^M. Indeed, let $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k \in \pi$ such that $q_1, \ldots, q_m \notin M$. Then $\sim q_i \in \overline{M}$ for every $i = 1, \ldots, m$. Suppose furthermore that $p_1, \ldots, p_n \in N$. Then (since $N \subset M$), also $p_1, \ldots, p_n \in M$ and thus $M \vdash r_1 \vee \ldots \vee r_k$, which implies that $r_1 \vee \ldots \vee r_k \in \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. It follows that $r_i \in N$ for some $i = 1 \ldots, k$ (as we assumed that $N \models \mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$), and so N is a model of the rule $p_1, \ldots, p_n \rightarrow r_1 \vee \ldots \vee r_k \in \pi^M$. This means that N is a model of π', which contradicts the minimality of M. □

Lemma 4.
Let M be a stable model of π. Then $M = \min_{\subseteq}\{N \subseteq \mathcal{A}(\pi) \mid N \models \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \overline{M})\}$.

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 $\mathsf{D}_{\mathfrak{L}}(\psi)$.[7] 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 $M \models \psi$.

[7] Namely, $\mathsf{D}_{\mathfrak{L}}(\psi)$ is a finite sequence $\langle T_1, \ldots, T_n \rangle$ of proof tuples, where $T_n = \psi$, and for each $1 \leq i \leq n$ T_i is either an element of $\overline{M} \cup \pi$, or is obtained by an application of one of the inference rules of $\mathfrak{L}$ on proof tuples in $\{T_1, \ldots, T_{i-1}\}$. The size of $\mathsf{D}_{\mathfrak{L}}(\psi)$ is n.

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 $i = 1, \ldots, n$. By the induction hypothesis, $M \models p'_1 \vee \ldots \vee p'_m \vee q_1 \vee \ldots \vee q_n \vee p''_1 \vee \ldots \vee p''_k$. Thus, by Definition 6, $M \models p'_i$ for some $1 \leq i \leq m$, or $M \models p''_j$ for some $1 \leq j \leq k$. By Definition 6 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 6 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 $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \to r_1 \vee \ldots \vee r_k \in \pi$. 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, $p_1, \ldots, p_n \to r_1 \vee \ldots \vee r_k \in \pi^M$. Since M is a model of π^M, $M \models r_i$ for some $1 \leqslant i \leqslant k$. Thus, $M \models r_1 \vee \ldots \vee r_k$.

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 subset of $\mathcal{A}(\pi)$ which is a model of $\mathsf{Cn}_{\mathfrak{L}}(\overline{M} \cup \pi)$. □

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 $p \in M$. Thus $M \models p$. 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 so $M \models p$. □

Example 9.
Corollary 1 actually says that any atom p that is verified by a 2-valued sable model M of π, can be derived by assuming that all the assumptions not in M are false (and using the derivation rules together with the rules of the program). To illustrate this, take for example the program $\{\sim q \to p_1 \vee p_2\}$. This program has two stable models: $\{p_1\}$ and $\{p_2\}$. The result above then says that p_1 belongs to $\mathsf{Cn}_{\mathfrak{L}}(\overline{\{p_1\}} \cup \pi) = \mathsf{Cn}_{\mathfrak{L}}(\{\sim p_2, \sim q\} \cup \pi)$. Indeed, it holds that p_1 is derivable from $\{\sim p_2, \sim q\} \cup \pi$ using the rules $\{\sim q\} \cup \{\sim q \to p_1 \vee p_2\} \vdash p_1 \vee p_2$ and $\{p_1 \vee p_2, \sim p_2\} \vdash p_1$.

Lemma 5.
Let π be a disjunctive logic program, $\Delta \subseteq \sim\mathcal{A}(\pi)$, and $r_1 \vee \ldots \vee r_k \in$ $\mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$. If M is a model of $\pi^{\lfloor \Delta \rfloor}$ and $M \subseteq \lfloor \Delta \rfloor$, then $r_i \in M$ for some $1 \leqslant i \leqslant k$.

Proof.
We show by induction on the number of steps used in deriving $r_1 \vee \ldots \vee r_k$ from $\pi \cup \Delta$, that $r_1 \vee \ldots \vee r_k \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$ implies that $M \models r_1 \vee \ldots \vee r_k$. This means that $M \models r_i$ for some $1 \leqslant i \leqslant k$, and so $r_i \in M$ for that i.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

For the base step, since every element of Δ is of the form $\sim p$, we have that $r_1 \vee \ldots \vee r_k \notin \Delta$. Thus, $r_1 \vee \ldots \vee r_k$ is obtained from some rule $\sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k \in \pi$, where $\sim q_1, \ldots, \sim q_m \in \Delta$. In that case, since (1) $M \subseteq \underline{\Delta}$; (2) M is a model of $\pi^{\lfloor \Delta \rfloor}$; and (3) $\rightarrow r_1 \vee \ldots \vee r_k \in \pi^{\lfloor \Delta \rfloor}$, by Definition 7 there is a $1 \leqslant i \leqslant k$ such that $r_i \in M$.

For the induction step, suppose that the claim holds for every $r_1 \vee \ldots \vee r_k$ that is derived from Δ using n or less derivation steps. We consider three cases:

- $r_1 \vee \ldots \vee r_k$ is obtained by applying [MP] to the rule $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \rightarrow r_1 \vee \ldots \vee r_k$. In this case, $\sim q_1, \ldots, \sim q_m \in \Delta$ and so $q_i \notin \underline{\Delta}$ for any $1 \leqslant i \leqslant m$. It follows that $p_1, \ldots, p_n \rightarrow r_1 \vee \ldots \vee r_k \in \pi^{\lfloor \Delta \rfloor}$. Let now M be a model of $\pi^{\lfloor \Delta \rfloor}$ such that $M \subseteq \lfloor \Delta \rfloor$. By the induction hypothesis, $p_i \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$ implies that $p_i \in M$ (for every $1 \leqslant i \leqslant n$). Thus, since M is a model of $\pi^{\lfloor \Delta \rfloor}$, $r_i \in M$ for some $1 \leqslant i \leqslant k$.
- $r_1 \vee \ldots \vee r_k$ is obtained by applying [Res] from $r_1 \vee \ldots \vee r_k \vee r_{k+1} \vee \ldots r_n$ and $\sim r_{k+1}, \ldots, \sim r_n$. Suppose furthermore that $M \subseteq \lfloor \Delta \rfloor$ is a model of $\pi^{\lfloor \Delta \rfloor}$. By the induction hypothesis and since $r_1 \vee \ldots \vee r_k \vee r_{k+1} \vee \ldots r_n \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$, we have that $r_i \in M$ for some $1 \leqslant i \leqslant n$. Since $\sim r_i \in \Delta$ for $k+1 \leqslant i \leqslant n$ and $M \subseteq \lfloor \Delta \rfloor$, $r_i \notin M$ for every $k+1 \leqslant i \leqslant n$. This means that $r_i \in M$ for some $1 \leqslant i \leqslant k$.
- $r_1 \vee \ldots \vee r_k$ is obtained by applying [RBC], since $s_1 \vee \ldots \vee s_n \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$ and since $r_1 \vee \ldots \vee r_k \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \{\rightarrow s_i\} \cup \Delta)$ for every $1 \leqslant i \leqslant n$. By the induction hypothesis, (†): for every $1 \leqslant i \leqslant n$ and every model M' of $(\pi \cup \{\rightarrow s_i\})^{\lfloor \Delta \rfloor}$ such that $M' \subseteq \underline{\Delta}$, we have that $r_1 \vee \ldots \vee r_k \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \{\rightarrow s_i\} \cup \Delta)$ implies $r_i \in M'$ for some $1 \leqslant i \leqslant k$. Now, since $s_1 \vee \ldots \vee s_n \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \Delta)$, by the induction hypothesis this means that $s_j \in M$ for some $1 \leqslant j \leqslant n$ for every model M of $\pi^{\lfloor \Delta \rfloor}$. In other words, M is a model of $\pi^{\lfloor \Delta \rfloor}$ iff it is a model of $(\pi \cup \{\rightarrow s_j\})^{\lfloor \Delta \rfloor} = \pi^{\lfloor \Delta \rfloor} \cup \{\rightarrow s_j\}$ for some $1 \leqslant j \leqslant n$. By (†), then, $r_i \in M$ for some $1 \leqslant i \leqslant k$.

□

By the last lemma, in case that $k = 1$, we therefore have:

Corollary 2.
Let π be a disjunctive logic program, $\Delta \subseteq \sim\mathcal{A}(\pi)$ 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 $\mathsf{ABF}(\pi)$.

Proof.
Suppose that M is a stable model of π. We show first that $\overline{M}$ is conflict-free in $\mathsf{ABF}(\pi)$. 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 Corollary 2, $p \in \mathsf{Cn}_{\mathfrak{L}}(\pi \cup \overline{M})$ implies that $p \in M$, a contradiction to $p \notin M$

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 $\pi, \overline{M} \vdash p$ for every $p \in M$. This follows from Corollary 1. □

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

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

Proof.
We first show that $\underline{\mathcal{E}}$ is a model of $\pi^{\underline{\mathcal{E}}}$. Indeed, let $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \to r_1 \vee \ldots \vee r_k \in \pi$. If $q_j \in \underline{\mathcal{E}}$ for some $1 \le j \le m$ we are done: the rule is satisfied by $\underline{\mathcal{E}}$. Otherwise, $q_1, \ldots, q_m \notin \underline{\mathcal{E}}$, and so $p_1, \ldots, p_n \to r_1 \vee \ldots \vee r_k \in \pi^{\underline{\mathcal{E}}}$. Again, if $p_j \notin \underline{\mathcal{E}}$ for some $1 \le j \le n$ we are done, as the rule is satisfied by $\underline{\mathcal{E}}$. Thus, $p_1, \ldots, p_n \in \underline{\mathcal{E}}$. 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 a stable extension of $\mathsf{ABF}(\pi)$, this means that $\pi, \mathcal{E} \vdash r_1 \vee \ldots \vee r_k$. 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 $\sim r_i \notin \mathcal{E}$, thus $r_i \in \underline{\mathcal{E}}$, which means that $\underline{\mathcal{E}}$ satisfies $p_1, \ldots, p_n \sim q_1, \ldots, \sim q_m \to r_1 \vee \ldots \vee r_k$.

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 Corollary 2, any model of $\pi^{\underline{\mathcal{E}}}$ satisfies p, a contradiction to $p \notin M$. □

4. Representation of ABA by DLP

The main body of literature on ABA frameworks is concentrated on languages that consist solely of formulas of the form $p_1, \ldots, p_n \to p$ (where $p, p_1, \ldots, p_n$ are atomic formulas). As noted previously, 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 Caminada and Schulz 2017, 2018). To the best of our knowledge, the more complicated classes of ABA frameworks that are considered in this paper for characterizing disjunctive LP (and which are based on a logic allowing to reason with disjunctive rules of the form $p_1, \ldots, p_n, \sim q_1, \ldots, \sim q_m \to r_1 \vee \ldots \vee r_k$) 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 based on an arbitrary propositional logic (as defined and studied in Arieli and Heyninck 2021, 2025) and Heyninck and Arieli (2020)) can be translated in a class of logic programs, probably more general than disjunctive ones. This is a subject for a future work.

5. Beyond two-valued stable semantics

So far we have discussed two-valued stable models for (disjunctive) logic programs and their correspondence to the stable extensions of the induced ABA frameworks. In this section we check whether similar correspondence may be established between other semantics of logic programs, such as three-valued stable semantics, and other types of argumentative extensions.

For switching to a three-valued semantics we add to the two propositional constants T and F, representing in the language $\mathcal{L}$ the Boolean values, a third elements, denoted U, which intuitively represents uncertainty. These constants correspond, respectively, to the three truth values of the underlying semantics, t, f, u, representing truth, falsity and uncertainty. As in Kleene's semantics (Kleene 1950), these values may be arranged in two orders:

- a total order $\leq_t$, representing difference in the amount of *truth* that each value represent, in which $\mathsf{f} <_t \mathsf{u} <_t \mathsf{t}$, and
- a partial order $\leq_i$, representing difference in the amount of *information* that each value depicts, in which $\mathsf{u} <_i \mathsf{f}$ and $\mathsf{u} <_i \mathsf{t}$.

In what follows, we denote by $-$ the $\leq_t$-involution, namely $-\mathsf{f} = \mathsf{t}$, $-\mathsf{t} = \mathsf{f}$, and $-\ \mathsf{u} = \mathsf{u}$.

Accordingly, we extend the notion of interpretations of a logic program π from subsets of $\mathcal{A}(\pi)$ (as in Definition 6) to *pairs* of subsets of $\mathcal{A}(\pi)$:

Definition 10.
A three-valued interpretation of a program π is a pair $M = (x, y)$, where $x \subseteq \mathcal{A}(\pi)$ is the set of the atoms that are assigned the value t *and $y \subseteq \mathcal{A}(\pi)$ is the set of atoms assigned a value in* $\{\mathsf{t},\ \mathsf{u}\}$.

Clearly, in every three-valued interpretation it holds that $x \subseteq y$, and the two-valued interpretations (onto $\{\mathsf{t}, \mathsf{f}\}$) of the previous sections may be viewed as three-valued interpretations in which $x = y$.

Interpretations may be compared by two order relations, generalized from the order relations among the truth values:

1. the *truth order* $\leq_t$, where $(x_1, y_1) \leq_t (x_2, y_2)$ iff $x_1 \subseteq x_2$ and $y_1 \subseteq y_2$, and
2. the *information order* $\leq_i$, where $(x_1, y_1) \leq_i (x_2, y_2)$ iff $x_1 \subseteq x_2$ and $y_2 \subseteq y_1$.

The information order represents differences in the "precisions" of the interpretations. Thus, the components of higher values according to this order represent tighter evaluations. The truth order represents increased "positive" evaluations. Truth assignments to complex formulas are then recursively defined as in the next definition (see also Przymusinski 1991):

Definition 11.
The truth assignments of a 3-valued interpretation (x, y) are defined as follows:

- $(x,y)(p) = \begin{cases} \mathsf{t} & \text{if } p \in x \text{ and } p \in y, \\ \mathsf{u} & \text{if } p \notin x \text{ and } p \in y, \\ \mathsf{f} & \text{if } p \notin x \text{ and } p \notin y, \end{cases}$
- $(x,y)(\sim\phi) = -(x,y)(\phi)$,
- $(x,y)(\psi \wedge \phi) = lub_{\leq_t}\{(x,y)(\phi), (x,y)(\psi)\}$,
- $(x,y)(\psi \vee \phi) = glb_{\leq_t}\{(x,y)(\phi), (x,y)(\psi)\}$.

The next definition is the three-valued counterpart of Definitions 6 and 7.

Definition 12.
Let π be a logic program and let (x, y) be a 3-valued interpretation of π.

- *(x, y) is a (3–valued) model of π, if for every $\phi \to \psi \in \pi$, $(x,y)(\phi) \leq_t (x,y)(\psi)$.*

- *The Gelfond-Lifschitz transformation (Gelfond and Lifschitz 1991) of a disjunctively normal program π with respect to a 3-valued interpretation (x, y), denoted $\pi^{(x,y)}$, is the positive program obtained by replacing in every rule in π of the form $q_1, \ldots, q_m, \sim r_1, \ldots, \sim r_k \to p_1 \vee \ldots \vee p_n$, any negated literal $\sim r_i$ $(1 \leq i \leq k)$ by: F if $(x, y)(r_i) = \mathsf{t}$, T if $(x, y)(r_i) = \mathsf{f}$, and U if $(x, y)(r_i) = \mathsf{u}$. That is, replacing $\sim r_i$ by the propositional constant that corresponds to $(x, y)(\sim r_i)$.*
- *An interpretation (x, y) is a 3-valued stable model of π* (Przymusinski 1991), *if it is a $\leq_t$-minimal model of $\pi^{(x,y)}$.*

There is a host of other semantics for both normal and disjunctive logic programs, including several semantics that refine the three-valued stable models by, for example, selecting only the $\leq_i$-*minimal* three-valued stable models (as is done for normal logic programs in the so-called *3-valued well-founded models*), or taking the $\leq_i$-*maximal* three-valued stable models. The latter, called *M-stable models* (or *L-stable models*), were introduced for DLPs by Sacca and Zaniolo Sacca and Zaniolo (1990, 1991, 1997). For normal logic programs, M-stable models coincide with *regular models*, as defined by You and Yuan (1994, Definition 8). However, for disjunctive logic programs, M-stable models do *not* coincide with regular models, since the latter are guaranteed to exist (see [You and Yuan, 1994, Proposition 5.1]) whereas the former are not (as three-valued stable models in general might not exist for DLPs, see Przymusinski 1991). A detailed overview of the relations between the above-mentioned semantics, as well as other semantics for DLPs, is given by Eiter, Leone and Saccá (1997).

Likewise, there are many proposals in the literature for extending the well-founded semantics from normal logic programs to disjunctive logic programs. We refer, for example, to Brass and Dix (1995), Seipel (1998), Alcantara *et al*. (2005), Wang and Zhou (2005), and Knorr and Hitzler (2007), for some examples. However, there is no consensus about which are the most suitable ones, or even what should be the criteria for comparing them (see Alcantara *et al*. 2005; Wang and Zhou 2005), including existence and uniqueness. In what follows, for considering the correspondence with ABA frameworks, we restrict our attention to the 3-valued stable models, leaving the investigations of other semantics such as the ones discussed above to future work.

Note 6.
As shown in Caminada and Schulz (2017, 2018), when π is a normal logic program, its 3-valued models that are defined above have some counterparts in terms of the induced assumption-based argumentation framework $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$, described in Example 8. To recall these results, we first need to extend the notions in Definition 4 with the following argumentative concepts:

Definition 13.
Let $\mathsf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ be an assumption-based framework and let $\Delta \subseteq \Lambda$. The set of all the formulas in Λ that are attacked by Δ is denoted by Δ^+, and the set of all the formulas in Λ that are defended by Δ (namely, the formulas whose attackers are in Δ^+) is denoted $\mathsf{Def}(\Delta)$. *We say that Δ is a complete extension of* ABF, *if it is conflict-free (Definition 4) and* $\mathsf{Def}(\Delta) = \Delta$ *(namely, Δ defends exactly its own elements). A $\subseteq$-maximally complete*

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

extension of ABF *is called preferred extension of* ABF, *and a* $\subseteq$*-minimally complete extension of* ABF *is called grounded extension of* ABF.[8]

In Caminada and Schulz (2017, 2018), it was shown that the 3-valued stable models of a normal logic program π *correspond to the complete extensions of* $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$, *the M-stable models of* π *correspond to preferred extensions of* $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$, *and the well-founded model of* π *corresponds to the grounded extension of* $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$. *Using the notations in Definition 9, these results can be expressed as follows:*

- *If* (x, y) *is a 3-valued stable model of a normal logic program* π, *then* $\overline{y}$ *is a complete extension of* $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$.
- *If* $\mathcal{E}$ *is a complete extension of* $\mathsf{ABF}_{\mathsf{Norm}}(\pi)$, *where* π *is a normal logic program, then* $(\lfloor \mathcal{E}^+ \rfloor, \underline{\mathcal{E}})$ *is a 3-valued stable model of* π.

We note that Caminada and Schulz (2017, 2018) *also establish a connection between the well-founded models [respectively, the M-stable models] for normal logic programs and the grounded extensions [respectively, the preferred extensions] of assumption-based argumentation frameworks.*[9] *In other words, true atoms* $p \in x$ *on the LP-side correspond to attacked negations of these atoms* $\sim p \in \overline{y}^+$ *on the ABA-side, whereas false atoms* $p \in \mathcal{A}(\pi) \setminus y$ *correspond to accepted negations of these atoms* $\sim p \in \overline{y}$ *on the ABA-side. We illustrate this with an example:*

Example 10.
Consider the normal logic program $\pi_4 = \{\sim q \to q,\ \sim r \to s\}$. *This program has a single 3-valued stable model* $(\{s\}, \{s, q\})$, *in which* s *is true,* q *is undecided and* r *is false. The induced assumption-based framework is* $\mathsf{ABF}_{\mathsf{Norm}}(\pi_4) = \langle \mathfrak{L}_{\mathsf{MP}}, \pi_4, \{\sim q, \sim r, \sim s\}, - \rangle$. *(A fragment of) its attack diagram is the following:*

$$\begin{tikzcd} \{\sim s\} & \{\sim r\} \arrow[l] & \{\sim q\} \arrow[loop above] \end{tikzcd}$$

$\mathsf{ABF}_{\mathsf{Norm}}(\pi_4)$ *has a single preferred extension:* $\{\sim r\}$, *which attacks* $\{\sim s\}$. *Intuitively (also according to the 3-valued labeling semantics for this case, see Baroni et al. 2018), this makes* $\sim r$ *true (i.e.,* r *is false),* $\sim s$ *false (and so* s *is true), and* $\sim q$ *undecided (since it is neither accepted nor attacked by* $\sim r$*).*[10] *This corresponds to the 3-valued stable model of* π_4, *which is in-line with the results stated above.*

The correspondence described above, between 3-valued stable models of logic programs and extensions of the induced argumentation frameworks, does *not* carry on to disjunctive

[8] Note that, using the notations in Definition 13, $\Delta \subseteq \Lambda$ is a stable extension of ABF according to Definition 4, iff $\Delta \cap \Delta^+ = \emptyset$ (namely, Δ is conflict-free) and $\Delta \cup \Delta^+ = \Lambda$.

[9] In Caminada and Schulz (2017, 2018), the M-stable models are called regular models, but as Caminada and Schulz restrict the attention to normal logic programs, these are the same as M-stable models (as we have noted before). The comparison of preferred extensions with M-stable models of normal logic programs is justified by the similarity of their intuitions, namely minimization of undecidability.

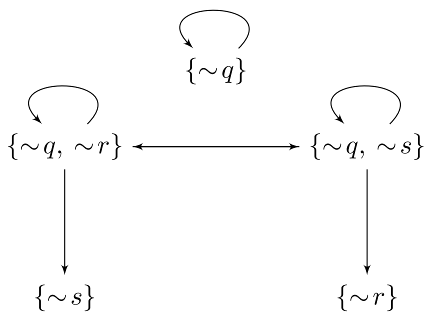
[10] In general, an extension $\mathcal{E} \subseteq \Lambda$ may be associated with a 3-valued 'labeling' in which the element of $\mathcal{E}$ are accepted, the elements that are attacked by $\mathcal{E}$ (those in $\mathcal{E}^+$) are rejected, and the other elements are undecided (see, e.g., Baroni *et al.* 2018). In the notations of Definition 11, then, this corresponds to the 3-valued interpretation $(\mathcal{E}, \Lambda - \mathcal{E}^+)$.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

logic programs. This immediately follows from the fact that, while a 3-valued stable model (respectively: a $\leq_i$-minimal 3-valued stable model, a $\leq_i$-maximal 3-valued stable model) of a DLP are not guaranteed to exist (Reiter 1978),[11] complete (respectively: grounded, preferred) extensions of the induced ABF always exist. Clearly, this consideration is independent of the used translation. However, the next example shows that, moreover, *even when 3-valued stable models of a DLP do exist, the correspondence to the related ABF extensions ceases to hold under our translation.*

Example 11.
Consider the disjunctive logic program $\pi_5 = \{\sim q \to q,\ \ q \to r \vee s\}$. *The 3-valued stable models of* π_5 *are* $(\emptyset, \{q, r\})$ *and* $(\emptyset, \{q, s\})$.

The assumption-based argumentation that is induced by π_5 *is* $\mathsf{ABF}(\pi_5) = \langle \mathfrak{L}, \pi_5, \{\sim q, \sim r, \sim s\}, - \rangle$. *This results in the following (fragment of the) attack diagram:*



The unique complete set of assumptions in this case is $\emptyset$. *This set does not attack any assumption, thus corresponding to* $(\emptyset, \{q, r, s\})$. *We thus see here that there is a discrepancy between the three-valued stable models of* π_5 *and complete extensions of* $\mathsf{ABF}(\pi_5)$.

Note 7.
A closer inspection of the last example may reveal some of the reasons for the inadequacy of our translation under 3-valued semantics: In the 3-valued setting, the interpretations $(\emptyset, \{q, r\})$, $(\emptyset, \{q, s\})$, *and* $(\emptyset, \{q, r, s\})$, *are all models of* π_5, *whereas only the first two are stable. This means that in this setting,* f*-assignments are not minimized (indeed, in either of the first two models, the set of the atoms that are assigned* f *properly contains the corresponding set of the other model). To reflect such a minimization in the induced argumentation framework, additional rules are required. For instance, in the last example, if* $r \vee s$ *is undecided (since* $\sim q$ *is undecided), one needs a rule that would bring about mutual attacks between* $\sim r$ *and* $\sim s$.

[11] Indeed, Przymusinski (1991) shows that the DLP $\pi = \{\to \sim p \vee \sim q \vee \sim r,\ \sim r \to p,\ \sim p \to q,\ \sim q \to r\}$ does not have a 3-valued stable model. There are other semantics that are guaranteed to exist for this program, such as the regular models (You and Yuan 1994) or stationary semantics (Przymusinski 1990). We will see below that for these semantics, the correspondence breaks down as well.

We conclude this section by noting that some other semantics for disjunctive logic programs have been suggested in the literature, for example stationary semantics (Przymusinski 1991), (weakly) supported semantics (Brass and Dix 1995; Heyninck *et al.* 2024), determining inference semantics (Shen and Eiter 2019), semi-equilibrium semantics (Amendola *et al.* 2016; Heyninck and Bogaerts 2023), and several variants of the well-founded semantics (Amendola *et al.* 2016; Heyninck and Bogaerts 2023). Whether these semantics can be characterized by argumentation frameworks is a subject for future work.

6. Extended disjunctive logic programs

Intuitively, the connective $\sim$ may be understood as representing "negation as failure" (to prove the converse) (Clark 1978). This kind of negation is also known as "weak negation." Extended (disjunctive) logic programs are obtained by introducing another negation connective, $\neg$, which acts as an explicit (strong, classical) negation, and allowing $\neg$-literals (namely, atomic formulas or their explicit negation) instead of the atoms in the rules ($\star$) of Definition 5. Formally:

Definition 14.
An extended disjunctive logic program π *(Przymusinski 1991) is a finite set of rules of the form*

$$(\star\star) \quad l_1, \ldots, l_m, {\sim} l_{m+1}, \ldots, {\sim} l_{m+k} \rightarrow l_{m+k+1} \vee \ldots \vee l_{m+k+n}$$

where $m, k \geq 0$, $n \geq 1$, *and each* l_i $(1 \leq i \leq m+k+n)$ *is a* $\neg$*-literal, that is:* $l_i \in \{p_i, \neg p_i\}$ *for some* $p_i \in \mathcal{A}(\pi)$.

The semantics of extended disjunctive logic programs is defined by reduction to disjunctive logic programs, using a standard method in LP that views a literal of the form $\neg p$ as a strangely written atomic formula. Under this view, rules of the form ($\star\star$) may be treated just as if they are of the form ($\star$). Relations between atomic formulas that correspond to positive and negative occurrences of the same atomic assertion (namely, without or with a leading explicit negation, respectively) are made on the semantic level. Indeed, while the semantics of disjunctive logic programs is two-valued (where an atom p is verified in a model M if $p \in M$ and is falsified otherwise), the semantics of extended disjunctive programs is *four-valued*. To see this, given a rule r of the form ($\star\star$), replace every occurrence of p by p^+ and every occurrence of $\neg p$ by p^-. Denote by $r^\pm$ by the resulting rule (and by $\pi^\pm$ the corresponding program). Suppose that M is a model of $r^\pm$. Then:

- p is *true* in M, if $p^+ \in M$ and $p^- \notin M$,
- p is *false* in M, if $p^+ \notin M$ and $p^- \in M$,
- p is *contradictory* in M, if $p^+ \in M$ and $p^- \in M$,
- p is *undecided* in M, if $p^+ \notin M$ and $p^- \notin M$.

The next example illustrates this distinction between the semantics of disjunctive logic programs and extended disjunctive logic programs, and shows how this is reflected by the stable extensions of the corresponding ABFs.

Example 12.

1. *Let* $\pi_6 = \{\sim p \to \neg p\}$. *This program depicts a '"closed world assumption" (Reiter 1978) regarding p: Any statement that is true is also known to be true, therefore, if p is not known to be true, it must be false. The conversion of* π_6 *to a normal logic program is* $\pi_6^{\pm} = \{\sim p^+ \to p^-\}$, *the stable model is* $\{p^-\}$, *which in turn is associated with the the two-valued stable model of* π_6, *in which p is false (see above). In the corresponding assumption-based framework* $\mathsf{ABF}(\pi_6) = \langle \mathfrak{L}, \pi_6^{\pm}, \{\sim p^+, \sim p^-\}, -\rangle$ *we have that* $\{\sim p^+\}$ *attacks* $\{\sim p^-\}$, *and so* $\{\sim p^+\}$ *is the unique stable model in this case (as indeed indicated in Proposition 2).*
   Note that if π_6 *is extended with* $\{p\}$, *the situation is reversed: the sole stable extension of* $(\pi_6 \cup \{p\})^{\pm}$ *is* $\{p^+\}$, *and in the corresponding ABF we have that* $\{\sim p^+\}$ *is attacked by* $\{\sim p^-\}$. *Thus, the latter is the unique stable extension in this case, as expected.*
2. *Consider the following extended disjunctive logic program:*

$$\pi_7 = \{\to \neg p \vee \neg q,\ \sim\neg p \to p,\ \sim\neg q \to q,\ p \to s,\ q \to s\}.$$

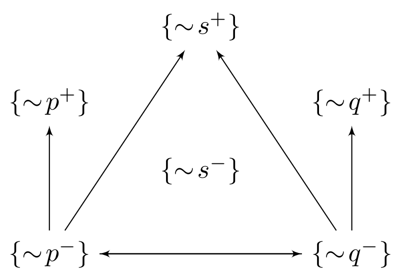
   We obtain the following translated disjunctive logic program:

$$\pi_7^{\pm} = \{\to p^- \vee q^-,\ \sim p^- \to p^+,\ \sim q^- \to q^+,\ p^+ \to s^+,\ q^+ \to s^+\}.$$

   The corresponding assumption-based framework is:

$$\mathsf{ABF}(\pi_7^{\pm}) = \langle \mathfrak{L}, \pi_7^{\pm}, \{\sim p^+, \sim p^-, \sim q^+, \sim q^-, \sim s^+, \sim s^-\}, -\rangle.$$

   A fragment of the resulting attack diagram is given below:



   There are two stable sets of assumptions, $\{\sim q^-, \sim p^+, \sim s^-\}$ *and* $\{\sim p^-, \sim q^+, \sim s^-\}$, *corresponding (respectively) to the stable models of* $\pi_7^{\pm}$ $\{p^-, q^+, s^+\}$ *and* $\{q^-, p^+, s^+\}$, *which in turn correspond (respectively) to the two stable models of* π_7 $\{\neg p, q, s\}$ *and* $\{\neg q, p, s\}$.
3. *Let* $\pi_8 = \{\to p,\ \to q,\ \to r,\ \to \neg p \vee \neg q\}$. *This logic program reflects an inconsistent information regarding* $\{p, q\}$. *The translated program is* $\pi_8^{\pm} = \{\to p^+,\ \to q^+,\ \to r^+,\ \to p^- \vee q^-\}$, *and the induce induced ABFs is* $\mathsf{ABF}(\pi_8^{\pm}) = \langle \mathfrak{L}, \pi_8^{\pm}, \{\sim p^+, \sim p^-, \sim q^+, \sim q^-, \sim r^+\}, -\rangle$. *Part of the attack diagram in this case is the following:*

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

$$
\begin{tikzcd}
 & \{\} \arrow[dl] \arrow[d] \arrow[dr] & \\
\{\sim p^+\} & \{\sim q^+\} & \{\sim r^+\}
\end{tikzcd}
$$

$$
\{\sim p^-\} \longleftrightarrow \{\sim q^-\}
$$

The stable extensions of $\mathsf{ABF}(\pi_8^{\pm})$ *are therefore* $\{\sim q^-\}$ *and* $\{\sim p^-\}$*, corresponding to the stable models* $\{p^+, p^-, q^+, r^+\}$ *and* $\{q^+, q^-, p^+, r^+\}$ *of* $\pi_8^{\pm}$*. In both models, inconsistency is '"localized": p is the only contradictory atoms in the former model, and q is the only contradictory atom in the latter.*

We observe that in Item 3 of the last example the two stable models of the logic program $\pi_8^{\pm}$ coincide with the *paraconsistent stable models* of π_8 according to Sakama and Inoue (1995). Next, we show that this is not a coincidence. First, we recall the definition of the paraconsistent stable semantics in Sakama and Inoue (1995):

Definition 15.
Let M *be a set of* $\neg$*-literals.*

- *The satisfiability relation* $\models$ *is defined as follows (cf. Definition 6, where* M *is a set of atoms):*
  - $M \models l$ *iff* $l \in M$ *(for any* $\neg$*-literal* l*)*
  - $M \models l_1 \vee \ldots \vee l_n$ *iff* $M \models l_i$ *for some* $1 \leq i \leq n$ *(for any set of* $\neg$*-literals* $\{l_1, \ldots, l_n\}$*),*
  - $M \models l_1, \ldots, l_m, \sim l_{m+1}, \ldots, \sim l_{m+k} \to l_{m+k+1} \vee \ldots \vee l_{m+k+n}$ *iff whenever* $M \models l_i$ *for every* $1 \leq i \leq m$ *and* $M \not\models l_{m+i}$ *for every* $1 \leq i \leq k$*, then* $M \models l_{m+k+1} \vee \ldots \vee l_{m+k+n}$.

  We say that M *is a model of an extended disjunctive logic program* π *iff* $M \models r$ *for every* $r \in \pi$*.*
- *The (2-valued) reducts of extended DLPs can be constructed just as in the case without strong negation (cf. Definition 7), that is:* π^M *consists of all the rules* $l_1, \ldots, l_m, \to l_{m+k+1} \vee \ldots \vee l_{m+k+n}$ *such that* $l_1, \ldots, l_m, \sim l_{m+1}, \ldots, \sim l_{m+k} \to l_{m+k+1} \vee \ldots \vee l_{m+k+n} \in \pi$ *and* $l_{m+i} \notin M$ *for every* $1 \leq i \leq k$*.*
- *A set of literals* M *is a paraconsistent stable model of* π*, iff* M *is a* $\subseteq$*-minimal model of* π^M*.*[12]

We now define a translation δ from $\pm$-atomic formulas to $\neg$-literals by: $\delta(p^-) = \neg p$, $\delta(p^+) = p$, and denote $\delta(\Theta) = \{\delta(p^{\pm}) \mid p^{\pm} \in \Theta\}$. Likewise, we define: $\delta^{-1}(\neg p) = p^-$, $\delta^{-1}(p) = p^+$, and denote $\delta^{-1}(\Theta) = \{\delta^{-1}(l) \mid l \in \Theta\}$.

[12] These notions are identical to the corresponding notions of Sakama and Inoue (2000), adapted to our notations.

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

The following correspondence result is now easily obtained:

Proposition 4.
let π be an extended disjunctive logic program. Then:

a) *If $\mathcal{E}$ is a stable extension of $\mathsf{ABF}(\pi^{\pm})$, then $\delta(\underline{\mathcal{E}})$ is a paraconsistent stable model of π.*
b) *If M is a paraconsistent stable model of π, then $\overline{\delta^{-1}(M)}$ is a stable extension of $\mathsf{ABF}(\pi^{\pm})$.*

Proof.
The proof immediately follows from the following two observations:

- A set M of $\neg$-literals is a paraconsistent stable model of π iff $\delta(M)$ is a stable model of $\pi^{\pm}$. (This is straightforward from the definition of a paraconsistent stable model. In particular, there are no rules governing interactions between $\neg p$ and p for any atom p, thus p and $\neg p$ behave as unrelated literals in this semantics.)
- If $\mathcal{E}$ is a stable extension of $\mathsf{ABF}(\pi^{\pm})$, then $\underline{\mathcal{E}}$ is a stable model of $\pi^{\pm}$, and vice-versa: if M is a stable model of $\pi^{\pm}$ then $\overline{M}$ is a stable extension of $\mathsf{ABF}(\pi^{\pm})$. (This follows from Propositions 2 and 3.)

□

By the last proposition it follows that our approach allows to capture the paraconsistent stable semantics for extended disjunctive LP introduced in Sakama and Inoue (1995) by a rather simple revision of the language. We refer also to Wakaki (2022, 2024), where a further study of argumentative representations of extended disjunctive LP has been undertaken. The latter requires to extend the set of rules in the translation. Thus, depending on the application at hand, each approach has its benefits and downsides.

7. Related work and conclusion

This work generalizes translations from LP into assumption-based argumentation to cover also (extended) disjunctive logic programs. Our framework was introduced in Heyninck and Arieli (2019) and then generalized in Wakaki (2022), where Wakaki shows a semantic correspondence between generalized assumption-based argumentation (Definition 2), based on the logic in Section 3.1, and extended disjunctive LP. In Wakaki (2024), this correspondence is carried on to further formalisms for non-monotonic reasoning, including disjunctive default theory, parallel circumscription, and prioritized circumscription. Even though a different symbol for disjunction (denoted by $|$) is used in Wakaki 2022, 2024), the semantics of this connective is the same as the one we consider here, when programs are restricted to $\neg$-free disjunctive logic programs (i.e., programs without strong negation, see also the supplementary material). When looking at more general classes of logic programs, different semantics have been proposed, for example Gelfond's answer set semantics for extended disjunctive logic programs (which are represented argumentatively in Wakaki 2022, 2024)). Thus, the family of extended logic programs considered here is somewhat different than the extended logic programs considered in Wakaki 2022, 2024). In that respect, some difficulties that arise when representing disjunctive information in Reiter's default theory (Reiter 1980) are indicated and resolved in Wakaki (2024).

https://doi.org/10.1017/S1471068425100070 Published online by Cambridge University Press

Rationality postulates and connections to related non-monotonic formalisms, such as answer set semantics and disjunctive default theories, are also discussed in Wakaki 2022, 2024).

A work with a similar motivation is presented in Wang (2000), 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 Wang (2000), we define our translation in assumption-based argumentation, which means that meta-theoretical insights (e.g., complexity results (Dimopoulos 2002) or results on properties of the non-monotonic consequence relations (Čyras and Toni 2015; Heyninck and Arieli 2020; Arieli and Heyninck 2025), dialectical proof theories (Dung *et al*. 2006a, 2006b), and different implementations (Craven *et al*. 2013; Toni 2013), can be directly used.

A representation of disjunctive LP by *abstract argumentation* is studied in Bochman (2003). 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 Bochman (2003) has to allow for attacks on *sets of nodes*, instead of just nodes, necessitating a generalization of Dung's abstract argumentation frameworks (Dung 1995). 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. Yet, the characterizations in Bochman (2003) of 3-valued semantics of disjunctive logic programs by generalized abstract argumentation frameworks could provide valuable insights into the conditions (if any) under which a similar correspondence may be established with assumption-based argumentation.

Another related, but more distant line of work, is concerned with the integration of disjunctive reasoning in structured argumentation with defeasible rules (see Beirlaen *et al*. 2017, 2018). 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 LP, such as the disjunctive well-founded (Brass and Dix 1998), extended well-founded (Ross 1992), and stationary semantics (Przymusinski 1990). Some of these semantics are based on ideas that are very similar to the ideas underlying several well-known argumentation semantics. Likewise, for example, both the stationary semantics for disjunctive LP and the preferred semantics from abstract argumentation (Dung 1995) can be characterized using three instead of two "truth values". Indeed, as noted in Section 5, for normal logic programs the correspondence between the 3-valued stable models for normal logic programs (Przymusinski 1990) and complete labelings for ABA framework has been proven by Caminada and Schulz 2017, 2018). On what conditions, if any, the correspondence holds also for disjunctive logic programs, is still an open question. Finally, we hope to extend our results to more expressive languages, such as epistemic (Gelfond 1994) and parametrized LP (Goncalves and Alferes 2010).

Acknowledgments

We would like to thank the reviewers of this paper for their detailed and insightful comments. The work on this paper was partially supported by the Israel Science Foundation (Grant No. 550/19).

Supplementary material

To view supplementary material for this article, please visit https://doi.org/10.1017/S1471068425100070

References

Alcântara, J., Damásio, C. V. and Pereira, L. M. 2005. A well-founded semantics with disjunction. In *Proceedings of 21st International Conference on Logic Programming (ICLP'05), volume 3668 of Lecture Notes in Computer Science*. Springer, 341–355.

Amendola, G., Eiter, T., Fink, M., Leone, N. and Moura, J. 2016. Semi-equilibrium models for paracoherent answer set programs. *Artificial Intelligence* 234, 219–271.

Arieli, O. and Heyninck, J. 2021. Simple contrapositive assumption-based argumentation part II: reasoning with preferences. *Journal of Approximate Reasoning* 139, 28–53.

Arieli, O. and Heyninck, J. 2025. Simple contrapositive assumption-based argumentation frameworks with preferences: partial orders and collective attacks. *Journal of Approximate Reasoning* 178, 109340.

Baroni, P., Caminada, M. W. and Giacomin, M. 2018. Abstract argumentation frameworks and their semantics. In *Handbook of Formal Argumentation, volume 1*, P. Baroni, D. Gabbay, M. Giacomin and L. van der Torre, Eds. College Publications, 159–236.

Beirlaen, M., Heyninck, J. and Strasser, C. 2017. Reasoning by cases in structured argumentation. In *Proceedings of the 32nd ACM Symposium on Applied Computing (SAC'17)*. ACM, 989–994.

Beirlaen, M., Heyninck, J. and Strasser, C. 2018. A critical assessment of Pollock's work on logic-based argumentation with suppositions. In *Proceedings of 17th International Workshop on Non-Monotonic Reasoning (NMR'18)*, 63–72.

Bochman, A. 2003. Collective argumentation and disjunctive logic programming. *Journal of Logic and Computation* 13, 3, 405–428.

Bondarenko, A., Dung, P. M., Kowalski, R. and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93, 1, 63–101.

Brass, S. and Dix, J. 1995. Disjunctive semantics based upon partial and bottom-up evaluation. In *Proceedings of the Twelfth International Conference on Logic Programming (ICLP'95)*. MIT Press, 199–213.

Brass, S. and Dix, J. 1998. Characterizations of the disjunctive well-founded semantics: confluent calculi and iterated GCWA. *Journal of Automated Reasoning* 20, 1-2, 143–165.

Caminada, M. and Schulz, C. 2017. On the equivalence between assumption-based argumentation and logic programming. *Journal of Artificial Intelligence Research* 60, 779–825.

Caminada, M. and Schulz, C. 2018. On the equivalence between assumption-based argumentation and logic programming (extended abstract). In *Proceedings of the 27trh International Joint Conference on Artificial Intelligence (IJCAI'2018)*. ijcai.org, 5578–5582.

Clark, K. L. 1978. Negation as failure. In *Readings in Nonmonotonic Reasoning*, M. L. Ginsberg, Ed. Morgan Kaufmann, 311–325.

Craven, R., Toni, F. and Williams, M. 2013. Graph-based dispute derivations in assumption-based argumentation. In *Peoceeings of the 2nd International Workshop on Theory and Applications of Formal Argumentation (TAFA'13), volume 8306 of Lecture Notes in Computer Science*. Springer, 46–62.

Čyras, K., iFan, X., Schulz, C. and Toni, F. 2018. Assumption-based argumentation: disputes, explanations, preferences. In *Handbook of Formal Argumentation, volume 1*, P. Baroni, D. Gabbay, M. Giacomin and L. van der Torre, Eds. College Publications, pp. 365–408.

Čyras, K. and Toni, F. 2015. Non-monotonic inference properties for assumption-based argumentation. In *Proceedings of the 3rd International Workshop on Theory and Applications of Formal Argumentation, volume 9524 of Lecture Notes in Computer Science*. Springer, 92–111 .

Dimopoulos, Y., Nebel, B. and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence* 141, 1-2, 57–78.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77, 321–358.

Dung, P. M., Kowalski, R. A. and Toni, F. 2006a. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence* 170, 2, 114–159.

Dung, P. M., Mancarella, P. and Toni, F. 2006b. A dialectic procedure for sceptical, assumption-based argumentation. In *Proceedings of COMMA'2006, volume 144 of Frontiers in Artificial Intelligence and Applications*. IOS Press, 145–156.

Eiter, T., Gottlob, G. and Mannila, H. 1997. Disjunctive datalog. *ACM Transactions on Database Systems* 22, 3, 364–418.

Eiter, T., Leone, N. and Saccá, D. 1997. On the partial semantics for disjunctive deductive databases. *Annals of Mathematics and Artificial Intelligence* 19, 1-2, 59–96.

Etherington, D. W. and Crawford, J. M. 1996. Toward efficient default reasoning. In *Proceedings of AAAI/IAAI'96*, W. J. Clancey and D. S. Weld, Eds. AAAI Press / MIT Press, 627–632.

Gelfond, M. 1994. Logic programming and reasoning with incomplete information. *Annals of Mathematics and Artificial Intelligence* 12, 1-2, 89–116.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference on Logic Programming (ICLP'88)*. MIT Press, 1070–1080.

Gelfond, M. and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9, 3-4, 365–385.

Gelfond, M., Przymusinska, H., Lifschitz, V. and Truszczynski, M. 1991. Disjunctive defaults. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*. Morgan Kaufmann, 230–237.

Gonçalves, R. and Alferes, J. J. 2010. Parametrized logic programming. In *European Workshop on Logics in Artificial Intelligence*. Springer, 182–194.

Gottlob, G. 1994. Complexity and expressive power of disjunctive logic programming (research overview). In *Proceedings of the 1994 International Symposium on Logic programming*. MIT Press, 23–42.

Heyninck , J. and Arieli, O. 2019. An argumentative characterization of disjunctive logic programming. In *Proceedings of the 19th EPIA Conference on Artificial Intelligence, Part II, volume 11805 of Lecture Notes in Computer Science*. Springer, 526–538.

Heyninck, J. and Arieli, O. 2020. Simple contrapositive assumption-based argumentation frameworks. *Journal of Approximate Reasoning* 121, 103–124.

Heyninck, J., Arieli, O. and Bogaerts, B. 2024. Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming. *Artificial Intelligence* 331, 104110.

Heyninck, J. and Bogaerts, B. 2023. Non-deterministic approximation operators: ultimate operators, semi-equilibrium semantics, and aggregates. *Theory and Practice of Logic Programming* 23, 4, 632–647.

Kleene, S. C. 1950. *Introduction to Metamathematics*. D. Van Nostrand Company, New York.

Knorr, M. and Hitzler, P. 2007. A comparison of disjunctive well-founded semantics. In *Foundations of Artificial Intelligence (FAInt), volume 277 of CEUR Workshop Proceedings*. CEUR-WS.org.

Lifschitz, V. 1985. Computing circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI'85)*, A. K. Joshi, Ed. Morgan Kaufmann, 121–127.

Lloyd, J. W. 1987. *Foundations of Logic Programming*. Springer-Verlag, Berlin.

Przymusinski, T. 1990. Stationary semantics for disjunctive logic programs and deductive databases. In *Proceedings of the 1990 North American Conference on Logic Programming*. MIT Press, 40–59.

Przymusinski, T. C. 1990. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae* 13, 4, 445–463.

Przymusinski, T. C. 1991. Stable semantics for disjunctive programs. *New Generation Computing* 9, 401–424.

Reiter, R. 1978. On closed world data bases. In *Logic and Data Bases*, H. Gallaire and J. Minker, Eds., 119–140.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13, 1, 81–132.

Ross, K. A. 1992. A procedural semantics for well-founded negation in logic programs. *The Journal of Logic Programming* 13, 1, 1–22.

Sacca, D. and Zaniolo, C. 1990. Stable models and non-determinism in logic programs with negation. In *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 205–217.

Sacca, D. and Zaniolo, C. 1991. Partial models and three-valued models in logic programs with negation. In *LPNMR*, volume 91, 87–101.

Sacca, D. and Zaniolo, C. 1997. Deterministic and non-deterministic stable models. *Journal of Logic and Computation* 7, 5, 555–579.

Sakama, C. and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation* 5, 3, 265–285.

Sakama, C. and Inoue, K. 2000. Abductive logic programming and disjunctive logic programming: their relationship and transferability. *Journal of Logic Programming* 44, 1-3, 75–100.

Schulz, C. 2015. Graphical representation of assumption-based argumentation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Schulz, C., Satoh, K. and Toni, F. 2015. Characterising and explaining inconsistency in logic programs. In *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, 467–479.

Schulz, C. and Toni, F. 2016. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming* 16, 1, 59–110.

Seipel, D. 1998. An alternating well-founded semantics for query answering in disjunctive databases. In *Proceedings of FQAS'98*. Springer, 341–353.

SHEN, Y.-D. and EITER, T. 2019. Determining inference semantics for disjunctive logic programs. *Artificial Intelligence* 277, 103165.

SU, E.2015. *Extensions of equilibrium logic by modal concepts*. PhD thesis, IRIT-Institut De Recherche en Informatique De Toulouse.

THEVAPALAN, A., HEYNINCK, J. and KERN-ISBERNER, G. 2021. Establish coherence in logic programs modelling expert knowledge via argumentation. In *CAUSAL'21: Workshop on Causal Reasoning and Explanation in Logic Programming, volume 1613 of CEUR Workshop Proceedings*.

TONI, F. 2013. A generalised framework for dispute derivations in assumption-based argumentation. *Artificial Intelligence* 195, 1–43.

WAKAKI, T. 2020. Consistency in assumption-based argumentation. In *Proceedings of the the 8th International Conference on Computational Models of Argument (COMMA'20), volume 326 of Frontiers in Artificial Intelligence and Applications*. IOS Press, 371–382.

WAKAKI, T. 2022. Assumption-based argumentation for extended disjunctive logic programming. *Proceedings of the 12th International Symposium on Foundations of Information and Knowledge Systems (FoIKS-2022), volume 13388 of Lecture Note in Computer Science*. Springer, 35–54.

WAKAKI, T. 2024. Assumption-based argumentation for extended disjunctive logic programming and its relation to nonmonotonic reasoning. *Journal of Argument and Computation* 15, 3, 309–353.

WANG, K. 2000. Argumentation-based abduction in disjunctive logic programming. *The Journal of Logic Programming* 45, 1-3, 105–141.

WANG, K. and ZHOU, L. 2005. Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Transactions on Computational Logic* 6, 295–327.

YOU, J.-H. and YUAN, L.-Y. 1994. A three-valued semantics for deductive databases and logic programs. *Journal of Computer and System Sciences* 2, 334–361.