



Structured argumentation dynamics

Undermining attacks in default justification logic

Stipe Pandžić¹ 

Accepted: 2 July 2021 / Published online: 27 July 2021
© The Author(s) 2021

Abstract

This paper develops a logical theory that unifies all three standard types of argumentative attack in AI, namely rebutting, undercutting and undermining attacks. We build on default justification logic that already represents undercutting and rebutting attacks, and we add undermining attacks. Intuitively, undermining does not target default inference, as undercutting, or default conclusion, as rebutting, but rather attacks an argument's premise as a starting point for default reasoning. In default justification logic, reasoning starts from a set of premises, which is then extended by conclusions that hold by default. We argue that modeling undermining defeaters in the view of default theories requires changing the set of premises upon receiving new information. To model changes to premises, we give a dynamic aspect to default justification logic by using the techniques from the logic of belief revision. More specifically, undermining is modeled with belief revision operations that include contracting a set of premises, that is, removing some information from it. The novel combination of default reasoning and belief revision in justification logic enriches both approaches to reasoning under uncertainty. By the end of the paper, we show some important aspects of defeasible argumentation in which our logic compares favorably to structured argumentation frameworks.

Keywords Justification logic · Defeasible argumentation · Reasoning dynamics · Undermining attacks

Mathematics Subject Classification (2010) 03B60 · 68T27 · 68T30 · 68T37

1 Introduction

In this paper, we investigate the dynamics of default theories with justification logic formulas that represent defeasible arguments. The existing work about dynamics in formal argumentation [2, 3, 17, 23–26, 49, 64] almost entirely focuses on abstract argumentation

✉ Stipe Pandžić
s.pandzic@uu.nl

¹ Theoretical Philosophy Group, Department of Philosophy and Religious Studies, Utrecht University, Utrecht, The Netherlands

frameworks in the style of Dung [27]. The literature on the dynamics of structured argumentation is limited to the DeLP framework [4, 5], where the dynamics is understood as adding or removing strict and defeasible rules, and ASPIC+ [47], where the dynamic component is meant to resolve symmetric attacks by updating preferences. The current paper, based on [52], advances this line of research by specifying a variety of dynamic operators for modeling changes of argument systems based on default justification logic.

The basic format of our default theories has been developed in [50, 51, 54], where we interpret justification formulas of the form $t : F$ as arguments that can defeat other formulas by means of undercut or rebuttal. Informally, for some defeasible argument understood as a premise–claim pair, a rebutting defeater is a reason for the opposite claim. An undercutting defeater is a reason that attacks the connection between the premises and the claim supported by the premises. In Pollock’s [55] favourite example, an agent starts from the fact that a table looks red, which gives the agent a default reason to claim that the color of the table is red. If the agent finds out that the table is illuminated with red light, the agent has an undercutting reason against the initial reasoning to the conclusion that the table is in fact red. Moreover, were the agent also to find out that the table has been painted white recently, the agent would have a rebutting reason against to the claim that it is red-colored. These types of conflicts are illustrated in Fig. 1, together with *undermining* which will be introduced shortly.

Technically, the workings of undercut and rebuttal rely on defining default theories with default rules based on justification logic formulas. In such rules, justification terms codify defeasible inferences in their structure. Besides undercutting and rebutting defeat, AI researchers have investigated an additional standard type of argument defeat called “undermining”. According to [29, p. 626], an argument is undermined if its premises or assumptions are attacked. In an extension of Pollock’s example, an undermining attack would target the starting premise that the table looked red in the first place.

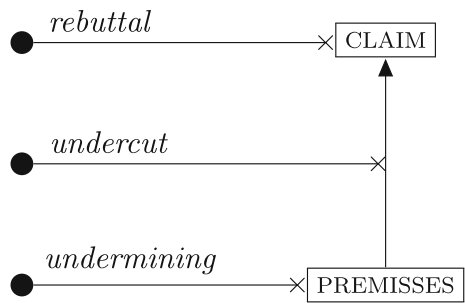
Notice that undermining does not target default inference, as undercutting, or default conclusion, as rebutting, but rather attacks an argument’s premise as a starting point for default reasoning. This motivated the distinction between default and plausible reasoning in formal argumentation that we adopt in this paper. In the plausible reasoning paradigm, fallibility of reasoning results from adding new information that questions old information and, thereby, it might question old conclusions.¹ In contrast, in the default reasoning paradigm, fallibility results from adding some further true information on top of existing information and this new information in turn gives reasons to question old conclusions, but they do not question old information.²

We will show that introducing dynamic operators for justification logic default theories enables us to model undermining. Defeating an argument by attacking its premise or its assumption is not new to structured argumentation. In assumption-based argumentation (ABA) [28], all attacks are reduced to this type and in ASPIC+ [56], ordinary premises of an

¹Rescher’s [62, 63] work is the landmark reference for the study of plausible reasoning. Rescher [63, p. 39] claims that “a thesis is more or less plausible depending on the reliability of the sources that vouch for it”.

²Note here that Prakken [57, p. 2198] refers to the difference between *defeasible* and *plausible* reasoning, instead of default and plausible reasoning. To be clear about the terminology, we use the etymologically close terms “defeasible” and “defeat” in a more general sense, so that, e.g., undermining is normally also considered as a type of defeat. This conforms to the standard usage of “defeat” and “defeasibility” which simply mean that something is annulled. The term “default”, on the other hand, has a more specific meaning related to default assumptions introduced by Reiter [60, p. 82]. Vreeswijk [71] introduced the distinction between the two kinds of non-monotonicity to argumentation theory (using the term “defeasible”). See [57, pp. 2241–2243] for more details on this distinction.

Fig. 1 Three types of defeat between arguments



argument are susceptible to undermining. However, these systems do not provide an insight into the logical workings of undermining, because they specify neither a concrete logical language nor inference rules.

In our default theories, undermining can be given a precise logical interpretation. While undercut and rebuttal rely on the uncertainty of default arguments, undermining changes the context from which agents make further inferences. For a specific default theory, this context is determined by the set of starting premises, W . Our idea is that, since undermining targets the given premises, it should be modeled as a result of non-inferential information inputs that require contracting the set of premises (or facts) of a default theory. This means that we will define undermining by “climbing up” the definitions of more fundamental operations of default theory changes. To elicit the reasoning process behind undermining, we specify four different logical operations that model undermining: prioritized and non-prioritized contraction and prioritized and non-prioritized revision.

The paper is organized as follows. The logic introduced in [50, 51, 54] already models non-monotonicity with the use of undercut and rebuttal. We use this logic to exemplify Toulmin’s [67] argumentation model in our default theories. The example is later extended to illustrate conceptual differences in the workings of undercutting, rebutting and undermining. After specifying the elements of arguments in default justification logic, we describe how we plan to connect dynamic operations for such default theories with undermining defeaters. Section 3 is the main technical contribution of this paper, where we define dynamic operations for default theories with justification formulas. The operations we introduce combine Hansson’s [39] base revision operations with a specific kind of standard Reiter default rules. Our approach to defining the dynamic operators for default theory revision has most in common with Antoniou’s [7] approach, which deals with the dynamics of Reiter’s default theories. We show by the end of Section 3 that undermining attacks on premises correspond to those dynamic operations that involve either contraction or a variant of non-prioritized contraction. In Section 4, our goal is to show that there are some important advantages of using default justification logic for argumentation over using structured frameworks for argumentation. The same section also establishes rational behaviour of our system with respect to the “non-interference” postulate [19]. Finally, Section 5 gives an extensive overview of the related work.

2 Logic of default reasons

The logic of default justifications was first defined in [50] and our definitions in this section follow those given in [54]. We start by outlining the underlying justification logic on which

we base default theories. Before we formally introduce syntax, several preliminary remarks on how to build justification assertions of the type $t : F$ are in order.

One of the basic operations of standard justification logics is *Application* (\cdot) exemplified by the following formula

$$u : (F \rightarrow G) \rightarrow (t : F \rightarrow (u \cdot t) : G).$$

By focusing only on the propositional content of the formula, one may notice the familiar structure of a *modus ponens* inference from $F \rightarrow G$ and F to G . The added machinery of reason terms syntactically captures the dependence of the consequent G on the propositions $F \rightarrow G$ and F . This dependence is witnessed by the structure of the reason term $(u \cdot t)$ which shows that the reason u has been applied to the reason t .

In addition to application, all standard justification logics include the operation known as *Sum* $+$. Sum admits of merging two reason terms into a single term as done, for example, in the formula

$$t : F \rightarrow (t + u) : F.$$

The intuition behind Sum is that, if evidence t justifies some formula F , then adding new evidence u to t will not invalidate t being a reason for F . Thus, standard justifications are inherently non-defeasible.

The underlying logic for our default theories is a standard justification logic with non-defeasible and truth-inducing reasons. We assume both Application and Sum operations for this logic. Moreover, to ensure that all reasons in this logic are truth-inducing, the logic contains the axiom $t : F \rightarrow F$ called *Factivity*. This axiom corresponds to the modal logic truth axiom: $\Box F \rightarrow F$ read as “If F is known, then F ”. Our choice of basing default theories on a logic of non-defeasible and truth-inducing reasons is in line with the strategy of standard default logics [6], where defeasible conclusions are based on certain information.

2.1 Logic of factive reasons $\mathbf{JT}_{\mathbf{CS}}$

The first variant of justification logic was the logic of proofs (LP) introduced in [8] and fully developed in [9]. Two recent monographs on justification logic systems [11, 41] report on this growing research area. The logic of non-defeasible and factive reasons that we use here was first defined by Brezhnev [18]. For more basic information on its relation to other justification logics see [32].

2.1.1 Syntax, axioms and rules

The basic format of justification assertions is “ $t : F$ ”, where a justification term “ t ” is informally interpreted as a reason or justification for “ F ”. The set Tm consists of exactly all justification terms, constructed from variables x_1, \dots, x_n, \dots and proof constants c_1, \dots, c_n, \dots by means of operations \cdot and $+$, i.e., by means of the above discussed Application and Sum. The following Backus-Naur form gives the grammar of justification terms:

$$t ::= x \mid c \mid (t \cdot t) \mid (t + t)$$

where x is a variable denoting an unspecified justification and c is a proof constant, taken as atomic within the system. A set of subterms $Sub(t)$ is defined by induction on the construction of a term t . The set Fm consists of exactly all $\mathbf{JT}_{\mathbf{CS}}$ formulas based on the countable set of propositional atoms \mathcal{P} and defined by the following Backus-Naur form:

$$F ::= \top \mid P \mid (F \rightarrow F) \mid (F \vee F) \mid (F \wedge F) \mid \neg F \mid t : F$$

where $P \in \mathcal{P}$ and $t \in Tm$.

We can now define the logic \mathbf{JT}_{CS} , which is the weakest logic with non-defeasible reasons containing axiom schemes for the two basic operations \cdot and $+$. These are the axioms of \mathbf{JT}_{CS} :

- A0 All the instances of propositional logic tautologies from Fm
- A1 $t : (F \rightarrow G) \rightarrow (u : F \rightarrow (t \cdot u) : G)$ (Application)
- A2 $t : F \rightarrow (t + u) : F; u : F \rightarrow (t + u) : F$ (Sum)
- A3 $t : F \rightarrow F$ (Factivity)

Before defining the rules of inference, we first discuss the role that proof constants c_1, \dots, c_n, \dots have within the system. Proof constants are justifications of basic logic axioms, which are considered to be “justified *ex officio*” [10, § 2.3], that is, by their status in the system. The set of all formulas justified by proof constants is called a *Constant Specification* (\mathcal{CS}) set:

Definition 1 (Constant specification) $\mathcal{CS} = \{c_n : c_{n-1} : \dots : c_1 : F \mid F \text{ is an instance of A0-A3, } c_n, c_{n-1}, \dots, c_1 \text{ are proof constants and } n \in \mathbb{N}\}$

Notice that \mathcal{CS} contains the basic logical axioms justified by proof constants and all the justification assertions likewise produced. In justification logics, constant specification sets are used to parametrize the counterpart of the modal logic *necessitation* rule: “If F is provable, then infer $\Box F$ ”.

Together with the standard *modus ponens* rule, \mathbf{JT}_{CS} logic includes the justification logic variant of the necessitation rule:

- R0 From F and $F \rightarrow G$ infer G (Modus ponens)
- R1 If F is an axiom instance of A0-A3 and c_n, c_{n-1}, \dots, c_1 proof constants such that $c_n : c_{n-1} : \dots : c_1 : F \in \mathcal{CS}$, then infer $c_n : c_{n-1} : \dots : c_1 : F$ (Iterated axiom necessitation)

The necessitation rules of justification logics regulate only logical awareness of axioms. Theorem necessitation turns into a constructive property of justification logic derivations called *Internalization* [11, p. 21]. To ensure that the Internalization holds for \mathbf{JT}_{CS} and following the idea that each logical axiom is justified by exactly one constant, we require that \mathcal{CS} in R1 respects the following two conditions:

- Axiomatically appropriate: for each axiom instance A , there is a constant c such that $c : A \in \mathcal{CS}$ and for each formula $c_n : c_{n-1} : \dots : c_1 : A \in \mathcal{CS}$ such that $n \geq 1$, there is a constant c_{n+1} such that $c_{n+1} : c_n : c_{n-1} : \dots : c_1 : A \in \mathcal{CS}$;
- Injective: each proof constant c justifies at most one formula.

We could think of a set of formulas generated by assigning a Gödel number to each axiom instance and to each instance of R1 as a concrete example of one such constant specification set.

We say that the formula F is \mathbf{JT}_{CS} -provable ($\mathbf{JT}_{CS} \vdash F$) if F can be derived using the axioms A0-A3 and rules R0 and R1. A concrete example of the workings behind the \mathbf{JT}_{CS} axioms is the derivation of $t : P \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)$.

Proof ($\mathbf{JT}_{CS} \vdash t : P \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)$)

- 1 $P \rightarrow (Q \rightarrow P)$ (A0)
- 2 $c : [P \rightarrow (Q \rightarrow P)]$ (1, R1)
- 3 $c : [P \rightarrow (Q \rightarrow P)] \rightarrow [t : P \rightarrow (c \cdot t) : (Q \rightarrow P)]$ (A1)
- 4 $t : P \rightarrow (c \cdot t) : (Q \rightarrow P)$ (2, 3R0)
- 5 $(c \cdot t) : (Q \rightarrow P) \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)$ (A2)
- 6 $t : P \rightarrow (c \cdot t) : (Q \rightarrow P) \rightarrow ([(c \cdot t) : (Q \rightarrow P) \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)] \rightarrow [t : P \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)])$ (A0)
- 7 $(c \cdot t) : (Q \rightarrow P) \rightarrow ((c \cdot t) + u) : (Q \rightarrow P) \rightarrow [t : P \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)]$ (4, 6R0)
- 8 $t : P \rightarrow ((c \cdot t) + u) : (Q \rightarrow P)$ (5, 7R0)

□

2.1.2 Semantics

The now-standard semantics of standard justification logics, known as “Fitting’s semantics” [30, 31], is essentially an elaborated possible worlds semantics for epistemic logic. As familiar from the standard epistemic logic, Fitting’s semantics requires F to be true in all accessible alternatives for the truth of the justification assertion $t : F$. In addition to this, the truth of $t : F$ is conditioned on the admissibility of t as a reason for the formula F in the current state. The semantics for \mathbf{JT}_{CS} we define here is based on a single-world justification model, which adapts Mkrtychev’s single-world model semantics for the logic of proofs (LP) [46]. The \mathbf{JT}_{CS} logic is sound and complete with respect to Mkrtychev’s reflexive interpretations [46].

In the logic of default reasons from Section 2.2, the main idea of the semantics is to give a procedure that determines the admissibility of reasons. This justifies our choice to fully focus on the admissibility condition by taking the single-world models perspective, as done in [54].

Definition 2 (\mathbf{JT}_{CS} model) We define a function *reason assignment* based on \mathcal{CS} ,

$$*(\cdot) : Tm \rightarrow 2^{Fm},$$

a function mapping each term to a set of formulas from Fm . We assume that it satisfies the following conditions:

1. If $F \rightarrow G \in *(t)$ and $F \in *(u)$, then $G \in *(t \cdot u)$,
2. $*(t) \cup *(u) \subseteq *(t + u)$ and
3. If $c : F \in \mathcal{CS}$, then $F \in *(c)$.

A *truth assignment* $v : \mathcal{P} \rightarrow \{\text{True}, \text{False}\}$ is a function assigning truth values to propositional atoms in \mathcal{P} . We define the interpretation \mathcal{I} as a pair $(v, *)$. For an interpretation \mathcal{I} , we define \models as a truth relation on the set of formulas of \mathbf{JT}_{CS} . We say that, for any formula $t : F \in Fm$, it holds that

$$\mathcal{I} \models t : F \text{ iff } F \in *(t).$$

Truth conditions for atomic propositions, \neg , \rightarrow , \wedge and \vee are defined as usual.

An interpretation \mathcal{I} is *reflexive* iff the truth relation for \mathcal{I} fulfills the following condition:

- For any term t and any formula F , if $F \in *(t)$, then $\mathcal{I} \models F$.

The consequence relation of the logic of factive reasons \mathbf{JT}_{CS} is defined on reflexive interpretations:

Definition 3 (\mathbf{JT}_{CS} consequence relation) $\Sigma \models F$ iff for all reflexive interpretations \mathcal{I} , if $\mathcal{I} \models B$ for all $B \in \Sigma$, then $\mathcal{I} \models F$.

For a set of formulas $\Gamma \subseteq Fm$ and the \mathbf{JT}_{CS} consequence relation \models defined above, a \mathbf{JT}_{CS} closure of Γ is given by $Th^{JT_{CS}}(\Gamma) = \{F \mid \Gamma \models F\}$. For a closure $Th^{JT_{CS}}(\Gamma)$, it holds that $\mathcal{CS} \subseteq Th^{JT_{CS}}(\Gamma)$. We will also make use of a weaker “non-monotonic” \mathbf{JT}^- closure for which a reason assignment function $*(\cdot)$ does not meet the sum condition (2) and a constant specification set is additionally restricted to $\mathcal{CS}^- \subset \mathcal{CS}$, such that \mathcal{CS}^- excludes each formula $c_n : c_{n-1} : \dots : c_1 : F$ with F being an instance of A2. The purpose of weakly closing a set of \mathbf{JT}_{CS} formulas is to investigate which formulas follow according to all the standard axioms and rules, but without assuming the monotonicity of justifications imposed by the sum axioms A2.

2.2 Logic of default reasons

Building on the \mathbf{JT}_{CS} syntax, we introduce the definition of the *default theory*:

Definition 4 (Default Theory) A default theory T is a pair (W, D) , where W is a finite set of \mathbf{JT}_{CS} formulas and D is a countable set of default rules. Each default rule is of the following form:

$$\delta = \frac{t : F :: (u \cdot t) : G}{(u \cdot t) : G}.$$

The informal reading of the default δ is: “If t is a reason justifying F , and it is consistent to assume that $(u \cdot t)$ is a reason justifying G , then $(u \cdot t)$ is a defeasible reason justifying G ”. In the rule δ , the formula $t : F$ is called the *prerequisite*, $(u \cdot t) : G$ above the line is called the *consistency requirement* and $(u \cdot t) : G$ below the line is called the *consequent* of δ . We refer to each of these formulas by $pre(\delta)$, $req(\delta)$ and $cons(\delta)$, respectively. The default rule δ introduces a *unique* reason term u , which means that, for a default theory T , the following three conditions are required:

1. For any formula $v : H$ such that $v : H \in Th^{JT_{CS}}(W)$, it holds that $u \neq v$;
2. For any formula $H \in W$, $u : (F \rightarrow G)$ is not a subformula of H and
3. For any default rule $\delta' \in D$ such that $\delta' = \frac{t' : F' :: (u' \cdot t') : G'}{(u' \cdot t') : G'}$, if $u = u'$, then $F = F'$ and $G = G'$.

The first uniqueness condition reflects the idea that the reason term u holds only *other things being equal*. As such, it cannot justify formulas entailed by the set of starting premises that are considered facts. In other words, u is not considered to be a factive reason. The second uniqueness condition has a twofold role. Technically, it blocks redundancy that would result from including $u : (F \rightarrow G)$ in W . Were it the case that both $u : (F \rightarrow G)$ and $t : F$ are in W , $(u \cdot t) : G$ would be contained in the closure of W by application. Conceptually, the second condition blocks that there is any certainty regarding whether the formula $u : (F \rightarrow G)$ holds or not. This is only determined in the semantics for default theories

that will be given shortly. The last uniqueness condition reserves the reasoning codified by u as a justification of the conditional $F \rightarrow G$, where F and G are some specific **JTCS** formulas. Although this condition does not preclude the possibility of two defaults sharing the underlying conditional $u : (F \rightarrow G)$, it imposes some limitations on the expresiveness of our defaults to which we return in Section 5. As it will become clearer by the end of this section, the reason term u in δ is ultimately a **JTCS** codification of some derivations steps taken to “prove” the formula $F \rightarrow G$. But the derivation is such that it is conditional on at least one assumption that we cannot get rid of by the end of derivation.

Every default rule produces a reason term whose structure codifies an application operation step. In contrast to axiom A1, we do not require the formula $u : (F \rightarrow G)$ in δ above to be a part of the knowledge base. Instead, $u : (F \rightarrow G)$ is the underlying assumption of δ on the basis of which we are able to extend an incomplete knowledge base. The propositions of this kind are important in the system since they function as rules allowing for default steps, but they are also specific **JTCS** formulas. They will be referred to as “warrants”, because their twofold role mirrors that of Toulmin’s argument warrants [67, p. 91]. To illustrate how we represent arguments in justification logic, we will use Toulmin’s famous example [67, p. 92] of arguing for the claim that Harry is a British subject.

Example 5 The claim that Harry is a British subject “can be defended by appeal to the information that he was born in Bermuda, for this datum lends support to our conclusion on account of the warrants implicit in the British Nationality Acts...” [67, p. 94]. The example is translated into a justification logic default as follows. Given the fact that Harry was born in Bermuda (B), an agent can conclude that Harry is a British subject (S):

$$\frac{r : B :: (s \cdot r) : S}{(s \cdot r) : S}$$

The default can be read as follows: “If r is a reason justifying that Harry was born in Bermuda and it is consistent to assume that $(s \cdot r)$ is a reason justifying that Harry is a British subject, then $(s \cdot r)$ is a defeasible reason justifying that Harry is a British subject”.

An argument in default justification logic can be delimited in three different ways, depending on how many layers of the argument are taken into consideration. Figure 2 depicts the above formalized argument as a three-layered structure inspired by Toulmin’s layout of arguments, including the concepts of data, claim, warrant and backing. Notice that Fig. 2 associates a concrete logical object to each component of Toulmin’s argument

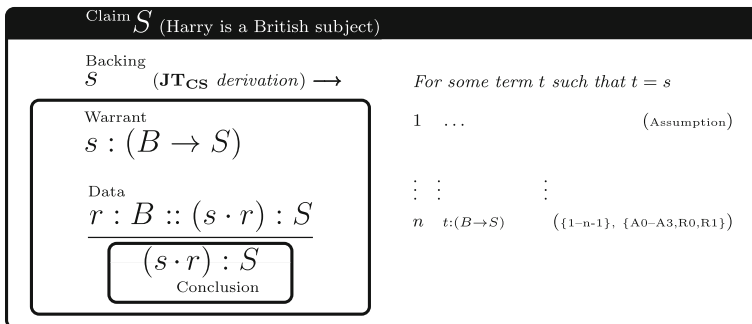


Fig. 2 Three-layered structure of arguments in the **JTCS** language. Toulmin’s layout of arguments

schema. For example, the prerequisite formula $r : B$ represents data for the argument understood as a default rule. By the end of the section, we emphasize the role of warrants, such as $s : (B \rightarrow S)$, in defining argumentation semantics. Before we give the definition of argumentation semantics, we first focus on defining the basis of collecting new information by applying the available defaults.

The procedure for applying defaults in our default theories is inspired by Antoniou’s [6] operational semantics for Reiter’s default theories [60]. The approach taken by Antoniou is to provide a constructive characterization of default theory extensions by determining those sequences of defaults that generate extensions. For our default theory $T = (D, W)$ a *sequence* of default rules $\Pi = (\delta_0, \delta_1, \dots)$ is a possible order in which a list of default rules without multiple occurrences from D is applied (Π is possibly empty). Applicability of defaults is determined by the following definition:

Definition 6 (Applicability) For a set of \mathbf{JT}_{CS} -closed formulas Γ we say that a default rule $\delta = \frac{t:F::(u:t):G}{(u:t):G}$ is applicable to Γ iff $t : F \in \Gamma$ and $\neg(u \cdot t) : G \notin \Gamma$.

Default consequents are brought together in the set of \mathbf{JT}_{CS} formulas that represents the current *evidence base*:

Definition 7 (Evidence base) $In(\Pi) = Th^{JT_{CS}}(W \cup \{cons(\delta) \mid \delta \in \Pi\})$.

By an abuse of set-builder notation, we use ‘ \in ’ as a shorthand for “occurs in” when defaults are singled out as members of sequences of defaults. The set $In(\Pi)$ pools reason-based formulas. Their acceptability depends on the acceptability of other available (counter-)reasons and this is dealt with later in argumentation semantics. All our default theories are *normal*, which means that any default has a single consistency requirement formula which corresponds to its consequent formula. Our operational semantics is thus simplified, because we do not need to define *Out*-sets [6, p. 32] that keep track of assumptions made in applying non-normal defaults, as done in the operational semantics for standard default logics.

In the operational part of the semantics, we need to determine those sequences of defaults that are significant for a theory T , namely, default processes. For a sequence Π , the initial segment of the sequence is denoted as $\Pi[k]$, where k indicates the number of elements contained in that segment of the sequence and where Π has at least k elements. Any segment $\Pi[k]$ is also a sequence. We can now define default processes:

Definition 8 (Process) A sequence of default rules Π is a *process* of a default theory $T = (W, D)$ iff every k such that $\delta_k \in \Pi$ is *applicable* to the set $In(\Pi[k])$, where $\Pi[k] = (\delta_0, \dots, \delta_{k-1})$.

The kind of process that we are focusing on is called *closed* process: a process Π is said to be closed iff every $\delta \in D$ that is applicable to $In(\Pi)$ is already in Π . Closed processes are necessary to define extension sets for default theories.

Building on the operational part of the semantics, we can now define the argumentation part of the semantics, with the main role that warrants perform in this part. Warrants extend the interpretation of the application operation “ \cdot ”, and their use determines how we define defeaters in justification logic. Each warrant is made explicit by means of a function *warrant assignment*:

$$\#(\cdot) : D \rightarrow Fm.$$

The function maps each default rule to a specific justified conditional as follows:

$$\#(\delta) = u : (F \rightarrow G),$$

where $\delta \in D$ and $\delta = \frac{t:F::(u-t):G}{(u-t):G}$, for some reason term t , a unique reason term u and some formulas F and G . As mentioned above, uniqueness of u does not prevent two default rules to share a warrant formula. This reflects also the informal idea of warrants as general rules that are, in principle, applicable to different starting data, that is, different prerequisites of defaults.³ A set of all such underlying warrants of default rules is called *Warrant Specification* (\mathcal{WS}) set.

Definition 9 (Warrant specification) For a default theory $T = (W, D)$, justified defeasible conditionals are given by the *Warrant Specification* set:

$$\mathcal{WS}^T = \#(D) = \{\#(\delta) \mid \delta \in D\}.$$

When relativized to a specific process Π , we will talk about the set \mathcal{WS}^Π .

The possibility to refer to warrants within the language enables us to model *undercutting* defeaters [21, 55]. Intuitively, an undercutter u for $t : F$ gives us a reason not to accept a claim F on account of establishing that there is some exclusionary circumstance in light of which which t does not support F . In our logic, undercutting reasons defeat other reasons by denying that their warrant provides support for the conclusion in an undercutting circumstance.

Definition 10 (Undercut) A reason u undercuts reason t being a reason for a formula F in a set of \mathbf{JT}_{CS} formulas $\Gamma \subseteq In(\Pi[k])$ iff

$$\bigvee_{v \in Sub(t)} u : \neg[v : (G \rightarrow H)] \in Th^{JT_{CS}}(\Gamma)$$

and $v : (G \rightarrow H) \in \mathcal{WS}^{\Pi'}$ for a process Π' of T .

We say that a set $\Gamma \subseteq In(\Pi[k])$ undercuts reason t being a reason for a formula F iff, for a subterm v of t , $\neg[v : (G \rightarrow H)] \in Th^{JT_{CS}}(\Gamma)$ and $v : (G \rightarrow H) \in \mathcal{WS}^{\Pi'}$ for a process Π' of T . According to the definition, undercut is first and foremost an asymmetric attack relation, but it is possible that two reasons undercut each other. The undercut relation can be intuitively explained through the three layers of justification logic arguments from Fig. 2. Undercut occurs at the second layer where we consider warrants. In the context of defaults, warrants function as formulas representing rules of inference whereby we introduce defeasible arguments. By introducing some argument $u : \neg[s : (B \rightarrow S)]$, the default rule, that is, the warrant $s : (B \rightarrow S)$, has been attacked and the defeasible argument $(s \cdot r) : S$ is undercut. At the layer of backings, undercut can be seen as occurring between two terms. For example, the \mathbf{JT}_{CS} derivation $u' \in Sub(u)$ “proves” that an assumption made in the \mathbf{JT}_{CS} derivation s does not provide support for $B \rightarrow S$ under an added circumstance. Notice that for these two layers, undercut might not be directly connected to the formula S . For example, u would also undercut some reason $(s'' \cdot (s' \cdot (s \cdot r)))$ for S' by undercutting

³Formally, we also do not require that $t = t'$ holds in the antecedent of condition 3.) for the general definition of defaults above. This reflects the independence of the warrant $u : (F \rightarrow G)$ from the data (or premise) $t : F$ to which we apply the warrant.

$(s \cdot r) : S$ as its sub-argument.⁴ This justifies a reading of Definition 10 on which u undercuts t for any formula F . Informally, we could say that an undercutter u attacks t whenever t is quoted as a step to get to the claim F from some starting data.

Besides undercut, default reasons may conflict each other because they cannot be both added to a same consistent process. For example, if a knowledge base contains a reason for a statement F , then any default introducing a reason for $\neg F$ will be inapplicable. This enables us to obtain a formal representation of *rebuttal* among reasons for free already from the operational semantics. Rebuttal is ultimately based on \mathbf{JT}_{CS} inconsistency and inapplicability of defaults, which means that this type of conflict induces symmetric attacks. We use both undercut and \mathbf{JT}_{CS} inconsistency to characterize some of the standard argumentation extension notions [27] as, e.g., in defining stable semantics. The following definition specifies conflict-free sets of arguments as those sets of arguments that are undercut-free as follows:

Definition 11 (Conflict-free sets) A set of \mathbf{JT}_{CS} formulas Γ is conflict-free iff $Th^{JT_{CS}}(\Gamma)$ does not undercut reason t for a formula F such that $t : F \in Th^{JT^-}(\Gamma)$.

Notice the use of the \mathbf{JT}^- closure in this definition. This type of closure, which we called “non-monotonic” avoids closing under the Sum operation. The \mathbf{JT}^- closure is a tool to “filter out” some instances of undercut that do not generate genuine conflicts between arguments. Assume, for instance, that a reason term u undercuts some term t as a reason for F . Then u also undercuts $(v + t)$ as a reason for G , for some G justified by v . Such undercutter might be considered irrelevant. More specifically, if we assume that v is a factive justification for G , then the undercut of $(v + t)$ by u would not seem to genuinely defeat $(v + t)$ as a reason for G . The weak closure operation \mathbf{JT}^- is used to define conflict-free sets by setting aside such cases of undercut that are induced by merely concatenating reasons.

Using the above definitions, we now define the semantics of acceptability for justification assertions of the type $t : F$. This is the way to interpret justification assertions as arguments in our semantics. In reasoning from a default theory $T = (W, D)$, an agent always considers potential extension sets of \mathbf{JT}_{CS} formulas that meet the following conditions:

1. $W \subseteq \Gamma$ and
2. $\Gamma \subseteq W \cup \{cons(\Pi) \mid \Pi \text{ is some process of } T\}$,

where $cons(\Pi) = \{cons(\delta) \mid \delta \in \Pi\}$. Potential extension sets are not closed under the \mathbf{JT}_{CS} closure. The idea is that, before we are able check whether there are any potential conflicts in a set of arguments and ascertain that a set of arguments is defensible, we do not know yet whether all the logical consequences of those arguments are desirable. In Definition 13, we apply the same strategy for \mathbf{JT}_{CS} -admissible extensions, which neither maximize inclusion of acceptable arguments nor do they maximize inclusion of \mathbf{JT}_{CS} -derivable formulas. Before we define admissible sets, we define the notion of acceptability of an argument $t : F$:

Definition 12 (Acceptability) For a process Π of a default theory $T = (W, D)$, a formula $t : F \in cons(\Pi)$ is acceptable w.r.t. a set of \mathbf{JT}_{CS} formulas $\Gamma \subseteq In(\Pi)$ iff for each undercutting reason u for t being a reason for F such that $u : G \in In(\Pi)$, $Th^{JT_{CS}}(\Gamma)$ undercuts u being a reason for G .

⁴Notice the use of the set of subterms $Sub(t)$ above, which can also be used to determine sub-arguments in our system. See [54, p. 35] for more details.

As it is known from formal argumentation, there are multiple strategies to decide on the acceptability of arguments. Each of these strategies can be considered as a specific semantics that generates sets of acceptable arguments called “extensions”. For example, a more credulous strategy tends to maximize the set of acceptable arguments and it is represented, e.g., by *preferred* extensions, while *grounded* extension represents skeptical strategies. The \mathbf{JT}_{CS} variants of abstract argumentation semantics are used to determine whether a \mathbf{JT}_{CS} formula is valid under a given semantics.

Definition 13 (\mathbf{JT}_{CS} Extensions) We define multiple argumentation theory extensions for any default theory $T = (W, D)$ based on the \mathbf{JT}_{CS} language:

\mathbf{JT}_{CS} -Admissible Extension A potential extension set of \mathbf{JT}_{CS} formulas $\Gamma \subset In(\Pi)$ is a \mathbf{JT}_{CS} -admissible extension of a default theory $T = (W, D)$ iff $Th^{JT_{CS}}(\Gamma)$ is conflict-free, each formula $t : F \in \Gamma$ is acceptable with respect to Γ and Π is a closed process.

\mathbf{JT}_{CS} -Preferred Extension A closure $Th^{JT_{CS}}(\Gamma)$ of a \mathbf{JT}_{CS} -admissible extension Γ is a \mathbf{JT}_{CS} -preferred extension of T iff for any other \mathbf{JT}_{CS} -admissible extension Γ' , $\Gamma \not\subset \Gamma'$.

\mathbf{JT}_{CS} -Complete Extension A closure $Th^{JT_{CS}}(\Gamma)$ of a \mathbf{JT}_{CS} -admissible extension Γ is a \mathbf{JT}_{CS} -complete extension of T iff for each closed process Π of T such that there is a \mathbf{JT}_{CS} -admissible extension Γ' in $In(\Pi)$ and $\Gamma \subset \Gamma'$, if a formula $t : F \in cons(D)$ is acceptable with respect to Γ in $In(\Pi)$, then $t : F$ belongs to Γ .

\mathbf{JT}_{CS} -Grounded Extension A \mathbf{JT}_{CS} -complete extension $Th^{JT_{CS}}(\Gamma)$ is the unique \mathbf{JT}_{CS} -grounded extension of T if Γ is the smallest potential extension with respect to set inclusion such that $Th^{JT_{CS}}(\Gamma)$ is a \mathbf{JT}_{CS} -complete extension of T .

\mathbf{JT}_{CS} -Stable Extension A conflict-free closure $Th^{JT_{CS}}(\Gamma)$ of a potential extension Γ is a \mathbf{JT}_{CS} -stable extension iff for each $\delta \in D$ such that $t : F = cons(\delta)$ and $t : F \notin Th^{JT_{CS}}(\Gamma)$, it holds that (1) Γ undercuts $t : F$ or (2) δ is inapplicable to $In(\Pi)$ for any process Π such that $\Gamma \subset In(\Pi)$.

It turns out that a large subclass of Dung’s abstract argumentation frameworks, excluding only some kinds of attack cycles, can be proven to be a special case of our logic via establishing the correspondences of extensions. Therefore, we can obtain Dung’s frameworks from justification logic-based theories by focusing on a single aspect of attack direction among justification logic arguments. Moreover, Dung’s frameworks, modulo some attack cycles, always have multiple justification logic instantiations called “realizations”. These results are shown in [54]. Throughout the rest of the paper, we only rely on the procedure to obtain an abstract argumentation framework from a default theory T called *Forgetful projection* [54, p. 26].

Notice that, in Dung’s original approach, we cannot represent joint attacks that possibly emerge from the interaction of arguments. Forgetful projection is thus limited to a subclass of default theories such that an argument can only attack other arguments if this attack is a \mathbf{JT}_{CS} consequence of a single consequent formula for an applied default rule. Here, we only informally recall the procedure that suffices for the purposes of specifying an attack graph in our examples. Firstly, we obtain the graph nodes from those defaults that generate potential extension sets, that is, that are applicable to a process of T . Each default rule from D that is applied to a process of T is mapped to a node (an argument) in the corresponding abstract argument framework. Then we specify the structure of attacks among the obtained nodes. For any two defaults such that both defaults are applicable, but never occur together in any process of T , the corresponding nodes or arguments attack each other. Moreover, if a consequent of a default entails an undercutter for some consequent of a default (possibly

the same default), this entailment translates into an asymmetric attack from the argument that corresponds to the default whose consequent entails the undercutter.

The workings of argumentation semantics can be illustrated with a default theory that develops from Example 5 and the argument for the claim that Harry is a British subject.

Example 14 (Continuation of Example 5) Recall that the agent started reasoning from the fact that Harry was born in Bermuda (B) to conclude that Harry is a British subject (S) via the following default:

$$\delta_1 = \frac{r : B :: (s \cdot r) : S}{(s \cdot r) : S}.$$

As mentioned above, the consequent of the default can be read as follows: “ $(s \cdot r)$ is a defeasible reason justifying that Harry is a British subject”.

However, if the agent were in possession of the additional information saying that both Harry’s parents are aliens (P), the “general authority of the warrant” $s : (B \rightarrow S)$ for the claim S would have to be set aside. This is modeled with the following rule that introduces an undercutting reason:

$$\delta_2 = \frac{t : P :: (u \cdot t) : \neg[s : (B \rightarrow S)]}{(u \cdot t) : \neg[s : (B \rightarrow S)]}.$$

The consequent can be read as follows: “ $(u \cdot t)$ is a defeasible reason denying that the reason s justifies that if Harry was born in Bermuda, then Harry is a British subject”. This is a classical argumentation theory example of a defeater that leads to the suspension of the conclusion supported by the reason $(s \cdot r)$. For a default theory $T = (W, D)$ with $W = \{r : B, t : P\}$ and $D = \{\delta_1, \delta_2\}$, the process $\Pi_1 = (\delta_1, \delta_2)$ corresponds to such course of reasoning with revised \mathbf{JT}_{CS} extensions.

In total T has two closed processes, Π_1 and $\Pi_2 = (\delta_2, \delta_1)$. The structure of attacks induced by the undercutting defeat is shown in Fig. 3, where the simple attack graph structure represents arguments $cons(\delta_1)$ and $cons(\delta_2)$ with the nodes A and B , respectively.

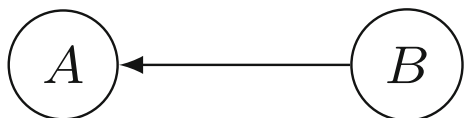
The theory T is well-founded [54, p. 24] and it has a unique \mathbf{JT}_{CS} -complete extension $Th^{\mathbf{JT}_{CS}}(W \cup cons(\delta_2))$, which is also \mathbf{JT}_{CS} -grounded, \mathbf{JT}_{CS} -preferred and \mathbf{JT}_{CS} -stable. It is easy to check that the argumentation framework from Fig. 3 has a corresponding unique complete, grounded, preferred and stable extension, namely, $\{B\}$.

Notice that the warrant underlying δ_2 can also be questioned in a further course of reasoning. For example, one could find out that one of Harry’s parents was settled in Bermuda at the time when he was born (T), which would prompt a new default reason via the rule

$$\delta_3 = \frac{t' : T :: (u' \cdot t') : \neg[u : (P \rightarrow \neg[s : (B \rightarrow S)])]}{(u' \cdot t') : \neg[u : (P \rightarrow \neg[s : (B \rightarrow S)])]}.$$

The consequent reads as “ $(u' \cdot t')$ is a defeasible reason denying that the reason u justifies that if both Harry’s parents are aliens then it is not the case that s justifies that if Harry was born in Bermuda, then Harry is a British subject”. Adding $t' : T$ to W and δ_3 to D would thus reinstate the authority of the warrant used in δ_1 . Formally, this would mean that there is an available default rule that gives you a reason denying that the warrant of δ_2 is true.

Fig. 3 Abstract argumentation framework based on T



The above presented account of default reasons suffices to represent reasoning from an incomplete knowledge base, but it does not represent reasoning with information changes that alter the premises from which an agent starts to reason. Still, the basic account can already model non-monotonic behavior. One type of non-monotonicity is induced by the definition of undercut as exemplified by the undercut of $(s \cdot r)$. We will refer to this way of revising as “inferential” revision, because the argument $(s \cdot r) : S$ is susceptible to attack and potentially defeated due to the fallibility of inference δ_1 . This is characteristic for *default* reasoning. For the argument $(s \cdot r) : S$ to be undermined, we consider a wider Toulminian interpretation of the argument that includes the formula $r : B$ as the data for the argument. Since $r : B$ is in the set W , the only possibility to attack $r : B$ is to remove it from W and to thereby undermine $(s \cdot r) : S$. This kind of attack on arguments is studied under the paradigm of *plausible* reasoning. In this paradigm, arguments are taken to be susceptible to attack due to the uncertainty of their premises. The aim of the current work is to unify the two paradigms in a single logical system.

3 Dynamic operations for default theories: Introducing undermining attack

As mentioned in Section 1, undermining can be interpreted as an attack on the formulas that are considered to be facts, that is, factively justified formulas contained in a set of premises W . In our view, undermining is essentially non-inferential because introducing conflicting information that undermines facts cannot be done with the use of warrants.⁵ For a default theory, these facts are represented by justification logic formulas from the set of premises W and, in constructing a default argument, such formulas can be prerequisites of default rules. A plausible interpretation of undermining defeaters would be that they propose alternative states of facts which ground further reasoning steps. To be able to incorporate factual changes, we need methods from belief revision. Our selection of the belief-revision operations follows the way in which default theories are defined — since the set of premises W is typically finite, it is natural to use operators for sets that do not require closure. Therefore, our choice is to make use of base revision operators [39] instead of the AGM operators [1].

To model changes to default theories, we will use the capacity of default logic to represent two levels of information certainty, roughly corresponding to the distinction between skeptical and credulous reasoning (cf. [7]). The top-level of information certainty is represented by formulas that are included in all extensions. Typical examples of such formulas are those contained in a set of premises W . The lower-level of information certainty is represented by formulas whose status is contingent on whether they become defeated by other available information. Such formulas are typically consequents of default rules. Our goal is to use the two levels and to define dynamic operators that can bring about the changes that fix whether a formula is included in or excluded from all extensions, but also to define *non-prioritizing* operations that leave the status of a formula undecided.

⁵The non-inferential view of information change is also relevant for human interaction. As Hlobil [40] argues, we can believe by accepting testimonies, but we cannot make inferences by merely accepting testimony. Two testimonies that contradict each other are to be, *ceteris paribus*, equally treated and the acceptance of new information is not the same process as inferentially extending the existing (incomplete) information. We will return to this issue in Section 5 where we present an “inferential” reading of undermining compatible with the interpretation of undermining in, e.g. ASPIC+.

To be able to model the dynamics at the two levels of information certainty, we extend the above defined default theories with defaults without warrants, which correspond to Reiter's supernormal defaults, but (possibly) containing justification assertions:

$$\delta_F = \frac{\top :: F}{F}.$$

Standard default rules with justification assertions encode inferential steps supported by warrants. Rules such as δ_F above “are givens” [70, p. 227] and they do not rely on warrants as their reason-based assumptions, which is why we refer to them as “unwarranted” defaults. In contrast to inferential steps, supernormal defaults will be used to represent non-inferential, information-changing actions in which an agent accepts that a formula can be included in (at least) one extension. We will extend sets of defaults with supernormal defaults whenever we represent introducing uncertain information to a theory T or relegate information from W to the status of uncertain information.

Why would we want to make changes only to the lower-level of information certainty or alter a default theory at the level of some, instead of all extensions? Sometimes, an agent has doubts with respect to whether it is safe to include some information or not and, analogously, whether it is safe to remove some information or not. In the standard base revision approach to modeling information change, incoming information is always prioritized over existing information, which is ensured by the success postulate. Consider again Example 14 with the agent reasoning about Harry's eligibility for British nationality. It is possible that, according to the census record, Harry was born in Bermuda and, according to the military record, he was born outside Bermuda. The fact that the agent first collected the census record data and then collected the military record data cannot justify the prioritization of the newly acquired information. If the agent does not know which information source is reliable, the order of data input is irrelevant. In these cases, default logic can avoid the “naive” priority ascription by the use of multiple extensions. The rest of this section gives a solution to the problem of non-prioritized change of default theories, along with the more standard prioritized change. In a case of non-prioritized change, the corresponding dynamic operator uses supernormal defaults with an aim to alter the lower-level of information certainty of a default theory. On our interpretation of undermining attacks, whether undermining fully or partially realizes its defeating potential depends on whether the new information is prioritized or not.

3.1 Default theory expansion

We consider three kinds of change: expansion, contraction and revision. The first kind of change corresponds to learning new information. For example, adding a formula F to a set of premises W can be based on the information provided by some information channel. The formal operation that naively adds new information without checking the joint consistency of the resulting set of beliefs is called “expansion”:

Definition 15 (Expansion) For a default theory $T = (W, D)$ and a formula F , $T_F^+ = (W_F^+, D)$ is the expansion of the default theory T , where W_F^+ is the base expansion of the set W such that $W_F^+ = W \cup \{F\}$.

If the added information results in an inconsistent set W_F^+ , any definable \mathbf{JT}_{CS} extension will be inconsistent. Notice that default theory expansion can already cause non-monotonic behaviour on the level of default theory extensions. For example, if the added formula is

a prerequisite for a default rule with an undercutter for some other default consequent, the new information can result in removing elements from \mathbf{JT}_{CS} extensions of T .

An agent can approach accepting incoming information more cautiously. If the agent accepts new information as a plausible premise, but hesitates to consider it a fact, the change is made to the set of default rules:

Definition 16 (Conservative Expansion) For a default theory $T = (W, D)$ and a formula F , $T_F^\times = (W, D \cup \{\delta_F\})$ is the conservative expansion of the default theory T with F , where $\delta_F = \frac{T::F}{F}$.

Notice that the operation \times opens up a possibility that the formula F is included in all extensions, but it can also be excluded from all extensions. For example, if $\neg F$ is contained in $Th^{JT_{CS}}(W)$, then δ_F is not applicable. The introduced operators have the following properties:

Proposition 17 For a default theory $T = (W, D)$ possibly containing unwarranted default rules and a \mathbf{JT}_{CS} formula F , it holds that

- If F is not a contradiction, then F is contained in each \mathbf{JT}_{CS} extension of the theory T_F^+ .
- If F is not a contradiction and if $\neg F$ is not contained in any \mathbf{JT}_{CS} extension of T , then F is contained in each \mathbf{JT}_{CS} extension of the theory T_F^\times .
- If W is not inconsistent and if $\neg F$ is contained in $Th^{JT_{CS}}(W)$, then F is not contained in any \mathbf{JT}_{CS} extension of the theory T_F^\times .

Proving Proposition 17 is straightforward.

3.2 Default theory contraction

How does an agent discard some information that is no longer considered to be reliable? We will again differentiate between two strategies of discarding information or, more technically, of *contracting* default theories: one aims to remove information when an agent is confident that the information is unreliable and another aims to relegate the status of information reliability to a lower level. In our default theories, this will mean that the first operation removes a formula from all extensions while the second operation leaves the possibility that extensions still contain the formula. One problem we face in removing a formula from all theory extensions is that the base contraction of a set of premises W is necessary, but not sufficient to secure that the formula will not be reintroduced by the application of a default rule. To illustrate the need for such operation, consider that changes in information may cause that a certain source of justification t is denied its reliability as a reason for some formula F .

To deal with this problem, we propose to put restrictions on the application of default rules. The aim of restrictions on application is to prevent an unwanted \mathbf{JT}_{CS} formula to become a part of any default theory extension. For a \mathbf{JT}_{CS} formula R , a default rule $\delta = \frac{F::G}{G}$ is *restrictedly-applicable* to a \mathbf{JT}_{CS} -closed set of formulas Γ according to the restriction R (or *R-applicable* for short) iff:

- $F \in \Gamma$ and
- $\neg G \notin {}_R\Gamma$,

where ${}_R\Gamma = Th^{JTCS}(\Gamma \cup \{R\})$. The role of the restriction R is to prevent the formula $\neg R$ to extend the \mathbf{JTCS} theory Γ by blocking the applicability of each default whose consequent formula would introduce $\neg R$ to the extended theory, if the consequent is added to Γ .

Using the restricted variant of default applicability, we will define restricted variants of default theories and their processes. A corresponding operational semantics, in which we restrict the application of defaults by a \mathbf{JTCS} formula R , can then be defined for those sets of premises W that do not entail $\neg R$.

Definition 18 (Application-Restricted Default Theory) For a \mathbf{JTCS} formula R , an application-restricted default theory ${}_{[R]}T$ is defined as a pair (W, D) , where the set W is a finite set of \mathbf{JTCS} formulas such that $\neg R \notin Th^{JTCS}(W)$ and D is a countable set of default rules.

Application-restricted default theories differ from default theories defined in Section 2.2 only in the view of restrictions that might eliminate some possible ways to build default processes that would otherwise be possible without restrictions. To define processes for an application-restricted theory ${}_{[R]}T$, recall that the *In*-set from Definition 7 is a closed set of \mathbf{JTCS} formulas that represents current evidence base. The *In*-set is now relativized to R in application-restricted default theories: ${}_RIn(\Pi) = Th^{JTCS}(In(\Pi) \cup \{R\})$. Processes of an application restricted theory are built taking into account the restricted notion of applicability, i.e., we require each default of a sequence to be R -applicable (and not only applicable) to $\Pi[k]$. The definition of closed processes is adapted for a theory ${}_{[R]}T$ as follows: an application-restricted process Π is said to be closed iff every $\delta \in D$ that is R -applicable to $In(\Pi)$ is already in Π .

Notice that the definitions of warrant, undercut, acceptability, as well as potential, \mathbf{JTCS} -admissible, \mathbf{JTCS} -preferred, \mathbf{JTCS} -stable, \mathbf{JTCS} -complete and \mathbf{JTCS} -grounded extensions from Section 2.2 all depend on the definitions of *In*-sets and default processes. Therefore, the definitions of these concepts have equivalent formulations for application-restricted default theory pairs (W, D) . In fact, application-restricted default theories are a generalization of default theories from Definition 4. Each default theory $T = (W, D)$ can be defined as an application-restricted default theory ${}_{[\top]}T = (W, D)$, where the restriction formula is a tautology. For the theory ${}_{[\top]}T$, the sets $In(\Pi)$ and ${}_{\top}In(\Pi)$ coincide.

For any application-restricted default theory, the expansion operation $({}_{[F]}T)_G^+$ and the conservative expansion operation $({}_{[F]}T)_G^\times$ are both defined analogously to the corresponding default theory operations. Notice the following exception: expanding an application restricted theory ${}_{[F]}T$ with the formula $\neg F$, that is, $({}_{[F]}T)_{\neg F}^+$. According to Definition 18, an application-restricted theory can only be defined for a set of premises W that does not entail the negation of a restriction formula, so the theory $({}_{[F]}T)_{\neg F}^+$ is not defined. In general, the following holds in such cases of expansion:

- For an application-restricted theory ${}_{[F]}T$ and a \mathbf{JTCS} formula F , if $\neg F \in Th^{JTCS}(W \cup \{G\})$, then the expansion of an application-constrained theory ${}_{[F]}T$ with a formula G is defined as the the default theory T_G^+ .

Therefore, the expansion of ${}_{[F]}T$ with $\neg F$ is the default theory $T_{\neg F}^+$, which possibly contains unwarranted rules. After expanding ${}_{[F]}T$ with $\neg F$, $W_{\neg F}^+$ contains the formula $\neg F$, which means that the resulting theory cannot be application-restricted by the formula F .

We can now define a contraction operation that aims at removing a formula at the level of a whole default theory. The operation corresponds to the action of removing information when an agent is confident that the information is not reliable. To achieve this in a default

theory, a formula has to be removed from the set of premises by a base contraction and its reintroduction should be prevented. In the definition of contraction, *remainder sets* will be used: for any set of \mathbf{JT}_{CS} formulas Γ and a formula F , the remainder set $\Gamma \perp F$ is defined as the set of maximal subsets of Γ that do not entail F .

Definition 19 (Contraction) For a default theory $T = (W, D)$ and a formula F , the application-restricted theory ${}_{[\neg F]}T_F^- = (W_F^-, D \cup D_{\perp F}^-)$ is the contraction of the default theory T by F , where

1. W_F^- is the (full) meet contraction of the set W , defined as $W_F^- = \bigcap(W \perp F)$ and
2. $D_{\perp F}^- = \{\delta_G \mid G \in W \setminus (\bigcap(W \perp F) \cup \{F\})\}$.

Notice that an application-restricted default theory ${}_{[\neg F]}T_F^-$ is definable for any theory T since, due to condition 1, the formula F cannot be an element of the set $Th^{JT_{CS}}(W_F^-)$.

The combination of the restriction $\neg F$ and the set of default rules $D_{\perp F}^-$ provides a balanced solution for avoiding extremely cautious and extremely incautious behavior. Since the set of formulas $W \perp F$ usually contains many elements, theory contraction operations need to include a procedure of selecting the formulas that can be kept after contracting by F , excluding F itself. It is difficult to define such procedures in a principled and intuitively plausible way. In default theory contraction, we do not need to force selection by a function. Instead, the choice of formulas selected upon contraction depends on the type of extension that is being computed. For example, a \mathbf{JT}_{CS} -preferred extension corresponds to the idea of maxichoice contraction, while \mathbf{JT}_{CS} -grounded extension corresponds to the idea of full meet contraction [39, pp. 12-13].

Using again the two-leveled perspective on changing default theories, we can define a more conservative way of giving up a belief. In conservative contraction, agents are reluctant to entirely give up on some information, but the information is no longer considered to be a fact. To relegate the status of a formula in such a way within a default theory, the formula is removed from the set of premises W and then reintroduced through application of a supernormal default rule.

Definition 20 (Conservative Contraction) For a default theory $T = (W, D)$ and a formula F , $T_F^{\ddagger} = (W_F^-, D \cup D_{\perp F})$ is the conservative contraction of the default theory T by F , where

1. W_F^- is the (full) meet contraction of the set W , defined as $W_F^- = \bigcap(W \perp F)$ and
2. $D_{\perp F} = \{\delta_G \mid G \in W \setminus \bigcap(W \perp F)\}$.

By analogy with T_F^{\ddagger} , we define the conservative contraction ${}_{[\neg F]}T_F^{\ddagger}$ that realizes the same set of \mathbf{JT}_{CS} extensions as the theory ${}_{[\neg F]}T_F^-$.

Conservative contraction is an open-ended operation in the sense that it does not preclude the possibility of reintroducing a formula F in an extension through a default rule application. Furthermore, it leaves open the possibility that F occurs in all extensions of the resulting default theory. Recall that, analogous to conservative contraction, conservative expansion might not guarantee that there will be any extension containing a formula F , after a default theory has been conservatively expanded with F . Antoniou [7, p. 1149] takes a different approach in defining a more conservative contraction operation for Reiter's default theories. His idea is to secure that there is at least one extension added that does not contain the formula removed from the set of premises. In our view, it is unnecessary to have such an operation. If some formula is not regarded to be a fact, but it is still plausible that the formula is true, accepting it as the only available information might be the only

reasonable action. Instead of “forcing” an extension without the formula, conservative contraction enables the possibility of an extension without the formula. If there is no support for the contrary statement whatsoever, an agent might still need to hold on to the only available information. The following statement follows from Definition 19:

Proposition 21 *For a default theory $T = (W, D)$ possibly containing unwarranted default rules and a non-tautological \mathbf{JT}_{CS} formula F , it holds that F is not contained in any \mathbf{JT}_{CS} extension of the theory ${}_{[\neg F]}T_F^-$.*

Proof From the condition 1 of the definition, we know that the full meet contraction of W removes the formula F from the set of premises of ${}_{[\neg F]}T_F^-$. Moreover, from condition 2 and the fact that the application of each default rule from $D_{\uparrow F}$ in ${}_{[\neg F]}T_F^-$ is restricted by $\neg F$, we know that F cannot be reintroduced into an evidence base $In(\Pi)$ by applying defaults from a process Π for any process Π of ${}_{[\neg F]}T_F^-$. Therefore, F cannot be contained in any \mathbf{JT}_{CS} extension of ${}_{[\neg F]}T_F^-$. □

3.3 Default theory revision

The task of adding new information to the set of premises by the expansion operation (Definition 15) can lead to an inconsistent set of premises. A more realistic dynamic operator for adding information needs to specify a process by which an agent adds information inconsistent with W without being committed to an inconsistent set of premises. One possible way is to only add information via the conservative expansion operation (Definition 16), but this comes with an obvious flaw: an agent is not able to confidently replace an old, unreliable piece of information with a new, reliable one. This is one of the motivations to define a default theory revision operator that not only adds a formula, but also removes inconsistent formulas at one of the two levels of the default theory.

A revision operation can be defined from a combination of the expansion and contraction operations.⁶ In our approach, we will follow the traditional arrangement of the operations as proposed by Hansson [39, p. 203], namely, removing formulas will precede adding a formula. Those revision operations in which contraction is followed by expansion are called “internal revision” operators. The choice of a revision operation used for a particular revision example depends on both whether old information is to be removed confidently or conservatively and whether new information is to be added confidently or conservatively. We define the following four internal revision operators for each combination of the operations.

Definition 22 (Revision Operators) For a default theory $T = (W, D)$ and a formula F , (internal) revision operators for T are defined as follows:

1. $T_F^{\pm} = ({}_{[F]}T_{\neg F}^-)_F^+$
2. $T_F^{\times} = ({}_{[F]}T_{\neg F}^-)_F^{\times}$
3. $T_F^{\dagger} = (T_{\neg F}^{\div})_F^{\dagger}$
4. $T_F^{\ast} = (T_{\neg F}^{\div})_F^{\ast}$

⁶If we were to exhaust all possible combinations, eight revision operators could be defined. Note that the revision operation symbols used here reflect the composition of the introduced revision operations that are defined in terms of contraction and expansion variants. The symbols are not intended to be in continuity with the standard usage of revision operation symbols.

The variety of possible revision operators raises the question about what kinds of revision strategies they represent.

We can show that the four operations amount to two strategies. Again, as in the cases of expansion and contraction, one strategy is meant to revise confidently and the other strategy to revise more conservatively. The key to show this is to prove that the operations \mp , \times and \dagger give equivalent extensions in revising a default theory with some formula F . This is the result that we will prove shortly, based on the a) clause of Proposition 23:

Proposition 23 *For a default theory $T = (W, D)$ possibly containing unwarranted default rules and a \mathbf{JT}_{CS} formula F , it holds that*

- If F is not a contradiction, then for all \mathbf{JT}_{CS} extensions Γ of the theories T_F^\mp , T_F^\times and T_F^\dagger , it holds that $F \in \Gamma$.*
- If F is not a contradiction, then there is a \mathbf{JT}_{CS} extension Γ of the theory T_F^\times such that $F \in \Gamma$.*
- If F is not a contradiction, then there is a \mathbf{JT}_{CS} extension Γ of the theory T_F^\times such that $\neg F \notin \Gamma$.*

Proof To prove that a) holds, consider the three revision operators \mp , \times and \dagger and the resulting theories T_F^\mp , T_F^\times and T_F^\dagger . For the case of the default theory T_F^\mp , it follows from Proposition 21 that $\neg F$ is not contained in any \mathbf{JT}_{CS} extension of ${}_{[F]}T_{\neg F}^-$. By Proposition 17 a), F is contained in each \mathbf{JT}_{CS} extension of $({}_{[F]}T_{\neg F}^-)_F^\mp$.

For the case of the default theory T_F^\times , it follows from Proposition 21 that $\neg F$ is not contained in any \mathbf{JT}_{CS} extension of ${}_{[F]}T_{\neg F}^-$. Moreover, the set ${}_F In(\Pi)$ contains the formula F and it is not \mathbf{JT}_{CS} -inconsistent, which means that the default rule $\frac{\top::F}{F}$ is restrictedly-applicable to any \mathbf{JT}_{CS} extension of the conservative expansion $({}_{[F]}T_{\neg F}^-)_F^\times$ of the theory ${}_{[F]}T_{\neg F}^-$. Therefore, F is contained in each \mathbf{JT}_{CS} extension of $({}_{[F]}T_{\neg F}^-)_F^\times$.

For the case of the default theory T_F^\dagger , consider that the base contraction of W ensures that $\neg F$ cannot be contained in the set of premises $W_{\neg F}^-$ of the default theory $T_{\neg F}^\dagger$, but $\neg F$ can still be reintroduced by applying the defaults from $D_{\neg F}$. However, after expanding the theory $T_{\neg F}^\dagger$ by F , the inclusion of the formula $\neg F$ into any \mathbf{JT}_{CS} extension of the theory T_F^\dagger is blocked and, by Proposition 17 a), F is contained in each \mathbf{JT}_{CS} extension of T_F^\dagger .

To prove that b) holds, consider that the base contraction of W ensures that $\neg F$ cannot be contained in the set of premises $W_{\neg F}^-$ for the conservative contraction $T_{\neg F}^\dagger$. This means that, for the conservative expansion $(T_{\neg F}^\dagger)_F^\times$, it holds that the default rule $\frac{\top::F}{F}$ is applicable to $Th^{\mathbf{JT}_{CS}}(W)$ and, therefore, contained in at least one \mathbf{JT}_{CS} extension of T_F^\times .

To prove c), consider that after the base contraction of W by $\neg F$, \mathbf{JT}_{CS} extensions of $T_{\neg F}^\dagger$ are \mathbf{JT}_{CS} -consistent. Since we also know that after the conservative expansion $(T_{\neg F}^\dagger)_F^\times$, b) holds, then c) holds. \square

To show the equivalence of the operators \mp , \times and \dagger , we first say that for any σ -extension, where

$$\sigma \in \{\mathbf{JT}_{CS}\text{-admissible, } \mathbf{JT}_{CS}\text{-complete, } \mathbf{JT}_{CS}\text{-grounded, } \mathbf{JT}_{CS}\text{-preferred, } \mathbf{JT}_{CS}\text{-stable}\},$$

$\sigma(T)$ is the set of all σ -extensions for a theory T . Then we prove that for any default theory T , the default theories T_F^\mp , T_F^\times and T_F^\dagger realize the same set of extensions under any **JTCS** extension-based semantics for default theories. The following result is obtainable from Proposition 23 a), together with the fact that none of the three operators \mp , \times and \dagger change the status of formulas that do not take part in F -implying sets:

Theorem 24 *For any default theory $T = (W, D)$, a **JTCS** formula F and the (internal) revision operators \mp , \times and \dagger , it holds that $\sigma(T_F^\mp) = \sigma(T_F^\times) = \sigma(T_F^\dagger)$.*

Intuitively, the three operations represent a type of revision in which an agent confidently includes new and possibly inconsistent information into all **JTCS** extensions. Another option specified by the operator \times is to accept the new information in some extensions while maintaining the old information in other extensions. The revision operators comply to the two-leveled view of default semantics: the first three revision operators of Definition 22 fix the status of a revision at the level of a default theory as a whole, while the last revision operator targets at modifying only some extensions. Any of the three operations T_F^\mp , T_F^\times and T_F^\dagger will be referred to as the *Revision* of T with F and the operation T_F^* will be referred to as the *Conservative Revision* of T with F .

3.4 The notion of undermining

Finally, we are now able to say in what way the dynamic operations connect to the notion of undermining defeat. It was mentioned in the Introduction that by undermining we understand the attack whereby argument premises are being questioned. This intuition can now be cashed out by using those dynamic operators for default theories that involve contracting a default theory.

Definition 25 (Undermining) *For a default theory $T = (W, D)$ and a **JTCS** formula F such that $F \in W$ and $F = pre(\delta)$ for some $\delta \in D$, F is undermined iff W is contracted by F by applying any of the following operations to T :*

1. $[\neg F]T_F^-$ (Contraction)
2. T_F^\ddagger (Conservative Contraction)
3. T_G^\mp , T_G^\times or T_G^\dagger for **JTCS**-inconsistent formulas F and G (Revision)
4. T_G^* for **JTCS**-inconsistent formulas F and G (Conservative Revision).

Notice that there is no requirement on the structure of F . However, each meaningful undermining targets justification assertions because W cannot be successfully contracted by a tautology and justification assertions are the only other type of formula occurring as a default prerequisite. Not every attack on the premises results in confidently revising the set W . It is possible that undermining leaves an agent undecided as to whether newly acquired information or older information should be prioritized.

Starting from the theory T defined in Example 14, we can give a formalized undermining example to show the difference between inferential and non-inferential ways of information acquisition. Recall that the agent started to reason from the information that Harry was born in Bermuda. This piece of information is represented in the set of premises W with the formula $r : B$, where r can now be taken to reflect the source of information as, e.g., data

from census records.⁷ However, if the information based on military records says that Harry was born outside Bermuda, and having no means to resolve this conflict of information, the theory T needs to be conservatively revised. The theory $T_{v:\neg B}^*$ is the revision of T with the formula $v : \neg B$, where v reflects the new source of information for the claim that Harry was not born in Bermuda. To see this revision in more detail, we extend Example 14.

Example 26 (Continuation of Example 14) We start from the theory $T = (W, D)$ that consists of the set of premises $W = \{r : B, t : P\}$ and the set of defaults $D = \{\delta_1, \delta_2\}$ with

$$\delta_1 = \frac{r : B :: (s \cdot r) : S}{(s \cdot r) : S} \text{ and } \delta_2 = \frac{t : P :: (u \cdot t) : \neg[s : (B \rightarrow S)]}{(u \cdot t) : \neg[s : (B \rightarrow S)]}.$$

The first dynamic operation in revising with $v : \neg B$ is contracting the theory by $\neg v : \neg B$. The resulting theory $T_{\neg v:\neg B}^\dagger = (W_{\neg v:\neg B}^-, D \cup D_{\neg v:\neg B})$ consists of the set of premises $W_{\neg v:\neg B}^- = \{t : P\}$ and the set of defaults D extended with the default

$$\delta_{r:B} = \frac{\top :: r : B}{r : B}.$$

Finally, the agent conservatively expands the theory $T_{\neg v:\neg B}^\dagger$ with the information that Harry was not born in Bermuda. The new default theory is defined as $T_{v:\neg B}^* = (T_{\neg v:\neg B}^\dagger)_{v:\neg B}^\times$. The change of the theory after conservative expansion with $v : \neg B$ amounts to adding the new default rule

$$\delta_{v:\neg B} = \frac{\top :: v : \neg B}{v : \neg B},$$

which means that the new set of defaults is $D \cup D_{\neg v:\neg B} \cup \{\delta_{v:\neg B}\}$. The revised theory $T_{v:\neg B}^*$ changes the default processes in which the agent reasons about Harry’s nationality, as Fig. 4 illustrates. In total, the theory $T_{v:\neg B}^*$ has five closed processes.

The number of closed processes also indicates that the attack relation is now more complex, compared to the attack relation based on the starting theory T . As a result, we obtain the structure of attacks in Fig. 5. In Fig. 5, arguments $(s \cdot r) : S$ and $(u \cdot t) : \neg[s : (B \rightarrow S)]$ are represented with the nodes A and B . After the conservative revision of T with $v : \neg B$, the space of arguments is enlarged. The new arguments $r : B$ and $v : \neg B$ are represented with the nodes C and D . The nodes C and D attack each other because $r : B$ and $v : \neg B$ are mutually \mathbf{JT}_{CS} -inconsistent arguments that cannot co-occur as a part of the same In -set for $T_{v:\neg B}^*$.

Notice that the symmetric attack between A and D results from the fact that $(s \cdot r) : S$ and $v : \neg B$ cannot together extend an evidence base $In(\Pi)$ for any process Π of $T_{v:\neg B}^*$, even though these two arguments are not \mathbf{JT}_{CS} -inconsistent. Nonetheless, it holds that an evidence base that contains $(s \cdot r) : S$ is \mathbf{JT}_{CS} -inconsistent with an evidence base that contains $v : \neg B$. This exemplifies the importance of tracking \mathbf{JT}_{CS} -inconsistent evidence bases in mapping a default theory to a corresponding abstract argumentation output.

Finally, we look into the changes in the structure of computable \mathbf{JT}_{CS} extensions, compared to Example 14 before undermining took place. Unlike T , the theory $T_{v:\neg B}^*$ is not

⁷Technically, r and other justifications that we call “sources” are some specific justifications produced without “default application” inference steps. Each information source might be interpreted as a concatenation of more basic pieces of evidence by applying the sum operation. For example, r might be thought of as a concatenation of all the instances of entries from the census records $(r_1 + r_2 + \dots + r_n)$, where r_1, r_2, \dots , and r_n are formally justification constants. This interpretation could be fully formalized in \mathbf{JT}_{CS} by allowing for constants that justify not only basic logic axioms, but also contingent propositions, as done in [13].

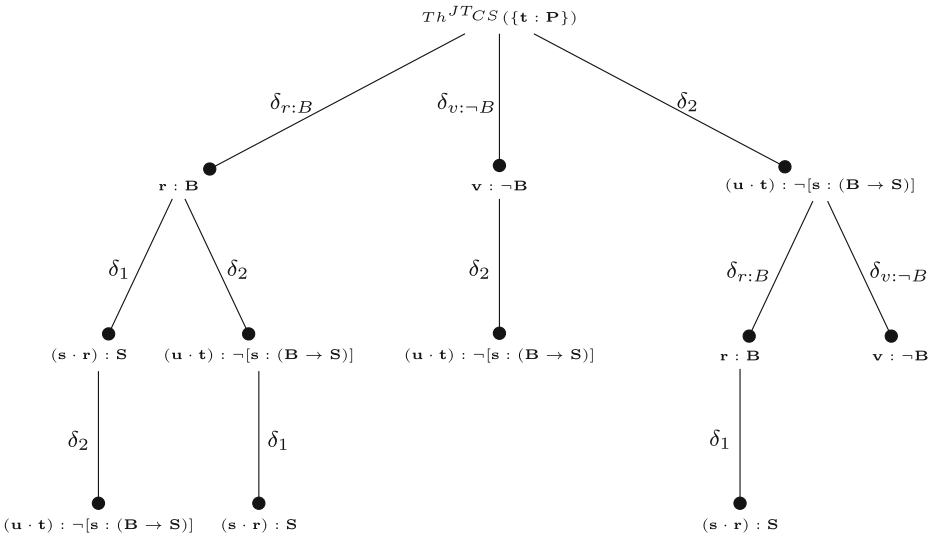


Fig. 4 The process tree shows all the processes of $T_{v:\neg B}^*$ as the branches of the tree. Each node of the tree is labeled with an *In*-set after a default rule (edges) has been applied, but in such a way that we display only the formulas that are added to *In*-sets as a result of applying the available defaults

well-founded. This is expected, as the added unresolved symmetric attacks generate multiple complete extensions. Table 1 displays all the possible \mathbf{JT}_{CS} extensions. It can be checked that they correspond to the extensions obtainable from the abstract argumentation framework in Fig. 5.

To illustrate how conservative revision differs from revision, consider an alternative scenario in which the agent prioritizes the new information based on military records and the

Fig. 5 Abstract argumentation framework corresponding to the default theory $T_{v:\neg B}^*$

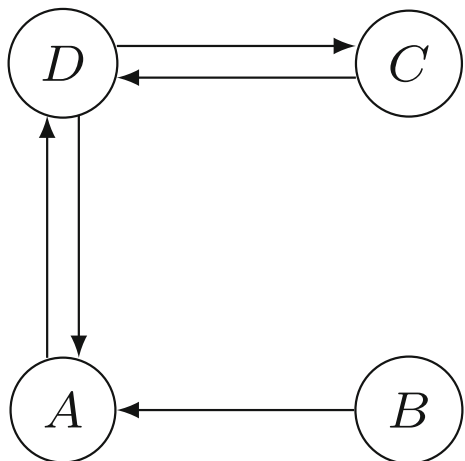


Table 1 All the \mathbf{JT}_{CS} extensions of $T_{v:\neg B}^*$

| | | | | |
|--------------------------------|------------|--------------------------|--------------------------|--------------------------|
| \mathbf{JT}_{CS} -admissible | Γ_1 | Γ_2 | Γ_3 | Γ_4 |
| \mathbf{JT}_{CS} -complete | | $Th^{JT_{CS}}(\Gamma_2)$ | $Th^{JT_{CS}}(\Gamma_3)$ | $Th^{JT_{CS}}(\Gamma_4)$ |
| \mathbf{JT}_{CS} -grounded | | $Th^{JT_{CS}}(\Gamma_2)$ | | |
| \mathbf{JT}_{CS} -preferred | | | $Th^{JT_{CS}}(\Gamma_3)$ | $Th^{JT_{CS}}(\Gamma_4)$ |
| \mathbf{JT}_{CS} -stable | | | $Th^{JT_{CS}}(\Gamma_3)$ | $Th^{JT_{CS}}(\Gamma_4)$ |

$\Gamma_1 = W, \Gamma_2 = W \cup \{cons(\delta_2)\}, \Gamma_3 = \Gamma_2 \cup \{cons(\delta_{r:B})\}, \Gamma_4 = \Gamma_2 \cup \{cons(\delta_{v:\neg B})\}$

agent is confident that the census records data should be given up. In this case, the theory T has to be revised with the formula $v : \neg B$ saying that Harry was born outside Bermuda. The resulting default theory is $T_{v:\neg B}^\mp$ defined as $([v:\neg B]T_{\neg v:\neg B}^-)_{v:\neg B}^+$. That is, T is first confidently contracted by $\neg v : \neg B$. The obtained application-restricted theory $[v:\neg B]T_{\neg v:\neg B}^-$ is then expanded with $v : \neg B$. In $T_{v:\neg B}^\mp$, the set of premises $(W_{\neg v:\neg B}^-)_{v:\neg B}^+$ now contains the formula $v : \neg B$ with δ_2 as the only rule that is applicable. The set of defaults $D_{\neg v:\neg B}^-$ also contains the rules δ_1 and $\delta_{r:B}$ that cannot be applied to any process of $T_{v:\neg B}^\mp$. Therefore, the only closed process of $T_{v:\neg B}^\mp$ is a result of applying δ_2 to $(W_{\neg v:\neg B}^-)_{v:\neg B}^+$. We are then left with a simple way to determine the extensions: $T_{v:\neg B}^\mp$ is well founded and $Th^{JT_{CS}}((W_{\neg v:\neg B}^-)_{v:\neg B}^+ \cup \{cons(\delta_2)\})$ is its unique \mathbf{JT}_{CS} -complete, \mathbf{JT}_{CS} -grounded, \mathbf{JT}_{CS} -preferred and \mathbf{JT}_{CS} -stable extension.

4 Trivialization in structured argumentation and argumentative paradoxes in justification logic

The logic studied in this paper distinguishes between strict rules (rules without exceptions) of inference and defeasible rules (rules that may have exceptions) of inference. Strict rules are determined by the \mathbf{JT}_{CS} axioms and rules. Defeasible inference rules are determined by default rules. It is common in formal argumentation frameworks with rule-based languages to distinguish between strict rules and defeasible rules as, for example, in ASPIC+ [48]. In such frameworks, arguments are built according to the available strict and defeasible rules. Recently, a number of authors have pointed out that using any “explosive” logic as the basis of strict rules may lead to undesired problems of trivialization of argumentation frameworks due to the principle *Ex falso sequitur quodlibet*.

In this section, we discuss why using our operational semantics to model rebutting and undermining attacks avoids the problem of trivialization. We first introduce Caminada’s examples [19] that motivate the trivialization problem. Then we provide an alternative formalization of these examples in default justification logic. Finally, we prove that default justification logic satisfies the postulate of non-interference (and, therefore, crash resistance) that specify conditions under which argumentation frameworks avoid trivialization.

4.1 Trivializing argumentation systems

To illustrate how the *Ex falso* principle trivializes argumentation frameworks, we first define the concept of a defeasible theory based on propositional logic, an instance of explosive logics.

Definition 27 (Defeasible theory) For the propositional logic language \mathbf{L} , a defeasible theory DT is a triple (P, RD, n) , where P is a (finite and consistent) set of propositional formulas and RD is a countable set of defeasible rules of the following form

$$\varphi_1, \dots, \varphi_n \Longrightarrow \psi,$$

where $\varphi_1, \dots, \varphi_n$ and ψ are meta-variables ranging over wff-s in \mathbf{L} , and n is a “rule-naming” function such that:

$$n : RD \rightarrow \mathbf{L}.$$

The following informal reading of the rule above is given by Caminada [19] and Wu and Podlaszewski [72]: “If $\varphi_1, \dots, \varphi_n$ hold, then probably ψ holds”.

Based on a defeasible theory $DT = (P, RD, n)$, we define an argumentation system AS as a triple (RS, RD, n) , where $RS = \{\rightarrow \varphi \mid \varphi \in P\} \cup \{\varphi_1, \dots, \varphi_n \rightarrow \psi \mid \varphi_1, \dots, \varphi_n \vdash_{\mathbf{L}} \psi\}$. Intuitively, RS is the set of rules without exceptions based on classical reasoning, while RD is the set of rules with exceptions. The function n makes it possible that defeasible rules refer to one another, which is essential to the way in which undercutters are modeled.⁸

We can now illustrate the problem of trivialization by the following example from [19, pp. 2724-2725].

Example 27 ([19]) Consider the defeasible theory $DT_1 = (P_1, RD_1, n_1)$, where $P_1 = \{wfr, js, mns\}$ and $RD_1 = \{wfr \Longrightarrow r, js \Longrightarrow s, mns \Longrightarrow \neg s\}$. The following interpretation is given for propositional formulas: let wfr be the proposition “the weather forecaster predicts rain tomorrow”, let js be the proposition “John says that the cup of coffee contains sugar”, let mns be the proposition “Mary says that the cup of coffee does not contain sugar”, let r be the proposition “It will rain tomorrow” and let s be the proposition “The cup of coffee contains sugar”. Throughout the current example, we assume an empty naming function, that is, $n_1 = \emptyset$.

Based on the defeasible theory DT_1 , we define the argumentation system $AS_1 = (RS_1, RD_1, n_1)$ with $RS_1 = \{\rightarrow wfr, \rightarrow js, \rightarrow mns\}$. The argumentation system AS_1 gives the following arguments:

$$\begin{array}{ll} A_1 : \rightarrow wfr & A_4 : A_1 \Longrightarrow r \\ A_2 : \rightarrow js & A_5 : A_2 \Longrightarrow s \\ A_3 : \rightarrow mns & A_6 : A_3 \Longrightarrow \neg s \end{array}$$

Additionally, propositional logic gives the strict rule $s, \neg s \rightarrow \neg r$, because it holds that $s, \neg s \vdash_{\mathbf{L}} \neg r$. This allows us to construct the following argument:

$$A_7 : A_5, A_6 \rightarrow \neg r.$$

Notice that the arguments A_5 and A_6 rebut each other, because their conclusions are inconsistent. It is also the case that argument A_5 rebuts argument A_7 , because the conclusion s of A_5 is inconsistent with the conclusion $\neg s$ of A_6 , which is a sub-argument of A_7 . Analogously, argument A_6 rebuts argument A_7 , because the conclusion $\neg s$ of A_6 is inconsistent

⁸For more technical details on how to construct arguments from strict and defeasible rules in an argumentation system and how arguments attack one another, we refer to [19, pp. 2710-2711]. Caminada [19] also specifies a partial preordering on RD , both for DT and AS . The current presentation allows us to discuss the problem of trivialization, without going into more technical details and without specifying ordering among defeasible rules.

with the conclusion s of A_5 , which is a sub-argument of A_7 . Finally, classical logic allows us to build the argument A_7 whose conclusion $\neg r$ rebuts the conclusion of the argument A_4 . The other way around, namely, that A_4 rebuts A_7 , does not hold. The conclusion of the argument A_7 is based on a strict rule and, therefore, A_7 is not defeasible on its “top rule” whereby we conclude that $\neg r$ holds (see [19, p. 2711] for technical details of the definition of rebuttal). The entire structure of attacks can be represented by the abstract argumentation framework from Fig. 6. It can be noticed from the figure that applying grounded semantics to the framework from Fig. 6 gives the grounded extension $\{A_1, A_2, A_3\}$, which does not include the argument A_4 .

The problem of trivialization is exemplified by the rebuttal of A_4 with the argument A_7 . Informally, the problem with the argument A_7 is that the question of whether to accept that “the cup of coffee contains sugar” or not should not decide on the acceptability of an unrelated conclusion, such as the conclusion “it will rain tomorrow” of the argument A_4 .

By the end of this section, we argue that our operational semantics is fully immune to the problem of trivialization. Before showing that, we first discuss some important differences between structured argumentation frameworks, such as the above discussed argumentation systems that define defeasible and strict rules over a propositional knowledge base, and default justification logic. Default justification logic models arguments as object-level formulas of the \mathbf{JT}_{CS} language. This brings multiple advantages. One of them is the ability to “hard-code” all the reasoning steps that were taken in building arguments, as well as attacks among arguments, in the structure of reason terms. In this paper, we focus on interpreting reason terms as sources of information that could be questioned by other sources. Notice that in Example 28, the argumentation system AS_1 does not yield any opposition between the arguments A_2 and A_3 , as can be seen in the corresponding Fig. 6. However, if John says that the cup of coffee contains sugar and Mary says that the cup of coffee does not contain sugar, there is, intuitively, an immediate opposition between the two testimonies. This opposition is more straightforward than the type of opposition between, for example, two conclusions of default inferences.

Recall from Section 3 that two testimonies that contradict each other are to be, *ceteris paribus*, equally treated. We argue there that the acceptance of new information based on a source of information is not the same process as inferentially extending incomplete information. While the latter process belongs to the default reasoning paradigm, the former process belongs to the plausible reasoning paradigm. In plausible reasoning, we focus on uncertain premises, rather than on uncertain inferences. In Example 28, A_2 represents the argument with the conclusion that “John says that the cup of coffee contains sugar”. This argument is a sub-argument of another argument, namely A_5 , with the conclusion that “the cup of coffee

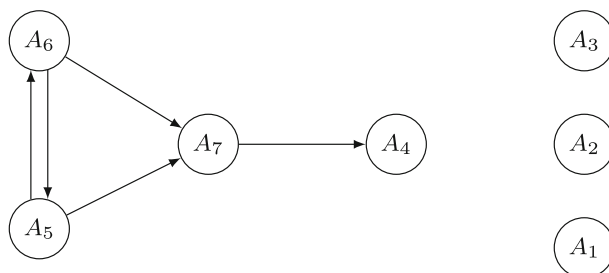


Fig. 6 Abstract argumentation framework corresponding to the argumentation system AS_1

contains sugar”. But the conclusion of A_5 is not inferentially extending incomplete information, as default rules do, but rather restating what is already entailed by John’s testimony. The conflict between John’s and Mary’s testimonies is in all the relevant aspects analogous to the conflict between the census record data and the military record data on whether Harry was born in Bermuda or not. Following that example, we will give a justification logic formalization of Example 28.

Example 29 Suppose that we start to reason from the information that the weather forecaster says that it is expected to rain tomorrow and that John says that the cup of coffee contains sugar. Let E be the proposition “it is expected to rain tomorrow”, let R be the proposition “it will rain tomorrow” and let S be the proposition “the cup of coffee contains sugar”. Moreover, let w represent the weather forecaster’s report as a source of information and let j represent John’s testimony as a source of information. We define the default theory $T_2 = (W_2, D_2)$ such that $W_2 = \{w : E, j : S\}$ and $D_2 = \{\delta_3\}$, where⁹

$$\delta_3 = \frac{w : E :: (v \cdot w) : R}{(v \cdot w) : R}.$$

Additionally, we learn that Mary says that the cup of coffee does not contain sugar. We take the reason term m to represent Mary’s testimony as a source of information. The opposition between Mary’s and John’s testimonies is modeled as an undermining attack, without using rebutting attacks, as done in [72] and [19]. In the situation where we do not prioritize John’s testimonies over Mary’s testimonies or *vice versa*, we need to conservatively contract the theory T_2 with the formula $m : \neg S$. The first step is contracting the theory by $\neg m : \neg S$. This gives us the theory $T_{2-\neg m:\neg S} = (W_{2-\neg m:\neg S}, D_2 \cup D_{1-\neg m:\neg S})$ with the set of premises $W_{2-\neg m:\neg S} = \{w : E\}$ and the set of defaults D_2 extended with the default

$$\delta_{j:S} = \frac{\top :: j : S}{j : S}.$$

Removing $j : S$ from the set of premises is followed by a conservative expansion of the theory $T_{2-\neg m:\neg S}$ with the information that Mary says that the cup of coffee does not contain sugar. The new default theory is defined as $T_{2-\neg m:\neg S}^* = (T_{2-\neg m:\neg S})_{m:\neg S}^\times$. According to the conservative expansion with $m : \neg S$, we add the default rule

$$\delta_{m:\neg S} = \frac{\top :: m : \neg S}{m : \neg S}.$$

Finally, the resulting set of defaults is $D_2 \cup D_{1-\neg m:\neg S} \cup \{\delta_{m:\neg S}\}$.

It can be checked that the rule δ_3 is applicable to all processes, regardless of the conflict between the reasons j and m that is produced by applying the rules $\delta_{j:S}$ and $\delta_{m:\neg S}$. The latter two rules will never be applicable to a same default process. This prevents the possibility of producing a default justification logic analogue to the spurious argument A_7 from Example 28. A similar characterization holds for rebutting attacks. Rebutting reasons are always contained in separate default processes and, therefore, they cannot be combined to

⁹Notice that we keep the inference from the datum that “it is expected to rain tomorrow” to the conclusion that “it will rain tomorrow”. Unlike testimonies about the cup of coffee, this inference is a proper default extension beyond the initial information, because predictions and expectations are not themselves factual claims.

produce arguments based on the *Ex falso sequitur quodlibet* principle. The problem of trivialization caused by the *Ex falso* principle from Example 28 affects structured argumentation frameworks with strict and defeasible rules, but it does not affect default justification logic.

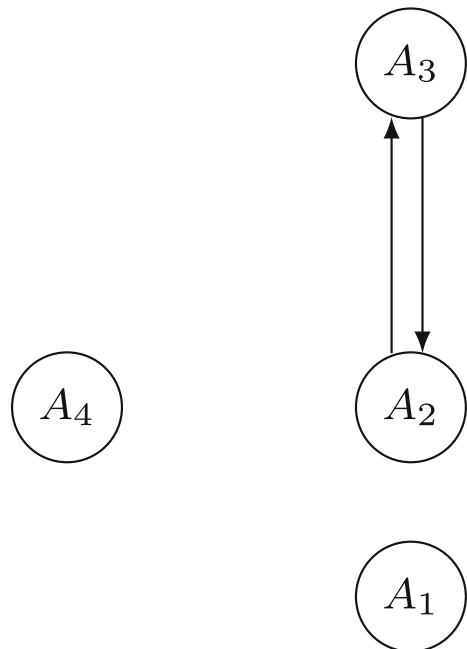
In the case of Example 29, the structure of attacks that can be obtained from the justification logic default theory $T_{2m:\neg S}^*$ is simpler than the structure represented by Fig. 6. This structure is illustrated by the abstract argumentation framework in Fig. 7.

In default justification logic two testimonies introduced by $\delta_{j:S}$ and $\delta_{m:\neg S}$ attack each other. This is captured by the symmetric attack between the arguments A_2 and A_3 that correspond to $j : S$ and $m : \neg S$, respectively. Notice that the “weather forecast” arguments A_1 and A_4 , that is, $w : E$ and $(v \cdot w) : R$, remain unquestioned by the conflict between John’s and Mary’s testimonies. Such an outcome matches our common intuitions behind the example. Accordingly, we can check that the argument $(v \cdot w) : R$ is not compromised and that it is contained in each \mathbf{JT}_{CS} variant of the complete, preferred, grounded and stable extensions defined for the argumentation framework in Fig. 7.

4.2 Argumentative paradoxes

Trivialization is recognized as a problem for structured argumentation frameworks such as ASPIC+. In response to the problem, Prakken [56, p. 111] suggested that preferred and stable semantics could avoid the problem. For example, it can be easily checked that the argument A_4 is contained in each preferred extension of the framework displayed in Fig. 6, but not in its grounded extension. Wu and Podlaszewski [72] and Caminada [19] extend Example 28 in such a way that it creates problems for all the standard semantics. We will now present this extension and discuss some further important points at which our logical approach diverges from structured argument frameworks in representing arguments.

Fig. 7 Abstract argumentation framework corresponding to the default theory $T_{2m:\neg S}^*$



Interestingly, the extended version of Example 28 leads us to an argumentative intensional paradox in justification logic similar to the intensional version of the “Liar paradox” [59].

Example 30 (Continuation of Example 28) Let the defeasible theory $DT_2 = (P_2, RD_2, n_2)$ be defined as follows: $P_2 = P_1 \cup \{junrel, munrel\}$ and $RD_2 = RD_1 \cup \{junrel \implies \neg jrel, munrel \implies \neg mrel\}$, $n_2(js \implies s) = n_2(junrel \implies \neg jrel) = jrel$ and $n_2(mns \implies \neg s) = n_2(munrel \implies \neg mrel) = mrel$. In the extended defeasible theory DT_2 , we include the information that “John says that John is unreliable” and that “Mary says that Mary is unreliable”, where the first piece of information is formally captured with the propositional formula $junrel$ and the second one with the propositional formula $munrel$. The extended set of defeasible rules RD_2 contains the rules $junrel \implies \neg jrel$ and $munrel \implies \neg mrel$ that read as “If John says that John is unreliable, then probably John is not reliable” and “If Mary says that Mary is unreliable, then probably Mary is not reliable” respectively. But if John is not reliable, then the fact that John says something is not a reason to believe what he is saying. This means, according to [72] and [19], that both the rule $js \implies s$ and the rule $junrel \implies \neg jrel$ are undercut. To capture this formally, the conclusion $\neg jrel$ (or “John is not reliable”) of the rule $junrel \implies \neg jrel$ is also the negation of the name of both the defeasible rule $js \implies s$ and $junrel \implies \neg jrel$ according to the rule-naming function n_2 . Analogously, $mns \implies \neg s$ and $munrel \implies \neg mrel$ are both undercut as a result of concluding that $\neg mrel$ or that “Mary is not reliable”, which is also the negation of the name of the two defeasible rules according to n_2 .

Based on the defeasible theory DT_2 , we define the argumentation system $AS_2 = (RS_2, RD_2, n_2)$ with $RS_2 = RS_1 \cup \{\longrightarrow junrel, \longrightarrow munrel\}$. The argumentation system AS_2 allows us to construct all the arguments $A_1, A_2, A_3, A_4, A_5, A_6$ and A_7 , as well as the following arguments:

$$\begin{array}{ll}
 A_8 : \longrightarrow junrel & A_{10} : A_8 \implies \neg jrel \\
 A_9 : \longrightarrow munrel & A_{11} : A_9 \implies \neg mrel
 \end{array}$$

With the added undercutting attacks, AS_2 yields the abstract argumentation framework in Fig. 8. There is a unique complete extension, namely $\{A_1, A_2, A_3, A_8, A_9\}$, which is also preferred and grounded, for the abstract argumentation framework in Fig. 8 and for the underlying argumentation system AS_2 .

In addition to overlooking the immediate conflict between A_2 and A_3 , the extended example treats the arguments A_8 and A_9 as unproblematic. They only result in self-defeating arguments after we build the arguments A_{10} with A_8 as its sub-argument and A_{11} with A_9 as its sub-argument. However, there is a more fundamental problem with the propositions

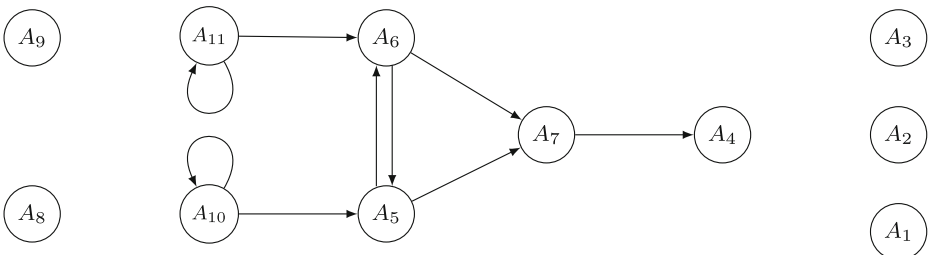


Fig. 8 Abstract argumentation framework corresponding to the argumentation system AS_2

“John says that John is unreliable” and “Mary says that Mary is unreliable”, that is, there is already a problem with the arguments A_8 and A_9 . Justification logic reveals that these two arguments belong to the class of “intensional paradoxes” [58].

Intensional paradoxes are paradoxes of self-reference that “involve intensional notions such as *knowing that*, *saying that*, etc.” [58, p. 193]. A classic example is given by Prior [59, p. 17] in an intensional version of the Liar paradox: “It is said by a Cretan that whatever is said by a Cretan is not the case”. To formalize this paradox, Prior uses propositional quantification [68, § 3.5]. Looking at our example sentences, we can notice one prominent divergence from the intensional Liar paradox, namely that John claims that *John is not reliable*, rather than that *whatever is said by John is not the case*.¹⁰ Thus, it is not the case that John necessarily denies what John is saying, but he is rather denying that what he is saying could be a reason to accept what he is saying. Given our interpretation of reason terms as sources of information, justification logic is able to translate this type of self-referentiality of sources induced by John’s and Mary’s statements. However, neither is the \mathbf{JT}_{CS} language nor any other standard justification logic language expressive enough to translate those statements into justification logic intensional paradoxes. What is missing is the mentioned mechanism of propositional quantification. In an extended language with the possibility to quantify over propositions, we could formalize A_8 and A_9 as follows:

$$j : \forall p(\neg j : p) \text{ and } m : \forall q(\neg m : q),$$

where p and q are propositional variables. This formalization confronts us with the question of whether there is any proposition that the reason j (or the reason m) justify or not. One possible way to address the problem of intensional paradoxes is to allow true contradictions in some contexts, e.g., allowing that the intensional variant of the Liar paradox is both true and false [58, p. 198].

Whatever the solution is to the presented problem, A_8 and A_9 are not ordinary arguments. The argument “John says that John is unreliable” opens up an interesting topic of argumentative paradoxes that has not been explored up to now. Such arguments are rightly deemed paradoxical arguments due to the self-referential character of the reason terms j and m that are meant to justify that they cannot justify a proposition. With the help of justification logic, we can conclude that there are problems for the grounded extension of the argumentation system AS_2 beyond the omission of the attack between the arguments A_2 and A_3 . The grounded extension of AS_2 also concedes two paradoxical arguments, namely A_8 and A_9 . According to our analysis, it is already at the level of A_8 and A_9 that each of these arguments defeats itself. What accounts for this difference is that, in AS_2 , self-defeat occurs as a result of the undercut formalized with the arguments A_{10} and A_{11} . However, justification logic shows that the self-defeat here is not targeting default inferences, but rather the sources j and m and their reliability. This means that the extension presented in Example 30 should be understood as *self-undermining*, rather than *self-undercutting*.¹¹

¹⁰Of course, “John” individuates a single entity, whereas “(a) Cretan” individuates a class of entities.

¹¹Again, as in the case of the arguments A_5 and A_6 and their sub-arguments A_2 and A_3 , [72] and [19] formalize A_{10} as an argument based on A_8 and A_{11} as an argument based on A_9 . But the conclusions of the arguments A_{10} and A_{11} do not go beyond what is specified in the testimonies formalized with A_8 and A_9 . The arguments A_{10} and A_{11} are thus superfluous, according to our non-inferential interpretation of information acquisition. Notice that, even if there were genuine examples of self-undercutting attacks that would defeat two arguments in rebuttal, as depicted in Fig. 8, the problem of trivialization could not be caused by the *Ex falso* principle in default justification logic for the same reasons given above.

If we would follow the proposed extension of the basic justification logic language with propositional quantifiers, there is an interesting connection with the cycles of attacks in Dung’s abstract argumentation frameworks. According to our discussion above, the acceptability status of the following self-defeating argument cannot be easily resolved:

$$j : \forall p(\neg j : p).$$

If we consider a cycle of two sources denying each other’s reliability, e.g.,

$$j : \forall p(\neg k : p) \text{ and } k : \forall q(\neg j : q),$$

there seems to be a possible resolution. Namely, if $j : \forall p(\neg k : p)$ is added to an evidence base $In(\Pi')$ and $k : \forall q(\neg j : q)$ to $In(\Pi'')$ such that $In(\Pi')$ and $In(\Pi'')$ are inconsistent, the arguments might still be accepted in, e.g., different preferred extensions. However, a three-member cycle of attacks, such as

$$j : \forall p(\neg k : p), k : \forall q(\neg l : q) \text{ and } l : \forall r(\neg j : r),$$

is again difficult to resolve and it is difficult to interpret the value of statements justified by the reason terms j , k and l . Following the same principle of adding a new member, a cycle of four reason terms denying one another’s reliability is resolvable in terms of determining the acceptability status of reasons. This pattern is familiar from the the study of the semantic properties of odd- and even-length cycles, especially in the context of determining preferred extensions [14]. Bench-Capon [15] argues that semantic differences in odd- and even-length cycles are justified because odd-length cycles can be seen as representing paradoxes and even-length cycles can be seen as representing dilemmas. Our default justification logic reconstruction of undermining cycles corroborates the intuition behind Bench-Capon’s analysis.

Is it a downside of our logic based on the \mathbf{JT}_{CS} language that we are not able to represent the arguments A_8 and A_9 ? There are reasons not to see this as a downside. The precision of the \mathbf{JT}_{CS} language and its ability to represent arguments as the object-level formulas gives a detailed insight into the nature of attacks between arguments and logical properties of defeasible reasons, but it also shows that there might be some limitations to the logical representation of defeasible arguments. These limitations have not been noticed in the context of structured argumentation frameworks, because such frameworks do not specify a single logical system and they do not represent reasons as formal objects.

4.3 The \mathbf{JT}_{CS} non-interference postulate

In the last part of this section, we focus on showing that our logic satisfies the so-called “non-interference” postulate, first proposed in [20] as a criteria for avoiding the trivialization problem. The idea behind the postulate is that those defeasible theories that allow us to construct arguments for any conclusion from contradictory premises would influence “the entailment of a completely unrelated (syntactically disjoint) defeasible theory when being merged to it” [19, p. 2728].

Before stating the postulate for defeasible theories, we give some preliminary definitions based on [19]. We say that the theories DT_1 and DT_2 are syntactically disjoint iff $Atoms(DT_1) \cap Atoms(DT_2) = \emptyset$, where $Atoms(DT)$ is the set of all atomic propositional formulas occurring in DT . The union of defeasible theories $DT_1 = (P_1, RD_1, n_1)$ and $DT_2 = (P_2, RD_2, n_2)$ is defined as the defeasible theory $DT_{1 \cup 2} = (P_1 \cup P_2, R_1 \cup R_2, n_1 \cup n_2)$. For an argumentation semantics σ , let $Cn_\sigma(DT)$ be the set of consequences defined as $\{Concs(E_1), \dots, Concs(E_n)\}$, where E_1, \dots, E_n are the σ extensions

of the argumentation system AS based on DT and $Concs(E_i)$ is the set of conclusions of all the arguments contained in E_i . Moreover, for a set of propositions S and the set of atomic propositional formulas A , we take $S|_A$ to be defined as $\{\varphi \in S \mid \text{each atomic propositional formula in } \varphi \text{ is in } A\}$. Finally, for a set \mathbf{S} defined as a set of sets of propositions $\{S_1, \dots, S_n\}$, take $\mathbf{S}|_A$ to be defined as $\{S_1|_A, \dots, S_n|_A\}$.

Non-interference For syntactically disjoint defeasible theories $DT_1 = (P_1, RD_1, n_1)$ and $DT_2 = (P_2, RD_2, n_2)$ such that P_1 and P_2 are consistent and an argumentation semantics σ , it holds that

$$Cn_\sigma(DT_1)|_{Atoms(DT_1)} = Cn_\sigma(DT_{1 \cup 2})|_{Atoms(DT_1)}.$$

Thus, the conclusions that are acceptable under some argumentation semantics for the defeasible theory DT_1 should remain acceptable under the same argumentation semantics after merging DT_1 with any other syntactically disjoint defeasible theory.

To define the \mathbf{JT}_{CS} variant of the postulate, we will use the application-restricted generalization of default theories from Definition 18. Recall that each default theory $T = (W, D)$ can be defined as an application-restricted default theory $[\top]T = (W, D)$ with a tautological restriction. We first take $Atoms([F]T)$ to be the set of all atomic propositional formulas occurring in an application-restricted default theory $[F]T = (W, D)$ and we take $Subterms([F]T)$ to be the set of all subterms occurring in $[F]T$. We say that the default theories $[F]T_1$ and $[G]T_2$ are syntactically disjoint iff $Atoms([F]T_1) \cap Atoms([G]T_2) = \emptyset$ and $Subterms([F]T_1) \cap Subterms([G]T_2) = \emptyset$. The union of application-restricted default theories $[F]T_1 = (W_1, D_1)$ and $[G]T_2 = (W_2, D_2)$ is defined as the application-restricted default theory $[F \wedge G]T_{1 \cup 2} = (W_1 \cup W_2, D_1 \cup D_2)$. For a \mathbf{JT}_{CS} semantics σ , let $Cn_\sigma([F]T)$ be the set of consequences defined as $\{\Gamma_1, \dots, \Gamma_n\}$, where $\Gamma_1, \dots, \Gamma_n$ are the \mathbf{JT}_{CS} - σ extensions of $[F]T$. For a set of \mathbf{JT}_{CS} formulas Γ and the set of atomic propositional formulas A , we take $\Gamma|_A$ to be defined as $\{F \in \Gamma \mid \text{each atomic propositional formula in } F \text{ is in } A\}$. Finally, for a set of sets of \mathbf{JT}_{CS} formulas $\mathfrak{S} = \{\Gamma_1, \dots, \Gamma_n\}$, take $\mathfrak{S}|_A$ to be defined as $\{\Gamma_1|_A, \dots, \Gamma_n|_A\}$. The following is the \mathbf{JT}_{CS} variant of the non-interference postulate:

\mathbf{JT}_{CS} non-interference For syntactically disjoint application-restricted default theories $[F]T_1 = (W_1, D_1)$ and $[G]T_2 = (W_2, D_2)$ such that W_1 and W_2 are \mathbf{JT}_{CS} -consistent and a \mathbf{JT}_{CS} - σ semantics, it holds that

$$Cn_\sigma([F]T_1)|_{Atoms([F]T_1)} = Cn_\sigma([F \wedge G]T_{1 \cup 2})|_{Atoms([F]T_1)}.$$

Default justification logic with rebutting, undercutting and undermining attacks can be shown to satisfy the \mathbf{JT}_{CS} non-interference postulate for all \mathbf{JT}_{CS} complete-based semantics. For \mathbf{JT}_{CS} -stable semantics, non-interference cannot be guaranteed for joined (applicability-constrained) default theories. The reason is that the presence of self-undercutting arguments in some default theory may eliminate the definability of the \mathbf{JT}_{CS} -stable extensions, after the two theories have been merged. The following proposition states the result:

Proposition 31 For any two syntactically disjoint application-restricted default theories $[F]T_1 = (W_1, D_1)$ and $[G]T_2 = (W_2, D_2)$ such that W_1 and W_2 are \mathbf{JT}_{CS} -consistent and any \mathbf{JT}_{CS} semantics σ , where

$$\sigma \in \{\mathbf{JT}_{CS}\text{-complete, } \mathbf{JT}_{CS}\text{-grounded, } \mathbf{JT}_{CS}\text{-preferred}\},$$

it holds that

$$Cn_\sigma([F]T_1)|_{Atoms([F]T_1)} = Cn_\sigma([F \wedge G]T_{1 \cup 2})|_{Atoms([F]T_1)}.$$

Proof In this proof, we use the notion of syntactically disjoint \mathbf{JT}_{CS} formulas. We say that F_1 and F_2 are syntactically disjoint iff $Atoms(F_1) \cap Atoms(F_2) = \emptyset$. We first show that adding the premises, default rules and the restriction of $[_G]T_2 = (W_2, D_2)$ cannot change the applicability status of default rules from $[_F]T_1 = (W_1, D_1)$. For some process Π of $[_{F \wedge G}]T_{1 \cup 2}$ containing $\delta^* \in D_2$, consider a segment $\Pi[k - 1]$ of Π to which δ^* is applicable. Consider also some default $\delta' \in D_1$. If δ' is applicable to $\Pi[k - 1]$, assume that applying δ^* blocks the application of δ' to $\Pi[k]$. According to the restricted applicability of defaults, this means that the formula $\neg req(\delta')$ is in $_{F \wedge G}In(\Pi[k])$, but not in $_{F \wedge G}In(\Pi[k - 1])$. Since $_{F \wedge G}In(\Pi[k - 1])$ is \mathbf{JT}_{CS} -consistent, either $cons(\delta^*)$ is not syntactically disjoint with $req(\delta')$ or there is some \mathbf{JT}_{CS} formula in $_{F \wedge G}In(\Pi[k - 1])$ that syntactically connects the atoms of $cons(\delta^*)$ and $req(\delta')$, which is not an instance of \mathbf{JT}_{CS} axioms. Contradiction.

If δ' is not applicable to $\Pi[k - 1]$, assume that applying δ^* activates the application of δ' to $\Pi[k]$. According to the restricted applicability of defaults, this means that the formula $pre(\delta')$ is not in $_{F \wedge G}In(\Pi[k])$, but it is in $_{F \wedge G}In(\Pi[k - 1])$. Again, since $_{F \wedge G}In(\Pi[k - 1])$ is \mathbf{JT}_{CS} -consistent, either $cons(\delta^*)$ is not syntactically disjoint with $pre(\delta')$ or there is some \mathbf{JT}_{CS} formula in $_{F \wedge G}In(\Pi[k - 1])$ that syntactically connects the atoms of $cons(\delta^*)$ and $pre(\delta')$, which is not an instance of \mathbf{JT}_{CS} axioms. Contradiction. Since we also know that the formulas from the \mathbf{JT}_{CS} -consistent sets W_1 and W_2 are syntactically disjoint and that the formulas F and G are syntactically disjoint, the applicability status of default rules from $[_F]T_1 = (W_1, D_1)$ cannot be changed by premises, default rules and the restriction of $[_G]T_2 = (W_2, D_2)$. Thus the set of potential extensions of $[_F]T_1 = (W_1, D_1)$ is identical to the set of potential extensions of $[_{F \wedge G}]T_{1 \cup 2}$, modulo formulas from W_2 and $cons(D_2)$.

If this holds then, in the union $[_{F \wedge G}]T_{1 \cup 2}$, there are no formulas from $cons(D_2)$ that are \mathbf{JT}_{CS} -inconsistent with any of the formulas from W_1 or $cons(D_1)$. Moreover, if $[_F]T_1$ and $[_G]T_2$ are syntactically disjoint, then $\mathcal{WS}^{[_F]T_1} \cap \mathcal{WS}^{[_G]T_2} = \emptyset$. This implies that there are no new attacks on the formulas from $cons(D_1)$ and that exactly all the \mathbf{JT}_{CS} -admissible extensions of $[_F]T_1$ are preserved in $[_{F \wedge G}]T_{1 \cup 2}$. This means that the formulas contained in \mathbf{JT}_{CS} -complete, \mathbf{JT}_{CS} -grounded and \mathbf{JT}_{CS} -preferred extensions of $[_F]T_1$ (or $Cn_\sigma([_F]T_1)_{Atoms([_F]T_1)}$) are identical to the formulas contained in \mathbf{JT}_{CS} -complete, \mathbf{JT}_{CS} -grounded and \mathbf{JT}_{CS} -preferred extensions of the union $[_{F \wedge G}]T_{1 \cup 2}$, relative to the atoms of $[_F]T_1$ (or $Cn_\sigma([_F \wedge G]T_{1 \cup 2})_{Atoms([_F]T_1)}$). \square

To conclude this section, we also note that the \mathbf{JT}_{CS} variants of indirect (and thus direct) consistency are guaranteed if \mathbf{JT}_{CS} consistency of the set of premises W is assumed (consistency of premises is also assumed when proving the postulate in, e.g., ASPIC+ [19, p. 2717]). The proof is straightforward, given the definitions of (restricted) applicability, \mathbf{JT}_{CS} -extensions and the assumption of consistent premises.

5 Related and future work

5.1 Related work

The study of basic default justification logic began with [50]. The logic is further developed and described in [53] and [54]. This paper extends [52] presented at the 11th International Symposium on Foundations of Information and Knowledge Systems (FoIKS2020). The new formal result reported here is the satisfaction of the non-interference postulate. In addition to non-interference, the paper proposes default justification logic as a toolkit to study a relatively unexplored domain of paradoxes in formal argumentation.

This paper contributes to the study of non-prioritized belief revision operations, that is, such operations for which the new information has no special priority due to its novelty [37, 38]. The way in which our operators are defined meaningfully combines resources from both belief revision and default logic. The relation between belief revision and non-monotonic reasoning has long been a matter of discussion [36, 43] among AI researchers. Gärdenfors [36] sees them as the two sides of the same coin. His views are based on the possibilities to translate belief revision postulates into different non-monotonic inference variants. Makinson [43] also follows the direction of translating between belief-revision postulates and conditions on non-monotonic inference conditions. Although it was not our primary aim to discuss the relation between modeling reasoning with *incomplete information* in default theories and modeling reasoning with *changing information* in belief revision, we showed that our justification logic creates a useful junction for the two approaches and provides a fresh outlook on their relation.

As mentioned in the introduction to this paper, our approach to structured argumentation dynamics builds on a similar strategy as Antoniou's [7] approach to the dynamics of standard default theories. Antoniou's approach significantly differs from ours in the way he treats those changes that add or remove a formula at the level of some, but not necessarily all extensions. He first defines [7, pp. 1146-1147] an operation that secures the existence of at least one extension that contains a formula and then [7, p. 1149] another operation that removes a formula from at least one extensions of a Reiter's default theory. Unlike our conservative contraction operation, none of Antoniou's contraction operations leaves the inclusion status of a formula undecided. In fact, to secure that a formula is not contained in at least one extension, Antoniou adds an extension where introducing the formula in question is blocked. Similarly, he does not consider an expansion operator that does not ensure in advance that a formula is contained in an extension, as done in our conservative expansion.

We will consider two problems with the operations from [7] that fix the status of a formula by either adding a formula in at least one extension by revision or removing a formula from at least one extensions by contraction. Both of them are technical in nature and they raise issues about the economy of introducing such operations. Firstly, the operations are based on adding a new atom to the language so as to prevent applying defaults with that atom as one of their "justifications" (our *consistency requirement* formulas) to a process together with those rules that have the negation of that atom as one of their justifications. This ensures manipulating the content of extensions, but it is rather involved and the new atom is introduced as an *ad hoc* solution to keep the formulas from unwanted interaction.

The second problem is also related to the method of adding new extensions for which a formula needs to be included in (or excluded from). As we mentioned, [7] relies on introducing a new atom and its denial to induce inconsistency. This ensures that there are extensions that do (by revision) or do not contain (by contraction) the target formula. As a side effect, the definitions of those operations make it impossible that *any* old default rule of the revised or contracted theory becomes applicable together with the rule which is added for the revising or contracting purposes. A downside of such operations is that they eliminate meaningful interaction between such defaults.

Some approaches to default reasoning such as [42] and [45] represent the idea of defaults in dynamic epistemic logic. The main focus of van Linder, van der Hoek and Meyer [42] is to embed supernormal defaults in a multi-agent modal logic with knowledge, belief and update modalities. The authors show that Reiter's extensions can be represented as a result of consecutive jump actions to default conclusions, but they do not focus on how such extensions are revised due to information changes. Meyer and van der Hoek [45] introduce a preference modality to distinguish between known and (provisionally) preferred

information. A non-monotonic belief revision component consists in changing preferences as a result of obtaining knowledge.

Baltag, Renne and Smets [12, 13] and Renne [61] define extensions of justification logic in which agents may acquire new information that defeats the reasons they accepted. The logics combine belief revision and dynamic epistemic logic techniques to model a kind of defeat that seems to correspond to undermining. However, each of the logics assumes prioritizing new information and none of these logics is able to model undercut and rebuttal. Even so, approaches based on dynamic epistemic logic are attractive because they open up a possibility of developing a multi-agent justification logic with defeaters.

We indicated in Section 1 to this paper that the work in the area of the dynamics of argumentation frameworks without argument structures is already well-developed. Among the approaches, it is worth mentioning those that follow the belief revision methods applied to Dung's argumentation frameworks, such as that of [17] and [24]. Booth et al. [17] start from a labelling approach to Dung's argumentation frameworks and constraints on a framework's output. Their focus is on finding the best way to recover a rational output given a framework and a constraint on its output. For this, they use ordering of conflict-free labellings in a way that the most rational conflict-free labelling is chosen when none of complete labellings respects the constraint. In the work by Diller et al. [24], we find two kinds of revision operators. One of them revises an abstract argumentation framework by taking a propositional formula as a means to represent the new information, while the other operation revises an input framework by information in the form of another framework. Both operations give a single output framework respecting a particular type of rankings on extensions.

Although Section 4 establishes that our logic behaves rationally with respect to the non-interference postulate, there is still a need for a thorough investigation of how our logic relates to other rationality postulates that are discussed by, e.g., Caminada [19]. This is especially interesting now that our logic models all the three standard types of argumentative attacks. Up to now, this has been possible only by using argumentation frameworks such as ASPIC+. In Section 4, we indicated that satisfying consistency postulate has a straightforward proof. In [54, § 5], other desirable properties of extensions (such as, e.g., closure, free-precedence etc.) are discussed in the context of default justification logic without undermining. The addition of undermining can be assumed not to affect the satisfaction of those properties.

It is still an open question how structured argumentation frameworks in general exactly relate to default justification logic. Similarities between the two consist in the examples that these two types of systems aim to express and in concepts that the two systems share. However, major differences in methodology and logical vocabularies prevent us from making bold claims about the connections between structured argumentation frameworks (including, but not limited to, ABA [66], ASPIC+ [48], DefLog [69], DeLP [35] and deductive argumentation [16]) and default justification logic. One point on which we diverge from all argumentation frameworks is that we represent arguments as object-level formulas of the \mathbf{JT}_{CS} language. Thus, when we define "argument acceptance" and \mathbf{JT}_{CS} extensions, we are in fact bringing about different ways to define a logical consequence relation. This means that our system can justly be considered as a *logic of arguments*, rather than a *framework for arguments*. From the technical perspective, what underlies the possibility to model arguments as object-level formulas is that justification terms label propositional formulas. This, in turn, enables the rendering of exclusionary reasons by means of undercut from Definition 10.

It is known that ASPIC+ [56] also explicitly models undercutting, rebutting and undermining defeat. We will illustrate how the above discussed difference between *framework* and *logic* reflects in the way in which ASPIC+ models undercutters. In ASPIC+, undercutting attack is understood as an attack on a defeasible inference step of an argument. But to express this relation in the system, we need a function for “lifting” names of defeasible rules as corresponding formulas of the base language. On our approach, warrants have the role of default rules in the system. But recall that, on our formalization of warrants, they already are justification logic formulas that underlie default rules. The relation between ASPIC+ and default justification logic could be appropriately described as a relation between a meta-theory of argumentation and a specific theory of argumentation.

One conceptual difference between ASPIC+ and our logic concerns the ways in which the two systems model undermining attacks. That is, ASPIC+ seems to adopt the idea of what we called “inferential” undermining attacks in the sense that an argument can be attacked on its defeasible premise if there is some argument whose conclusion is, for example, the denial of a defeasible premise of the first argument. At first glance, this notion of undermining is eliminated by our definition that specifies formulas from W as target formulas for undermining. However, the idea of inferential undermining attacks is compatible with the way we represent argument premises and our definition could in principle be adapted to include this undermining attack variation. For example, consider some justification assertion $t : F$ introduced as a consequent of a supernormal default. Such formula can plausibly be regarded as a defeasible (or ordinary) premise within a default theory, under the interpretation that the lower level of information certainty indicates the level of reliability for t . Now consider that there is some warranted default introducing $(u \cdot v) : \neg F$ as its consequent. By relaxing Definition 25, such cases would account for the inferential reading of undermining.

An interesting way to connect our logic to structured argumentation frameworks is to study how it relates to Defeasible Logic Programming [34, 35] (or DeLP, for short). The logic presented here is based on the ideas from default logic and the relationship between logic programming and default logic was already discovered in [44] through the stable model semantics. In DeLP, we find a different interpretation of Toulmin’s warrants. DeLP warrants are used to determine a “status of a claim” by resorting to a “set of dialectical trees that was generated to decide the warrant status of the arguments related to the claim” [35, p. 64]. In the context of the current work, it is interesting that DeLP has defeasible rules with an empty body that are interpreted as *presumptions*. Such rules seem to formally correspond to supernormal defaults used in this paper. Connections between DeLP and default justification logic are especially worth exploring given the new developments of Extended Defeasible Logic Programming [22] with the added notions of backing and undercutting.

5.2 Some limitations and future work

In this paper, we define *local* change operations that do not, in general, give a recipe on how to further change their output default theories. But to represent actual dynamic contexts of argumentation, we need to make our operators *global*, rather than local, and enable iterated revision. Note that, for example, the second output theory $(\llbracket F \rrbracket T_{-F}^-)^\times$ of Definition 22 is an application-restricted default theory. If we want to allow for iterated contraction and generalize the contraction operation to application-restricted theories such as $(\llbracket F \rrbracket T_{-F}^-)^\times$, we need to deal with multiple restrictions to the output theory, and possibly mutually inconsistent restrictions. This could be done if we allow that an application-restricted theory $\llbracket F \rrbracket T$ can be further restricted by a formula G in such a way that, for any default δ , we need to check if

$req(\delta)$ is consistent with both $FIn(\Pi)$ and $GIn(\Pi)$, thus defining the application-restricted default theory $[G, F]T$.

An important issue that we do not address here is that of computational costs related to the operational semantics with argumentation formulas, and in particular related to the problem of determining whether an argument is contained in an extension. Even restricted versions of propositional default logic do not fare well in terms of tractability [65] compared to classical propositional logic. Despite the fact that all our default theories are normal, the problem of, e.g., determining the existence of extensions is aggravated by the added grammar of reason terms. Consider that, in contrast to propositional default logic, we need to look at all the closed default processes to establish that, for example, a \mathbf{JT}_{CS} -stable extension is defined for some default theory. Moreover, asking whether a formula belongs to a \mathbf{JT}_{CS} extension turns into a problem for which we also need to keep track of all the branches of a process tree, which resembles the problems of determining skeptical or credulous entailment for a formula in propositional default logic.

It can also be noticed that our defaults are not given as default schemes, as it is the case with standard default logics. The fact that the only type of rules are exclusively specific defaults may be seen as a limitation. To see why, consider that we sometimes want to talk about reasons for why warrants hold as being of a general type, based, for example, on some observed regularity. This is one step that is missing for a full formalization of Toulmin’s argumentative schema [54, p. 16]. Recent developments of the first-order variant of the logic of proofs [33] might bring us closer to this goal. By using first-order logic of proofs, we will enable a formalization of generalized reasons for Toulmin’s warrants in the following format of default rule schemes:

$$\frac{t_{\{x\}} : F :: (u \cdot t)_{\{x\}} : G}{(u \cdot t)_{\{x\}} : G},$$

where x in $t_{\{x\}} : F$ is a free variable throughout the derivation t .

There are other paths for future research connected to quantification. As discussed in Section 4, semantic modeling of paradoxes in argumentation would require a system with quantification over propositional variables. In such a system, we would be able to precisely describe the relation between resolving the paradoxes in formal argumentation and resolving odd- and even-length cycles of attacks in Dung’s abstract argumentation frameworks (see Section 4). Note also that the idea of quantifying over justification terms has already been explored by the justification logic community [10, § 8].

6 Conclusions

In this paper, we took a step beyond the default reasoning paradigm and we investigated an extension of default justification logic with plausible reasoning patterns. In plausible reasoning, an agent needs to deal with uncertainty of premises and inconsistent premises. In contrast, the basic default justification logic assumed certainty of premises and uncertainty of inferences. The purpose of considering plausible reasoning was to provide a unified logical system for all three standard types of argumentative attack, namely undercutting, rebutting and undermining. Justification assertions turned out to be an appropriate logical format for representing undermining attacks. In the plausible-reasoning extension of our logic, an assertion $t : F$ from a set of premises W is interpreted in such a way that t represents a source that supports the formula F . On our account, undermining is then interpreted

as a removal of the unreliable source of information t . What we needed is to define operations that are able to alter even the set of premises W . To this end, we used techniques from belief revision to model information dynamics of justification logic default theories.

The assumption behind this paper is that default reasoning deals with inconsistencies that arise from extending the starting premises, while belief revision deals with inconsistencies that result from receiving information incompatible with the starting premises. When this assumption is applied to justification assertions, it becomes clear that belief revision has more to do with the plausible reasoning paradigm than with default reasoning paradigm. Once again, justification logic acted as an intermediary system, this time by throwing light on the relation between default and plausible reasoning that was translated into relations between default logic and belief-revision techniques. With their fine-grained representation of reasons, justification assertions gave us a better perspective on the ways in which incoming information causes different types of non-monotonic behavior in belief revision and default logic. To the best of our knowledge, the system presented here is the first *logic* to combine plausible and default reasoning to capture all the standard notions of defeat in AI.

Acknowledgements I am grateful to Allard Tamminga, Barteld Kooi and Rineke Verbrugge for their generous advice and valuable comments on the previous versions of this manuscript. A part of the research reported here was carried out with the support of the Ammodo KNAW project *Rational Dynamics and Reasoning* awarded to Barteld Kooi. My current work is supported by the NWO project *Empowering Human Intentions through Artificial Intelligence* led by Jan Broersen. Finally, I would like to thank the audiences at “The Symposium on Logical Foundations of Computer Science” and “The 11th International Symposium on Foundations of Information and Knowledge Systems”, as well as the reviewers of this journal for many constructive suggestions that helped me to improve the manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. *J. Symbolic Logic* **50**(2), 510–530 (1985)
2. Alfano, G., Cohen, A., Gottifredi, S., Greco, S., Parisi, F., Simari, G.R.: Dynamics in Abstract Argumentation Frameworks with Recursive Attack and Support Relations. In: Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) *ECAI 2020 - 24th European Conference on Artificial Intelligence*, *Frontiers in Artificial Intelligence and Applications*, vol. 325, pp. 577–584. IOS Press (2020). <https://doi.org/10.3233/FAIA200141>
3. Alfano, G., Greco, S., Parisi, F.: An efficient algorithm for skeptical preferred acceptance in dynamic argumentation frameworks. In: Kraus, S. (ed.) *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pp. 18–24 (2019). <https://doi.org/10.24963/ijcai.2019/3>
4. Alfano, G., Greco, S., Parisi, F., Simari, G.I., Simari, G.R.: An incremental approach to structured argumentation over dynamic knowledge bases. In: Thielscher, M., Toni, F., Wolter, F. (eds.) *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning, KR 2018*, pp. 78–87 (2018)
5. Alfano, G., Greco, S., Parisi, F., Simari, G.I., Simari, G.R.: Incremental computation of warranted arguments in dynamic defeasible argumentation: the rule addition case. In: Haddad, H.M., Wainwright, R.L., Chbeir, R. (eds.) *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018*, pp. 911–917. ACM (2018). <https://doi.org/10.1145/3167132.3167232>

6. Antoniou, G.: Nonmonotonic reasoning. MIT Press, Cambridge (1997)
7. Antoniou, G.: On the dynamics of default reasoning. *Int. J. Intell. Syst.* **17**(12), 1143–1155 (2002)
8. Artemov, S.N.: Operational modal logic. Tech. Rep. MSI 95–29, Mathematical Sciences Institute Cornell University (1995)
9. Artemov, S.N.: Explicit provability and constructive semantics. *Bull. Symbolic Logic* 1–36 (2001)
10. Artemov, S.N., Fitting, M.: Justification Logic. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*, Winter 2016 Edn. Metaphysics Research Lab, Stanford University (2016)
11. Artemov, S.N., Fitting, M.: Justification Logic: Reasoning with Reasons, Cambridge tracts in mathematics, vol. 216. Cambridge University Press, Cambridge (2019)
12. Baltag, A., Renne, B., Smets, S.: The logic of justified belief change, soft evidence and defeasible knowledge. In: Ong, L., de Queiroz, R. (eds.) *International Workshop on Logic, Language, Information, and Computation*, pp. 168–190. Springer (2012)
13. Baltag, A., Renne, B., Smets, S.: The logic of justified belief, explicit knowledge, and conclusive evidence. *Annals Pure Appl. Logic* **165**(1), 49–81 (2014)
14. Baroni, P., Giacomini, M.: Solving semantic problems with odd-length cycles in argumentation. In: Dyhre Nielsen, T., Lianwen Zhang N. (eds.) *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Vol. 2711 of LNCS, pp. 440–451. Springer (2003)
15. Bench-Capon, T.J.M.: Dilemmas and paradoxes: cycles in argumentation frameworks. *J. Log. Comput.* **26**(4), 1055–1064 (2014)
16. Besnard, P., Hunter, A.: Constructing argument graphs with deductive arguments: a tutorial. *Argument Computat.* **5**(1), 5–30 (2014)
17. Booth, R., Kaci, S., Rienstra, T., van der Torre, L.: A logical theory about dynamics in abstract argumentation. In: Liu, W., Subrahmanian, V.S., Wijsen, J. (eds.) *International Conference on Scalable Uncertainty Management*, SUM 2013, vol. 8078 of LNCS, pp. 148–161. Springer (2013)
18. Brezhnev, V.: On the logic of proofs. In: Striegnitz, K. (ed.) *Proceedings of the Sixth ESSLLI Student Session*, Helsinki, pp. 35–46 (2001)
19. Caminada, M.W.A.: Rationality postulates: Applying argumentation theory for non-monotonic reasoning. *J. Appl. Logics* **4**(8), 2707–2734 (2017)
20. Caminada, M.W.A., Carnielli, W.A., Dunne, P.E.: Semi-stable semantics. *J. Log. Comput.* **22**(5), 1207–1254 (2012)
21. Chisholm, R.M.: *Theory of knowledge*. Englewood cliffs NJ: Prentice-hall (1966)
22. Cohen, A., García, A.J., Simari, G.R.: A structured argumentation system with backing and undercutting. *Eng. Appl. Artif. Intell.* **49**, 149–166 (2016)
23. Coste-Marquis, S., Konieczny, S., Maily, J.G., Marquis, P.: On the revision of argumentation systems: Minimal change of arguments statuses. In: Baral, C., De Giacomo, G., Eiter, T. (eds.) *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, KR 2014 (2014)
24. Diller, M., Haret, A., Linsbichler, T., Rümmele, S., Woltran, S.: An extension-based approach to belief revision in abstract argumentation. In: Yang, Q., Wooldridge, M. (eds.) *Twenty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI 2015 (2015)
25. Doutre, S., Herzig, A., Perrussel, L.: A dynamic logic framework for abstract argumentation. In: Baral, C., De Giacomo, G., Eiter, T. (eds.) *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, KR 2014 (2014)
26. Doutre, S., Herzig, A., Perrussel, L.: Abstract argumentation in dynamic logic: Representation, reasoning and change. In: Liao, B., Ågotnes, T., Wáng, Y.N. (eds.) *Dynamics, Uncertainty and Reasoning*, The Second Chinese Conference on Logic and Argumentation, CLAR 2018, pp. 153–185. Springer (2018). https://doi.org/10.1007/978-981-13-7791-4_8
27. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* **77**(2), 321–357 (1995)
28. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-Based Argumentation. In: Rahwan, I., Simari, G.R. (eds.) *Argumentation in artificial intelligence*, pp. 199–218. Springer (2009)
29. van Eemeren, F.H., Garssen, B., Krabbe, E.C.W., Henkemans, A.F.S., Verheij, H.B., Wagemans, J.H.M.: Argumentation and artificial intelligence. In: *Handbook of Argumentation Theory*, pp. 615–675. Springer (2014)
30. Fitting, M., Běhounek, L., Bílková, M. (eds.): *A Logic of Explicit Knowledge*. Prague, Filosofia (2005)
31. Fitting, M.: The logic of proofs, semantically. *Annals Pure Appl. Logic* **132**(1), 1–25 (2005)
32. Fitting, M.: Justification logics, logics of knowledge, and conservativity. *Ann. Math. Artif. Intell.* **53**(1–4), 153–167 (2008)
33. Fitting, M.: Possible world semantics for first-order logic of proofs. *Annals Pure Appl. Logic* **165**(1), 225–240 (2014)

34. García, A.J., Simari, G.R.: Defeasible logic programming: an argumentative approach. *Theor. Pract. Logic Programm.* **4**(1+2), 95–138 (2004)
35. García, A.J., Simari, G.R.: Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. *Argument Comput.* **5**(1), 63–88 (2014)
36. Gärdenfors, P.: Belief revision and nonmonotonic logic: Two sides of the same coin? In: Van Eijck, J. (ed.) *European Workshop on Logics in Artificial Intelligence, JELIA 1990*, pp. 52–54. Springer (1990)
37. Hansson, S.O. (ed.): *Special Issue on Non-Prioritized Belief Revision, Theoria*, vol. 63(1-2). Wiley, Chichester (1997)
38. Hansson, S.O.: A survey of non-prioritized belief revision. *Erkenntnis* **50**(2-3), 413–427 (1999)
39. Hansson, S.O.: *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Kluwer, Dordrecht (1999)
40. Hlobil, U.: We cannot infer by accepting testimony. *Philos. Stud.* 1–10. <https://doi.org/10.1007/s11098-018-1142-3> (2018)
41. Kuznets, R., Studer, T.: *Logics of proofs and justifications college publications* (2019)
42. van Linder, B., van der Hoek, W., Meyer, J.J.C.: The dynamics of default reasoning. *Data Knowl. Eng.* **3**(21), 317–346 (1997)
43. Makinson, D., Gärdenfors, P.: Relations between the Logic of Theory Change and Nonmonotonic Logic. In: Fuhrmann, A., Morreau, M. (eds.) *The Logic of Theory Change: Workshop, Konstanz, FRG, October 13-15, 1989, Proceedings*, pp. 183–205. Springer (1991)
44. Marek, W., Truszczyński, M.: Stable semantics for logic programs and default theories, naclp'89. In: *Proceedings of the North American Conference on Logic Programming*, pp. 243–256. MIT Press (1989)
45. Meyer, J.J.C., van der Hoek, W.: Non-monotonic reasoning by monotonic means. In: Van Eijck, J. (ed.) *European Workshop on Logics in Artificial Intelligence, JELIA 1990*, pp. 399–411. Springer (1990)
46. Mkrtychev, A.: Models for the Logic of Proofs. In: Adian, S., Nerode, A. (eds.) *Logical Foundations of Computer Science, 4th International Symposium, LFCS '97*, vol. 1234 of LNCS, pp. 266–275. Springer (1997)
47. Modgil, S., Prakken, H.: Resolutions in structured argumentation. In: Verheij, B.H., Szeider, S., Woltran, S. (eds.) *Computational Models of Argument: Proceedings of COMMA 2012*, pp. 310–321. IOS Press (2012)
48. Modgil, S., Prakken, H.: The ASPIC+ framework for structured argumentation: A tutorial. *Argument Comput.* **5**(1), 31–62 (2014)
49. Niskanen, A., Järvisalo, M.: Strong refinements for hard problems in argumentation dynamics. In: Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) *ECAI 2020 - 24th European Conference on Artificial Intelligence, Frontiers in Artificial Intelligence and Applications*, vol. 325, pp. 841–848. IOS Press (2020). <https://doi.org/10.3233/FAIA200174>
50. Pandžić, S.: A logic of default justifications. In: FermÉ, E., Villata, S. (eds.) *Nonmonotonic Reasoning, 17th International Workshop, NMR 2018*, pp. 126–135 (2018)
51. Pandžić, S.: Reifying default reasons in justification logic. In: Beierle, C., Ragni, M., Stolzenburg, F., Thimm, M. (eds.) *Proceedings of the KI 2019 Workshop on Formal and Cognitive Reasoning, DKB-KIK 2019*, vol. 2445, pp. 59–70. CEUR Workshop Proceedings (2019)
52. Pandžić, S.: On the dynamics of structured argumentation: modeling changes in default justification logic. In: Herzog, A., Kontinen, J. (eds.) *Foundations of Information and Knowledge Systems, 11th International Symposium, FoIKS 2020*, vol. 12012 of LNCS, pp. 222–241. Springer (2020)
53. Pandžić, S.: *Reasoning with Defeasible Reasons*. Ph.D. thesis, University of Groningen (2020)
54. Pandžić, S.: A logic of defeasible argumentation: Constructing arguments in justification logic. *Argu. Comput. Pre-press* 1–45. <https://doi.org/10.3233/AAC-200536> (2021)
55. Pollock, J.L.: Defeasible reasoning. *Cognit. Sci.* **11**(4), 481–518 (1987)
56. Prakken, H.: An abstract framework for argumentation with structured arguments. *Argument Comput.* **1**(2), 93–124 (2010)
57. Prakken, H.: Historical overview of formal argumentation. *IfCoLog J. Logics Appl.* **4**(8), 2183–2262 (2017)
58. Priest, G.: Intensional paradoxes. *Notre Dame Journal of Formal Logic* **32**(2), 193–211 (1991)
59. Prior, A.N.: On a family of paradoxes. *Notre Dame Journal of Formal Logic* **2**(1), 16–32 (1961)
60. Reiter, R.: A logic for default reasoning. *Artif. Intell.* **13**(1-2), 81–132 (1980)
61. Renne, B.: Multi-agent justification logic: Communication and evidence elimination. *Synthese* **185**(1), 43–82 (2012)
62. Rescher, N.: *Plausible Reasoning: an Introduction to the Theory and Practice of Plausibilistic Inference*. Assen, Van Gorcum (1976)

63. Rescher, N.: *Dialectics: A Controversy-Oriented approach to the theory of knowledge*. SUNY Press, Albany (1977)
64. de Saint-Cyr, F.D., Bisquert, P., Cayrol, C., Lagasquie-Schiex, M.C.: Argumentation update in YALLA (yet another logic language for argumentation). *Int. J. Approx. Reason.* **75**, 57–92 (2016)
65. Stillman, J.: The complexity of propositional default logics. In: *AAAI-92: Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 794–799 (1992)
66. Toni, F.: A tutorial on assumption-based argumentation. *Argument Computat.* **5**(1), 89–117 (2014)
67. Toulmin, S.E.: *The Uses of Argument*. Cambridge University Press, Cambridge (1958/2003)
68. Uzquiano, G.: Quantifiers and Quantification. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*, Summer 2020 Edn. Metaphysics Research Lab, Stanford University (2020)
69. Verheij, B.: DefLog: On the logical interpretation of prima facie justified assumptions. *J. Log. Comput.* **13**(3), 319–346 (2003)
70. Verheij, B.: The Toulmin argument model in artificial intelligence. In: Rahwan, I., Simari, G.R. (eds.) *Argumentation in Artificial Intelligence*, pp. 219–238. Springer (2009)
71. Vreeswijk, G.A.W.: *Studies in defeasible argumentation*. Ph.D. thesis Free University of Amsterdam (1993)
72. Wu, Y., Podlaszewski, M.: Implementing crash-resistance and non-interference in logic-based argumentation. *J. Log. Comput.* **25**(2), 303–333 (2015)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.