Reifying Default Reasons in Justification Logic[*]

Stipe Pandžić[0000−0002−6812−1437]

Department of Theoretical Philosophy, Faculty of Philosophy &
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence,
Faculty of Science and Engineering
University of Groningen, Groningen, The Netherlands
s.pandzic@rug.nl

Abstract. The main goal of this paper is to argue that justification logic advances the formal study of default reasons. After introducing a variant of justification logic with default reasons, we first show how the logic can be used to model undercutting attacks and exclusionary reasons. Then we compare this logic to Reiter's default logic interpreted as an argumentation framework. The comparison is done by analyzing differences in the way in which process trees are built for the two logics.

Keywords: Justification logic · Default reasons · Defeaters · Formal argumentation · Reiter's default logic.

1 Introduction

In its most general sense, the notion of "reason" can be understood as a support or justification for a conclusion or an action. In reasoning from incomplete information, reasons are usually not strong enough to eliminate any possibility of conflicting reasons, that is, reasons are often only defeasible. A good example is human reasoning with its dedication to identifying undefeated reasons upon which actions and conclusions are to be based. But what are default reasons formally?

One notable logical account of defeasible reasoning is Reiter's default logic [17, 1] with its rules that enable "jumping" to conclusions. The answer to our question in Reiter's logic is, however, left ambiguous. This problem is identified by Horty in the following passage about the *reification* of reasons [9, p. 6]:

> Suppose, as in our example, that the agent's background theory contains the default $B \rightarrow F$, an instance for Tweety of the general default that birds fly, together with B, the proposition that Tweety is a bird, so that the default is triggered. In this case, it seems plain that the agent has a reason to conclude that Tweety flies. But how, exactly, should this reason be reified? Should it be identified with the default $B \rightarrow F$ itself, or with the proposition B?

Horty's conclusion is that this question "like many questions concerning reification, is somewhat artificial" and that "when it comes to reification, the reason relation could be projected in either direction, toward defaults or propositions, and the choice is largely arbitrary".

The goal of this paper is to show that the question of reification is important and that giving an answer to it opens up new paths in formalizing defeasible reasoning. In particular, our focus is on showing benefits of formalizing default reasons with the language of justification logic, which is expressive enough to encode the structure of default inferences within its reason terms.

The paper has the following structure. The next section introduces the basics of the logic of default justifications first introduced in [12] and developed in [13] as a theory of structured arguments that generalizes (a large subclass of) Dung's frameworks [7]. Section 3 elaborates on the conceptual differences between Reiter's logic interpreted as an argumentation framework and our justification logic argumentation theory. We start by showing how to represent exclusionary reasons and undercutting defeat in process trees for justification logic-based default theories. Then we show how to translate undercut into Reiter's default logic by interpreting its default processes as arguments. We argue that our logic conforms better to the idea of making a default inference without having to anticipate numerous exceptions to the inference.

2 Logic of default reasons

The logic of default justifications was first presented in [12]. Here, we can only give an overview of the key definitions, starting with the basic justification logic language.[1]

2.1 Syntax, axioms and rules of JT$_{\mathbf{CS}}$

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 all justification terms, constructed from variables $x_1, \ldots, x_n, \ldots$ and proof constants $c_1, \ldots, c_n, \ldots$ by means of operations '$\cdot$' and '$+$'. The grammar of justification terms is given by the following Backus-Naur form:
$$t ::= x \mid c \mid (t \cdot t) \mid (t + t)$$
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. Formulas of $\mathbf{JT_{CS}}$ based on the countable set of propositional atoms $\mathcal{P}$ formulas are defined by the following Backus-Naur form:
$$F ::= \top \mid P \mid (F \to F) \mid (F \vee F) \mid (F \wedge F) \mid \neg F \mid t : F$$

[1] Justification logic was first characterized as a logic of arithmetic proofs in [2]. For the basic workings of justification logic operators and the relation between justification logic variants see, e.g., [8].

where $P \in \mathcal{P}$ and $t \in Tm$. The set Fm consists of exactly all formulas.

We can now define the logic $\mathbf{JT_{CS}}$, which is the weakest logic with "truth inducing" justifications containing axiom schemes for the two basic operations *application* '·' and *sum* '+'. Informally, application (axiom A1 below) represents the reification of the *modus ponens* inference as a reason term, while sum (axiom A2 below) enables merging two reason terms to produce a new reason term. These are the axioms and rules 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 $t : F \rightarrow (t + u) : F$; $\qquad u : F \rightarrow (t + u) : F$ (Sum)
A3 $t : F \rightarrow F$ (Factivity)
R0 *From F and $F \rightarrow G$ infer G* (Modus ponens)
R1 *If F is an instance of A0-A3, $c_n, c_{n-1}, \ldots, c_1$ proof constants and $c_{n-1} : \cdots : c_1 : F \in \mathcal{CS}(c_n)$, then infer $c_n : c_{n-1} : \cdots : c_1 : F$* (Iterated axiom necessitation),

where the function $\mathcal{CS}(c)$ assigns formulas to any constant according to the following two conditions:

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

The intuition behind the function $\mathcal{CS}(c)$ in R1 is that the basic logical axioms are taken to be justified by proof constants and so are the formulas likewise produced. A set of instances of rule R1 is called *Constant Specification* ($\mathcal{CS}$) set:

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

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.

2.2 Semantics

The semantics for $\mathbf{JT_{CS}}$ is an adapted version of the semantics for the logic of proofs ($\mathbf{LP}$) given by [11]. The $\mathbf{JT_{CS}}$ logic can be proved to be sound and complete with respect to the semantics below.

Definition 2 ($\mathbf{JT_{CS}}$ model) *A function* reason assignment *based on $\mathcal{CS}$ is defined as* $*(\cdot) : Tm \rightarrow 2^{Fm}$, *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 $v : \mathcal{P} \rightarrow \{True, False\}$ is a function assigning truth values to propositional formulas in $\mathcal{P}$. We define the interpretation $\mathcal{I}$ as a pair $(v, *)$. For an interpretation $\mathcal{I}$, $\models$ is a truth relation on the set of formulas of $\mathbf{JT_{CS}}$. We say that, for any formula $t : F \in Fm$, $\mathcal{I} \models t : F$ 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 $\mathcal{I} \models F$.*

We define the consequence relation of the logic of factive reasons $\mathbf{JT_{CS}}$ on reflexive interpretations in such a way that, for any set of $\mathbf{JT_{CS}}$ formulas Σ:

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 JT_{CS} consequence relation $\models$ defined above, a $\mathbf{JT_{CS}}$ closure of Γ is given by $Th^{JT_{CS}}(\Gamma) = \{F | \Gamma \models F\}$. For a closure $Th^{JT_{CS}}(\Gamma)$, it holds that $\mathcal{CS} \subseteq Th^{JT_{CS}}(\Gamma)$.

2.3 Default theories based on the $\mathbf{JT_{CS}}$ language

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

Definition 4 (Default Theory) *A default theory T is defined as a pair (W, D), where the set 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". The default rule δ introduces a unique reason term u, which means that, for a default theory T, the following holds:

1.) For any formula $v : H \in Th^{JT_{CS}}(W)$, $u \neq v$;

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 $F \neq F'$ or $G \neq G'$, then $u \neq u'$.

Every default rule produces a reason term whose structure codifies an application operation step. Notice that in δ above, in contrast to axiom A1, we do not require the formula $u : (F \rightarrow G)$ 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 $\mathbf{JT_{CS}}$ formulas. They will be referred to as "warrants", because their twofold role mirrors that of Toulmin's argument warrants [19, p. 91]. Warrants extend the interpretation of the application operation "·" and 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: $\#(\delta_i) = u : (F \rightarrow G)$, where $\delta_i \in D$ and $\delta_i = \frac{t:F::(u\cdot t):G}{(u\cdot t):G}$, for some reason term t, a unique reason term u and some formulas F and G.

A set of all such underlying warrants of default rules is called *Warrant Specification ($\mathcal{WS}$) set*.

Definition 5 (Warrant specification) *For a default theory $T = (W, D)$, justified defeasible conditionals are given by the Warrant Specification set:*

$$\mathcal{WS}^T = \#[D] = \{u_n : (F \rightarrow G) \mid \#(\delta_i) = u_n : (F \rightarrow G) \ and \ \delta_i \in D\}.$$

The basis of operational semantics for a default theory $T = (W, D)$ is the procedure of collecting new, reason-based information from the available defaults. A *sequence* of default rules $\Pi = (\delta_0, \delta_1, \ldots)$ 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 in the following way: for a set of **JT$_{\mathbf{CS}}$**-closed formulas Γ we say that a default rule $\delta = \frac{t:F::(u \cdot t):G}{(u \cdot t):G}$ 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 **JT$_{\mathbf{CS}}$** formulas that represents the current evidence base: $In(\Pi) = Th^{JT_{CS}}(W \cup \{cons(\delta) \mid \delta \text{ occurs in } \Pi\})$. The set $In(\Pi)$ collects reason-based information that is yet to be determined as acceptable or unacceptable depending on the acceptability of reasons and counter-reasons for formulas.

We need to further specify sequences of defaults that are significant for a theory T: default processes.[2] 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 a minimal number of defaults for the sequence Π. Any segment $\Pi[k]$ is also a sequence. We can now define default processes:

Definition 6 (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 $\Pi[k] = (\delta_0, \ldots \delta_{k-1})$.*

We will use warrant specification sets $\mathcal{WS}^\Pi$ that are relativized to default processes. The kind of process that we are focusing on here is called *closed* process. A process Π is taken to be closed iff every $\delta \in D$ that is applicable to $In(\Pi)$ is already in Π.

The possibility to refer to warrants within the language enables us to model *undercutting* defeaters. They defeat other default reasons, not by contradicting their conclusions as rebutting reasons do, but by denying that their warrant provides support for the conclusion in an undercutting circumstance.

Definition 7 (Undercut) *A reason u undercuts reason t being a reason for a formula F in a set of* **JT$_{\mathbf{CS}}$***-closed formulas $\Gamma \subseteq In(\Pi[k])$ iff* $\bigvee_{(v) \in Sub(t)} u : \neg[v : (G \rightarrow H)] \in \Gamma \ and \ v : (G \rightarrow H) \in \mathcal{WS}^\Pi$.

Undercutting brings about revisions of default extensions, as we will show by way of an example in Section 3. For any default theory $T = (W, D)$, an agent always considers *potential extension* sets of **JT$_{\mathbf{CS}}$** formulas that meet the following conditions:

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

---

[2] Our definitions are inspired by Antoniou's [1] operational semantics for Reiter's default theories.

For any potentially acceptable set Γ we define the notion of acceptability of a justified formula $t : F$:

Definition 8 (Acceptability) *For 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 Γ iff for each reason u that undercuts t as a reason for F such that $u : G \in In(\Pi)$, $Th^{JT_{CS}}(\Gamma)$ undercuts u as a reason for G.*

Following [13], we provide formal argumentation semantics for justification assertions in terms of multiple extension notions for default theories (formal correspondence of $\mathbf{JT_{CS}}$ extensions to Dung's extensions is established in [13]).

Definition 9 ($\mathbf{JT_{CS}}$ 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 Γ is a $\mathbf{JT_{CS}}$ admissible extension of T iff $Th^{JT_{CS}}(\Gamma)$ is conflict-free and if each formula $t : F \in \Gamma$ is acceptable w.r.t. Γ.*

$\mathbf{JT_{CS}}$ **Preferred Extension** *A $\mathbf{JT_{CS}}$ admissible extension Γ, $Th^{JT_{CS}}(\Gamma)$ 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** *For closed processes Π and Π' of T, a set of $\mathbf{JT_{CS}}$ formulas $Th^{JT_{CS}}(\Gamma) \subseteq In(\Pi)$ is a $\mathbf{JT_{CS}}$ complete extension of T iff Γ is a $\mathbf{JT_{CS}}$ admissible extension of T and each formula $t : F \in cons(\Pi')$, if $Th^{JT_{CS}}(\Gamma) \cup \{t : F\}$ is $\mathbf{JT_{CS}}$ consistent and $t : F$ is acceptable w.r.t. Γ, 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** *For closed processes Π and Π' of T, a $\mathbf{JT_{CS}}$ stable extension is a $\mathbf{JT_{CS}}$ closure of a potential extension $\Gamma \subset In(\Pi)$ such that $(\mathbf{1})$ $Th^{JT_{CS}}(\Gamma)$ undercuts all the formulas $t : F \in In(\Pi)$ outside $Th^{JT_{CS}}(\Gamma)$ and $(\mathbf{2})$ for any formula $u : G \in \Gamma'$ such that $\Gamma' \subset In(\Pi')$ and $u : G \notin In(\Pi)$, it holds that $\Gamma \cup \{u : G\}$ is $\mathbf{JT_{CS}}$ inconsistent.*

3 Undercutting in justification logic and Reiter's logic

In this section, we want to illustrate the advantages of the expressiveness that our logic has in comparison to Reiter's default logic by means of an example with undercut. Consider an agent reasoning about whether a KLM Boeing 737 aircraft has cleared the take-off protocol or not, given that a source of information says that "the crosswind component at the default runway is at the speed of 35 knots" (C). Knowing that, at this speed of the crosswind component, the Boeing 737 type of aircraft is usually not allowed to proceed with taking off, the agent concludes that "the KLM Boeing 737 flight has been delayed" (K), according to the following default rule:

$$\delta_1 = \frac{r : C :: (s \cdot r) : K}{(s \cdot r) : K}.$$

The default can be read as follows: "If r is a reason justifying that the crosswind component at the default runway is at 35 knots and it is consistent to assume

that $(s \cdot r)$ is a reason justifying that the KLM Boeing 737 flight has been delayed, then $(s \cdot r)$ is a defeasible reason justifying that the KLM Boeing 737 flight has been delayed".

If the agent receives additional information that it is not the case that "the SAS Boeing 737 aircraft has been delayed" (S), then the agent has a reason to assume that "the aircraft can be allocated to an alternative runway" (R).

$$\delta_2 = \frac{t : \neg S :: (u \cdot t) : R}{(u \cdot t) : R}.$$

On a runway of a different orientation, the initial readings of the crosswind may even turn into a favorable headwind component. The information that there is an alternative runway undercuts the initial piece of reasoning codified by s, according to the following default rule:
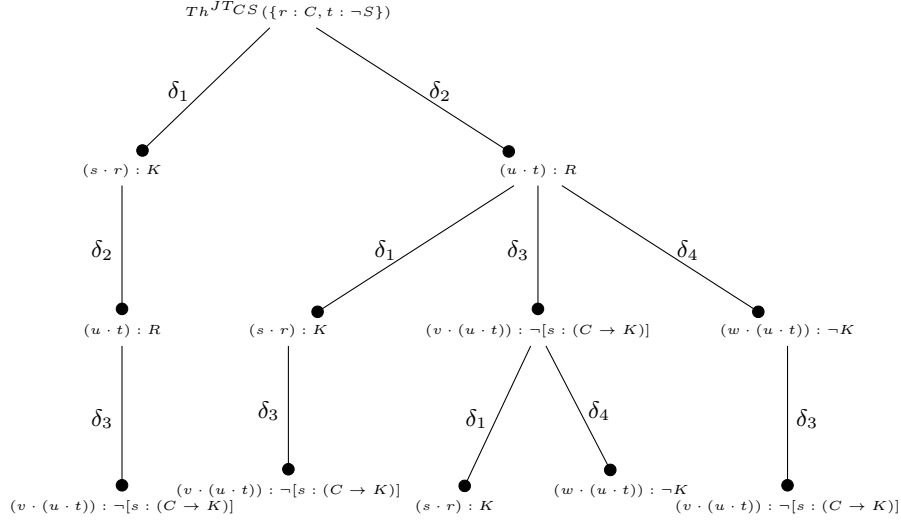
$$\delta_3 = \frac{(u \cdot t) : R :: (v \cdot (u \cdot t)) : \neg[s : (C \to K)]}{(v \cdot (u \cdot t)) : \neg[s : (C \to K)]}.$$

The consequent reads as follows: "$(v \cdot (u \cdot t))$ is a defeasible reason denying that the reason s justifies that if the crosswind component for the default runway is at the speed of 35 knots, then the KLM Boeing 737 flight has been delayed". Additionally, the agent has a reason to conclude that the KLM flight has not been delayed, grounded on the reasoning about an alternative runway:

$$\delta_4 = \frac{(u \cdot t) : R :: (w \cdot (u \cdot t)) : \neg K}{(w \cdot (u \cdot t)) : \neg K}.$$

Were it the case that the course of the agent's reasoning follows the proposed order, the agent would have to revise the conclusion supported by the reason $(s \cdot r)$. For a default theory $T_1 = (W, D)$ with $W = \{r : C, t : \neg S\}$ and $D = \{\delta_1, \delta_2, \delta_3, \delta_4\}$, the process $(\delta_1, \delta_2, \delta_3)$ corresponds to such course of reasoning with a revised $\mathbf{JT_{CS}}$ admissible extension. Figure 1 shows all the possible processes of T.

Is there a way to model undercut in Reiter's default logic, without extending the logic with, say, default priorities, as done in [10] and [5]? To answer this, we need to view default logic from the perspective of formal argumentation. The relation between formal argumentation and Reiter's default logic is known. It is shown in [7] that Reiter's default logic extensions can be defined in terms of stable extensions of Dung's frameworks and [13] shows that a large subclass of Dung's frameworks is a special case of our logic. But besides finding formal correspondences between their extensions, it is interesting to look at the conceptual relation of Reiter's logic to our justification logic. This relation is not straightforward, because the two logics are based on different underlying languages. This difference does not cause divergence in the way the two logics model rebuttal. Rebuttal is based on the workings of multiple incompatible extensions: two formulas extending some knowledge base rebut each other if they cannot both be included in a same default extension. However, the comparison of the ways in which the two logics deal with the concept of undercut reveals some immediate benefits of reifying default reasons in justification logic.

$$Th^{JTCS}(\{r : C, t : \neg S\})$$

Tree nodes and edges:

- δ_1 → $(s \cdot r) : K$
- δ_2 → $(u \cdot t) : R$

From $(s \cdot r) : K$:
- δ_2 → $(u \cdot t) : R$ → δ_3 → $(v \cdot (u \cdot t)) : \neg[s : (C \to K)]$

From $(u \cdot t) : R$:
- δ_1 → $(s \cdot r) : K$ → δ_3 → $(v \cdot (u \cdot t)) : \neg[s : (C \to K)]$
- δ_3 → $(v \cdot (u \cdot t)) : \neg[s : (C \to K)]$ → δ_1 → $(s \cdot r) : K$; δ_4 → $(w \cdot (u \cdot t)) : \neg K$
- δ_4 → $(w \cdot (u \cdot t)) : \neg K$ → δ_3 → $(v \cdot (u \cdot t)) : \neg[s : (C \to K)]$

Fig. 1. Each node of the process tree of T is labeled with an *In*-set after a default rule (edges) has been applied. Visually, we display only the formulas that are added to *In*-sets as a result of applying the available defaults.

The general form of a default rule in Reiter's logic is

$$\frac{\varphi : \psi_1, \ldots, \psi_m}{\chi},$$

for some predicate logic formulas φ, $\psi_1, \ldots, \psi_m$ and χ. An operational semantics similar to ours is given for Reiter's logic in [1]. Besides closure, [1] introduces an additional condition on the extension-producing processes of Reiter's default theories: *success*. A process is successful if each of the justifications $\psi_1, \ldots, \psi_m$ is consistent with the consequents added to an *In*-set after other defaults have been applied. To capture this formally, we define the set $Out(\Pi) = \{\neg\psi \mid \psi \in just(\delta)$ for some $\delta \in \Pi\}$, for some justification $just(\delta)$ of a rule δ in a process Π of Reiter's theory $\Delta = (W, D)$. None of the formulas from an *Out*-set should become a part of an *In*-set for the same process. For a set of first-order formulas $E = In(\Pi)$, E is an extension in Reiter's logic if and only if Π is both closed and successful.

In our logic, it is possible to consider consequents of defaults as arguments based on their underlying warrants. This enables representing conflicts simply by opposing reasons. In Reiter's logic, reasons are not reified and their conflicts cannot be reflected in the logical language. It is, however, possible to take the perspective of formal argumentation on Reiter's logic. To take such perspective, we follow [16, p. 52] in defining arguments in terms of finite processes of Reiter's theory and their mutual attacks through conflicts of *In*-sets with *Out*-sets. We start from defining attacks in terms of finite processes of a theory $\Delta = (W, D)$:

- Π attacks Π' if $\varphi \in In(\Pi)$ for some $\varphi \in Out(\Pi')$,

where Π and Π' are some finite processes of Δ. We can develop further on this definition to specify different kinds of attack:

- If all the default rules from Π and Π' could possibly form a finite process Π'' of Δ (in any possible order of the sequence), then the attack between Π and Π' is undercut. Otherwise, it is a rebuttal between Π and Π'.

The idea behind the refinement of the attack definition is that in Reiter's logic, non-normal default rules can be seen as a way to introduce "exclusionary reasons" [10] and undercut in Reiter's theory. Consider the following two Reiter's default rules:

$$\delta' = \frac{35Knots(crosswind) : \neg alternative(runway)}{delayed(KLM)},$$

saying that "if the crosswind component at the default runway is at 35 knots, and it is consistent to assume that the aircraft cannot be allocated to an alternative runway, then the KLM Boeing 737 flight has been delayed" and
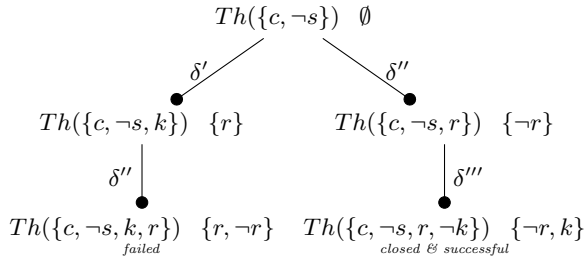
$$\delta'' = \frac{35Knots(crosswind) \wedge \neg delayed(SAS) : alternative(runway)}{alternative(runway)},$$

saying that "if the crosswind component at the default runway is at 35 knots and the SAS Boeing 737 flight has not been delayed, and it is consistent to assume that the flight can be allocated to an alternative runway, then the flight can be allocated to an alternative runway". Moreover, the following default is available to the agent:

$$\delta''' = \frac{alternative(runway) : \neg delayed(KLM)}{\neg delayed(KLM)}.$$

Take $\Delta = (W, D)$ to be a Reiter's default theory with $W = \{35Knots(crosswind), \neg delayed(SAS)\}$ and $D = \{\delta', \delta'', \delta'''\}$. The process tree for the Reiter's theory Δ is found in Figure 3. The idea of undercut can be illustrated by the way in which $\Pi' = (\delta'')$ attacks $\Pi = (\delta')$ via affirming the circumstance in which the conclusion of Π would not be reasonable any more, but it does not affirm the negation of that conclusion. This corresponds to the idea of undercut. Notice that, in contrast to δ_1 above, Reiter's rule δ' needs to include the negation of the exclusionary circumstance $alternative(runway)$ in conveying the idea of undercut.

Notice that the mechanism whereby Π' attacks Π is also responsible for the problem of process "destruction" [1, p. 63]. An example of process destruction occurs in the unsuccessful process $\Pi'' = (\delta', \delta'')$, where, after δ'' has been applied, Π'' becomes closed and unsuccessful. As it can be seen from the process tree of T, justification logic-based default theories are able to model interaction among default rules and undercut without having to resort to the use of process destruction. This is mainly due to the fact that the rule δ_3, which is "missing" in the Reiter's logic rendition of the example, brings forth the undercut of the default reason from δ_1 as a part of the example description.

$$Th(\{c, \neg s\}) \quad \emptyset$$

$$\delta' \qquad\qquad \delta''$$

$$Th(\{c, \neg s, k\}) \quad \{r\} \qquad\qquad Th(\{c, \neg s, r\}) \quad \{\neg r\}$$

$$\delta'' \qquad\qquad\qquad\qquad \delta'''$$

$$Th(\{c, \neg s, k, r\}) \quad \{r, \neg r\} \qquad Th(\{c, \neg s, r, \neg k\}) \quad \{\neg r, k\}$$
$$\text{\small failed} \qquad\qquad\qquad \text{\small closed \& successful}$$

Fig. 2. The process tree of Reiter's default theory Δ. Undercut is modelled by the failed process branch.

Several remarks are at hand by comparing the structures of the process trees of T and Δ. Although the theories model the same phenomenon, Reiter's logic noticeably simplifies the example. While some ways of simplifying are desirable, there are several reasons to prefer the representation of undercut in justification logic. One of them is that failed processes disable reinstatement of reasons that could, in principle, be reinstated by undercutting their undercutters. In our example above, even if the SAS flight has not been delayed, it might be the case that the current demand for the runway reassignment exceeds the operational capacities of the alternative runway. This, in turn, provides a reason to reinstate the initial reason in support of the claim that the KLM flight is delayed. Justification logic is able to represent such reason reinstatement and processes that are, in principle, infinitely extendable.

As can be seen from the definitions of $\mathbf{JT_{CS}}$ extensions, only $\mathbf{JT_{CS}}$ admissible extensions can be identified by looking at a single branch of the process tree of T. To determine the status of other $\mathbf{JT_{CS}}$ extensions, all reasons need to be taken into account.[3] In contrast, Reiter's default processes are self-contained with respect to the extensions status. Therefore, $\mathbf{JT_{CS}}$ extensions have more in common with the notion of Reiter's logic consequence relation (*credulous* and *skeptical*). For instance, $\mathbf{JT_{CS}}$ grounded extensions correspond to the skeptical notion of validity, which amounts to the intersection of all Reiter's extensions. It is only in taking the argumentation perspective on Reiter's processes that we need to look at the dependencies of different process tree branches.

Conceptually, the most important advantage of representing defeasible reasoning in justification logic is that neither for reaching a defeasible conclusion nor for undercutting that reason, agents do not need to anticipate exclusionary reasons. In an important sense, anticipating exclusionary reasons with $\neg alternative(runway)$ in the Reiter's rule δ' above goes against the idea of default reasoning. Namely, in the sense that the number of these conditions may be infinite. The need to anticipate exclusionary reasons brings us back to the initial problem: we want to find out how to *avoid* anticipating numerous exceptions before an agent is able to reach a conclusion. It is an advantage of our theory to be able to model exceptions to rules via undercut, but without the drawback of guessing all the conditions of undercut within a default theory.

---

[3] This could be further amended by considering "anytime reasoning" methods as, for example, those proposed in [6] for Reiter's logic.

Finally, notice that by using Reiter's non-normal defaults, we are not only able to define undercut, but also to define theories such as $\Delta^* = (W^*, D^*)$, where $W^* = \{\emptyset\}$ and $D^* = \{\delta'''' = \frac{\emptyset : \neg A}{A}\}$. The rule $\frac{\emptyset : \neg A}{A}$ invalidates its own applicability. Using the above defined translation to argument frameworks, it is possible to build a single-argument attack cycle in terms of the argument $\Pi = (\delta'''')$. This kind of attack is well-known from Dung's abstract argumentation frameworks [7]. In [13], it is proved that defining single-argument attack cycles is not possible once arguments' warrants have been included as underlying rules for each default. This shows that although δ'''' sanctions "jumping" to the conclusion A, this does not mean that the type of inference it instantiates counts a reasoned defeasible step.

4   Related work and conclusion

Some existing extensions of default theories that can deal with the problem of exclusionary reasons come close to our intention of reifying default reasons. Most notably, approaches that are based on reasoning about default rule priorities such as [5, 9, 10] include a variant of reasoning about other reasons, namely, by reasoning about the relative weights of defaults. Strictly speaking, reasoning about default priorities reifies default rules, not default reasons, by extending the underlying language of default logic with default names and a predicate symbol that represents priorities among defaults. In such default theories, agents may arrive to conclusions about which ordering of defaults is a preferred one and to, thereby, consider higher priority as a source of defeat. Priority weighing in the style of [5] can be represented in process trees, as done in [1, pp. 97-98]. It can be noticed that one of the difficulties with such reasoning is that before applying a default, an agent needs to consider all other applicable defaults. As in Reiter's logic, processes are possibly failed, where failure is now due to making application choices that are inconsistent with a valid ordering among defaults. Horty [10, p. 124] defines "exclusionary default theories" where he explicitly includes undercutters, but his undercut is logically only a predicate saying that a rule has been excluded.

A more elaborate study of the different ways to defeat reasons is carried out in argumentation theory, from the classical account of undercut and rebuttal in [14] to some later formal argumentation frameworks such as, e.g., [15, 4, 20]. These frameworks are not based on default logic nor do they base their formalism on a language with formulas that feature reason terms. Hence, as for our current discussion on the problem of reification, such systems do not provide explicit answers. Among justification logic systems, some of them [3, 18] combine belief revision and dynamic epistemic logic techniques to model defeat, which is closest in its kind to undermining. However, none of them is able to model undercut or to encode defeasibility in the structure of reason terms.

Finally, our answer to the problem of reification is that both a prerequisite of a default rule and the rule itself are involved in reifying default reasons. This is reflected in the way in which default application codifies default steps from

warrants and prerequisites of defaults to their consequents. An immediate advantage of reification is that, by referring to such reason-producing steps within the object language, we can provide a fine-grained logical account of defeat.

References

1. Antoniou, G.: Nonmonotonic Reasoning. Cambridge, MA: MIT Press (1997)
2. Artemov, S.N.: Explicit provability and constructive semantics. Bulletin of Symbolic Logic pp. 1–36 (2001)
3. Baltag, A., Renne, B., Smets, S.: The logic of justified belief, explicit knowledge, and conclusive evidence. Annals of Pure and Applied Logic **165**(1), 49–81 (2014)
4. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence **128**(1-2), 203–235 (2001)
5. Brewka, G.: Reasoning about priorities in default logic. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI'94. vol. 2, pp. 940–945. AAAI Press/The MIT Press (1994)
6. Cadoli, M., Schaerf, M.: Approximate inference in default logic and circumscription. Fundamenta Informaticae **21**(1, 2), 103–112 (1994)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321–357 (1995)
8. Fitting, M.: Justification logics, logics of knowledge, and conservativity. Annals of Mathematics and Artificial Intelligence **53**(1-4), 153–167 (2008)
9. Horty, J.F.: Reasons as defaults. Philosopher's Imprint **7**(3), 1–28 (2007)
10. Horty, J.F.: Reasons as Defaults. Oxford University Press (2012)
11. 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-Verlag (1997)
12. Pandžić, S.: A logic of default justifications. In: Fermé, E., Villata, S. (eds.) 17th International Workshop on Nonmonotonic Reasoning, NMR 2018. pp. 126–135 (2018)
13. Pandžić, S.: Justification logic as a theory of structured arguments. Unpublished manuscript (2019)
14. Pollock, J.L.: Defeasible reasoning. Cognitive Science **11**(4), 481–518 (1987)
15. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation **1**(2), 93–124 (2010)
16. Prakken, H.: Commonsense Reasoning and Argumentation. Utrecht University (2018), course reader
17. Reiter, R.: A logic for default reasoning. Artificial Intelligence **13**(1-2), 81–132 (1980)
18. Renne, B.: Multi-agent justification logic: Communication and evidence elimination. Synthese **185**(1), 43–82 (2012)
19. Toulmin, S.E.: The Uses of Argument. Cambridge University Press (2003)
20. Verheij, B.: Correct grounded reasoning with presumptive arguments. In: Loizos, M., Kakas, A. (eds.) European Conference on Logics in Artificial Intelligence, JELIA 2016. pp. 481–496. Springer (2016)