A formalism unifying Defeasible Logics and Repair Semantics for existential rules

Abdelraouf Hecham, Pierre Bisquert, Madalina Croitoru

A formalism unifying Defeasible Logics and Repair Semantics for existential rules

Abdelraouf Hecham[*,1], Pierre Bisquert[†,2], and Madalina Croitoru[‡,1]

[1]INRIA GraphIK, Université de Montpellier, France
[2]INRAE IATE, INRIA GraphIK, France

Abstract

Two prominent ways of handling inconsistency provided by the state of the art are repair semantics and Defeasible Logics. In this paper we place ourselves in the setting of inconsistent knowledge bases expressed using existential rules and investigate how these approaches relate to each other. We run an experiment that checks how human intuitions align with those of either repair-based or defeasible methods and propose a new semantics combining both worlds.

1 Introduction

Conflicts in knowledge representation cause severe problems, notably due the principle of explosion (from falsehood anything follows). These conflicts arise from two possible sources: either the facts are incorrect (known as *inconsistence*), or the rules themselves are contradictory (known as *incoherence*). In order to preserve the ability to reason in presence of conflicts, several approaches can be used, in particular Defeasible Logics [18] [8] and Repair Semantics [16]. These two approaches stem from different needs and address conflicts in different ways. A key difference between defeasible logics and Repair Semantics is that the first was designed for incoherence while the latter was designed for inconsistence. However, since inconsistence is a special type of incoherence [11], defeasible logics can be applied to inconsistent but coherent knowledge, and thus be compared to the Repair Semantics. In this paper we want to investigate how the different intuitions of defeasible logics and Repair Semantics relate to each other. In order to attain the above mentioned objective, we make use of a combinatorial structure called Statement Graph [13]. Statement Graphs have been defined as way to reason defeasibly with existential rules using forward chaining. The reasoning is based on labeling functions shown to correspond to various flavors of Defeasible Logics. This paper proposes a new labeling for Repair Semantics and paves the way to combine both conflict-tolerant approaches in one unifying formalism.

---

[*]hecham@lirmm.fr

[†]pierre.bisquert@inrae.fr

[‡]croitoru@lirmm.fr

2 Logical Language

We consider a first order language Ł with constants but no other function symbol composed of formulas built with the usual quantifiers $(\exists, \forall)$ and the connectives $(\rightarrow, \wedge)$, on a vocabulary constituted of infinite sets of predicates, constants and variables. A *fact* is a ground atom (an atom with only constants) or an existentially closed atom. An existential *rule* (a.k.a. a tuple generating dependency) r is a formula of the form $\forall \vec{X}, \vec{Y} \left(\mathcal{B}(\vec{X}, \vec{Y}) \rightarrow \exists \vec{Z} \, \mathcal{H}(\vec{X}, \vec{Z}) \right)$ where $\vec{X}, \vec{Y}$ are tuples of variables, $\vec{Z}$ is a tuple of *existential variables*, and $\mathcal{B}, \mathcal{H}$ are finite non-empty conjunctions of atoms respectively called *body* and *head* of r and denoted $Body(r)$ and $Head(r)$. In this paper we consider rules with atomic head (any rule can be transformed into a set of rules with atomic head [4]). A *negative constraint* is a rule of the form $\forall \vec{X} \, \mathcal{B}(\vec{X}) \rightarrow \perp$ where $\mathcal{B}$ is a conjunction of atoms and $\vec{X}$ is a set of variables with possibly constants. Negative constraints are used to express conflicts. In this paper, we take into account binary negative constraints, which contain only two atoms in their body.[1] A knowledge base is a tuple $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}$ is a finite set of facts, $\mathcal{R}$ is a finite set of rules, and $\mathcal{N}$ is a finite set of negative constraints.

We denote the set of ***models*** of a knowledge base by $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N})$.

A ***derivation*** is a (potentially infinite) sequence of tuples $D_i = (\mathcal{F}_i, r_i, \pi_i)$ composed of a set of facts $\mathcal{F}_i$, a rule r_i and a homomorphism π_i from $Body(r_i)$ to $\mathcal{F}_i$, where $D_0 = (\mathcal{F}, \emptyset, \emptyset)$ and such that $\mathcal{F}_i$ results from the application of rule r_i to F_{i-1} according to π_i, i.e. $\mathcal{F}_i = \alpha(\mathcal{F}_{i-1}, r_i, \pi_i)$. A derivation from a set of facts $\mathcal{F}$ to a fact f is a minimal sequence of rules applications starting from $D_0 = (\mathcal{F}_0 \subseteq \mathcal{F}, \emptyset, \emptyset)$ and ending with $D_n = (\mathcal{F}_n, r_n, \pi_n)$ such that $f \in \mathcal{F}_n$.

A ***chase*** (a.k.a. forward chaining) is the exhaustive application of a set of rules over a set of facts in a breadth-first manner (denoted $chase(\mathcal{F}, \mathcal{R})$) until no new facts are generated, the resulting "saturated" set of all initial and generated facts is denoted $\mathcal{F}^*$. While this is not always guaranteed to stop, certain recognizable classes of existential rules that are decidable for forward chaining have been defined [3]; we limit ourselves to the recognizable FES (Finite Expansion Set) class [5] and use Skolem chase [17]. We consider ground atomic queries. We denote that a query is entailed from a knowledge base $\mathcal{KB}$ by $\mathcal{KB} \models Q$ (equivalent to $chase(\mathcal{F}, \mathcal{R}) \models Q$ [9]).

Inconsistence vs. Incoherence. Conflicts appear in a knowledge base whenever a negative constraint becomes applicable: we say that two facts f_1 and f_2 are in ***conflict*** if the body of a negative constraint can be mapped to $\{f_1, f_2\}$. There are two possible sources of conflicts, either the facts are incorrect (known as *inconsistence*), or the rules themselves are contradictory (known as *incoherence*). A $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is *inconsistent* iff it has an empty set of models (i.e. $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N}) = \emptyset$). A knowledge base is *incoherent* iff $\mathcal{R} \cup \mathcal{N}$ are unsatisfiable, meaning that there does not exist any set of facts S (even outside the facts of the knowledge base) where all rules in $\mathcal{R}$ are applicable such that $models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$ [11].

Example 1 (Inconsistence) *Consider the following $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ that describes a simplified legal situation: If there is a scientific evidence incriminating a defendant then*

---

[1]It should be noted that this restriction does not lead to a loss of expressive power, as [2] shows.

he is responsible for the crime, if there is a scientific evidence absolving a defendant then he is not responsible for the crime. A defendant is guilty if responsibility is proven. If a defendant is guilty then he will be given a sentence. If a defendant has an alibi then he is innocent. There is a scientific evidence "e1" incriminating a female defendant "alice", while another scientific evidence "e2" is absolving her of the crime. She also has an alibi. Is Alice innocent (i.e. $Q_1 = innocent(alice)$)? Is she guilty (i.e. $Q_2 = guilty(alice)$)?

- $\mathcal{F} = \{incrim(e1, alice), absolv(e2, alice), alibi(alice), female(alice)\}$

- $\mathcal{R} = \{r_1 : \forall X, Y\ incrim(X, Y) \rightarrow resp(Y),$
  $r_2 : \forall X, Y\ absolv(X, Y) \rightarrow notResp(X),$
  $r_3 : \forall X\ resp(X) \rightarrow guilty(X),$
  $r_4 : \forall X\ alibi(X) \rightarrow innocent(X),$
  $r_5 : \forall X\ guilty(X) \rightarrow \exists Y\ sentence(X, Y)\}$

- $\mathcal{N} = \{\forall X\ resp(X) \wedge notResp(X) \rightarrow \bot,$
  $\forall X\ guilty(X) \wedge innocent(X) \rightarrow \bot\}$

The saturated set of facts resulting from the chase is

- $\mathcal{F}^* \quad = \quad \{incrim(e1, alice), \quad absolv(e2, alice), \quad alibi(alice),$
  $female(alice),\ resp(alice),\ notResp(alice),\ guilty(alice),\ innocent(alice),$
  $sentence(alice, null_1)\}.$

*This knowledge base is **inconsistent**, because a negative constraint is applicable, (thus $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N}) = \emptyset$). This inconsistency is due to an erroneous set of facts (either one of the evidences, the alibi, or all of them are not valid). The classical answer to the Boolean queries Q_1 and Q_2 is "true" (i.e. Alice is guilty and innocent), because from falsehood, anything follows. However, the knowledge base is **coherent** because the set of rules are satisfiable i.e. there exists a set of facts (e.g. $\mathcal{F}' = \{incrim(e1, \mathbf{bob}), absolv(e3, \mathbf{alice}), alibi(\mathbf{alice})\}$) such that all rules are applicable and $models(\mathcal{F}', \mathcal{R} \cup \mathcal{N})) \neq \emptyset$.*

Inconsistency Handling. Defeasible Logics and Repair Semantics are two approaches to handle conflicts. Defeasible Logics are applied to potentially incoherent situations were two types of rules are considered: strict rules expressing undeniable implications (i.e. if $\mathcal{B}(r)$ then definitely $\mathcal{H}(r)$), and defeasible rules expressing weaker implications (i.e. if $\mathcal{B}(r)$ then generally $\mathcal{H}(r)$). In this context, contradictions stem from either relying on incorrect facts, or from having exceptions to the defeasible implications.

Repair Semantics are applied to situations where rules are assumed to hold in the true state of affair and hence inconsistencies can only stem from incorrect facts. A *repair* $\mathcal{D}$ is an inclusion-maximal subset of the facts $\mathcal{D} \subseteq \mathcal{F}$ that is consistent with the rules and negative constraints (i.e. $models(\mathcal{D}, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$). We denote the set of repairs of a knowledge base by $repairs(\mathcal{KB})$. In presence of an incoherent set of rules, Repair Semantics yield an empty set of repairs [10].

3 Statement Graphs for Defeasible Reasoning.

A Statement Graph (SG) [13] is a representation of the reasoning process happening inside a knowledge base, it can be seen as an updated Inheritance Net [15] with a custom labeling function. An SG is built using logical building blocks (called statements) that describe a situation (premises) and a rule that can be applied on that situation.

A statement $\mathtt{s}$ is a pair that is either a 'query statement' $(Q, \emptyset)$ where Q is a query, a 'fact statement' $(\top, f)$ where f is a fact, or a 'rule application statement' (Φ, ψ) that represents a rule application $\alpha(\mathcal{F}, r, \pi)$ s.t. $\pi(\mathcal{B}(r)) = \Phi$ and $\pi(\mathcal{H}(r)) = \psi$. We denote by $Prem(\mathtt{s})$ the first element of the statement and by $Conc(\mathtt{s})$ the second element. A statement can be written as $Prem(\mathtt{s}) \to Conc(\mathtt{s})$.

A statement $\mathtt{s}_1$ **supports** another statement $\mathtt{s}_2$ iff $\exists f \in Prem(\mathtt{s}_2)$ s.t. $Conc(\mathtt{s}_1) = f$. A statement $\mathtt{s}_1$ **attacks** $\mathtt{s}_2$ $\exists f \in Prem(\mathtt{s}_2)$ s.t. $Conc(\mathtt{s}_1)$ and f are in conflict.

Statements are generated from a knowledge base, they can be structured in a graph according to the support and attack relations they have between each other.

Definition 1 (Statement Graph) *A Statement Graph of a $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is a directed graph $\mathbb{SG}_{\mathcal{KB}} = (\mathcal{V}, \mathcal{E}_A, \mathcal{E}_S)$: $\mathcal{V}$ is the set of **statements** generated from $\mathcal{KB}$; $\mathcal{E}_A \subseteq \mathcal{V} \times \mathcal{V}$ is the set of **attack edges** and $\mathcal{E}_S \subseteq \mathcal{V} \times \mathcal{V}$ is the set of **support edges**.*

For an edge $e = (\mathtt{s}_1, \mathtt{s}_2)$, we denote $\mathtt{s}_1$ by $Source(e)$ and $\mathtt{s}_2$ by $Target(e)$. For a statement $\mathtt{s}$ we denote its incoming attack edges by $\mathcal{E}_A^-(\mathtt{s})$ and its incoming support edges by $\mathcal{E}_S^-(\mathtt{s})$. We also denote its outgoing attack edges by $\mathcal{E}_A^+(\mathtt{s})$ and outgoing support edges by $\mathcal{E}_S^+(\mathtt{s})$.

A Statement Graph (SG) is constructed from the chase of a knowledge base. Starting facts are represented by fact statements and rule applications are represented using rule application statements. Figure 1 shows SG of Example 1.

An SG provides statements and edges with a label using a *labeling function*. Query answering can then be determined based on the label of the query statement. Continuing the previous example, in an **ambiguity propagating** setting (such as [12]), $innocent(alice)$ is ambiguous because $guilty(alice)$ can be derived, thus the ambiguity of $guilty(alice)$ is propagated to $innocent(alice)$ and consequently $\mathcal{KB} \nvDash_{prop}$ $innocent(alice)$ and $\mathcal{KB} \nvDash_{prop} guilty(alice)$ ($\vDash_{prop}$ denotes entailment in ambiguity propagating). On the other hand, in an **ambiguity blocking** setting (such as [18]), the ambiguity of $resp(alice)$ blocks any ambiguity derived from it, meaning that $guilty(alice)$ cannot be used to attack $innocent(alice)$. Therefore $innocent(alice)$ is not ambiguous, thus $\mathcal{KB} \vDash_{block} innocent(alice)$ and $\mathcal{KB} \nvDash_{block} guilty(alice)$ ($\vDash_{block}$ denotes entailment in ambiguity blocking). The labeling function '**PDL**' (Propagating Defeasible Logic) was proposed for Statement Graphs in [13] that yields equivalent entailment results to Defeasible Logics with ambiguity propagating [1]. Similarily, the labeling function '**BDL**' (Blocking Defeasible Logic) was proposed by [13] to obtain entailment results equivalent to Defeasible Logics with ambiguity blocking [7].
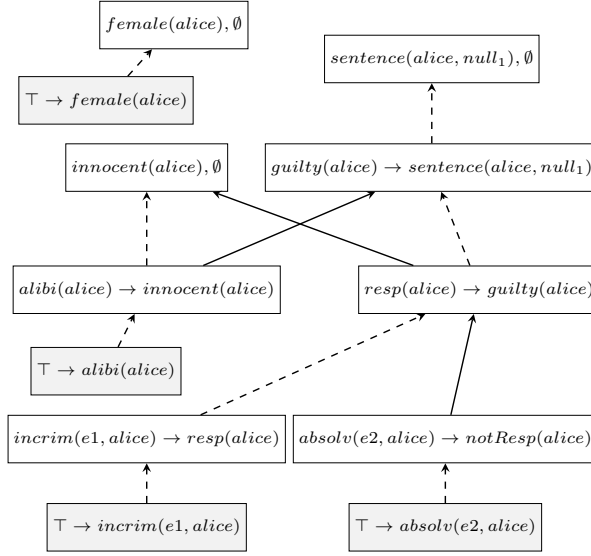
Figure 1: Example 1's Statement Graph (fact statements are gray).

4 Statement Graphs for Repair Semantics

In this paper we focus on two well-known semantics for inconsistent databases: IAR and ICAR repair Semantics. The Intersection of All Repairs semantic [16] is the most skeptical of the Repair Semantics. A query Q is IAR entailed ($\mathcal{KB} \vDash_{IAR} Q$) iff it is classically entailed by the intersection of all repairs constructed from the starting set of facts (i.e. $\bigcap repairs(\mathcal{KB}) \cup \mathcal{R} \vDash Q$). The Intersection of Closed ABox Repairs semantic [16] computes the repairs of the saturated set of facts. A query Q is ICAR entailed $\mathcal{KB} \vDash_{ICAR} Q$ iff it is classically entailed by the set of facts in the intersection of the repairs constructed after generating all facts.

Example 2 *Consider the $\mathcal{KB}$ in Example 1. The repairs constructed from the starting set of facts are:*

- $\mathcal{D}_1 = \{absolv(e2, alice), alibi(alice), female(alice)\}$

- $\mathcal{D}_2 = \{incrim(e1, alice), female(alice)\}$

$\mathcal{D}_1 \cap \mathcal{D}_2 = \{female(alice)\}$ *therefore only $female(alice)$ is entailed: $\mathcal{KB} \vDash_{IAR} female(alice)$. The repairs constructed from the saturated set of facts are:*

- $\mathcal{D}'_1 = \{absolv(e2, alice), alibi(alice), female(alice), notResp(alice), sentence(alice, null_1)\}$

- $\mathcal{D}'_2 = \{incrim(e1, alice), female(alice), resp(alice), guilty(alice), sentence(alice, null_1)\}$

$\mathcal{D}'_1 \cap \mathcal{D}'_2 = \{female(alice), sentence(alice, null_1)\}$ *thus $\mathcal{KB} \vDash_{ICAR} female(alice) \wedge sentence(alice, null_1)$.*

5

4.1 New labeling for IAR Semantics

The intuition behind IAR is to reject any fact that can be used to generate conflicting atoms, meaning that only the facts that produce no conflict will be accepted. From an SG point of view, any statement that is attacked, or that supports statements that lead by an attack or support edge to an attacked statement is discarded. This can be obtained by first detecting all conflicts, then discarding any statement that either leads to a conflict or is generated from conflicting atoms. In order to detect conflicts using Statement Graphs, we need to ensure that all conflicts are represented. Given that statements attack each other on the premise, it is necessary to handle in a particular way statements with no outgoing edges (i.e. statements that do not support or attack other statements) as they might still generate conflicting atoms. That is why any statement with no outgoing edges must be linked to a query statement. We first apply PDL to detect ambiguous statements, then backwardly broadcast this ambiguity to any statement that is linked (by a support or attack edge) to an ambiguous statement (cf. Figure 2). Labelings for Defeasible Logics start from fact statements and propagate upward towards query statements, however, for Repair Semantics, the labelings have to conduct a second pass from query statements and propagate downward towards fact statements. We use the labeling function '**IAR**' to obtain entailment results equivalent to IAR [16]. IAR is defined as follows: edges have the same label as their source statements (i.e. given an edge e, $\text{IAR}(e) = \text{IAR}(Source(e))$. Given a statement s:

(a) $\textbf{\textit{IAR}}(s) = \textbf{\textit{IN}}$ *iff IAR$(s) \neq$ AMBIG and PDL$(s) = $ IN.*

(b) $\textbf{\textit{IAR}}(s) = \textbf{\textit{AMBIG}}$ *iff either PDL$(s) = $ AMBIG or $\exists e \in \mathcal{E}_S^+(s) \cup \mathcal{E}_A^+(s)$ such that IAR$(Target(e)) = $ AMBIG.*

(c) $\textbf{\textit{IAR}}(s) = \textbf{\textit{OUT}}$ *iff PDL$(s) = $ OUT.*

A statement is labeled AMBIG if it was labeled ambiguous by PDL or if it leads to an ambiguous statement. Otherwise, it is IN if it has an IN complete support and is not attacked (i.e PDL labels it IN).

In the following proposition, we will denote by $\mathbb{SG}_{\mathcal{KB}}^{\text{IAR}}$ an SG built on the KB $\mathcal{KB}$ that uses the ICAR labeling function and by $\mathbb{SG}_{\mathcal{KB}}^{\text{IAR}}\langle s \rangle$ the label of a statement s.

Proposition 1 *Let f be a fact in a $\mathcal{KB}$ that contains only defeasible facts and strict rules. $\mathcal{KB} \vDash_{IAR} f$ iff $\mathbb{SG}_{\mathcal{KB}}^{IAR}\langle(f, \emptyset)\rangle = IN$ and $\mathcal{KB} \nvDash_{IAR} f$ iff $\mathbb{SG}_{\mathcal{KB}}^{IAR}\langle(f, \emptyset)\rangle \in \{AMBIG, OUT\}$.*

We split the proof of (1.) in two parts, first we prove by contradiction that if $\mathcal{KB} \vDash_{IAR} f$ then $\mathbb{SG}_{\mathcal{KB}}^{\text{IAR}}\langle(f, \emptyset)\rangle = $ IN: Suppose we have a fact f such that $\mathcal{KB} \vDash_{IAR} f$ and $\mathbb{SG}_{\mathcal{KB}}^{\text{IAR}}\langle(f, \emptyset)\rangle \neq $ IN:

1. $\mathcal{KB} \vDash_{IAR} f$ means that there is a derivation for f from an initial set of facts $T \subseteq \mathcal{F}$ and there is no consistent set of initial facts $S \subseteq \mathcal{F}$ such that $S \cup T$ is inconsistent (i.e $models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$ and $models(S \cup T, \mathcal{R} \cup \mathcal{N}) = \emptyset$), which means that there is no derivation for an atom conflicting with an atom used in the derivation for f i.e. f is not generated from or used to generate ambiguous atoms, thus $PDL((f \rightarrow \emptyset)) = $ IN.

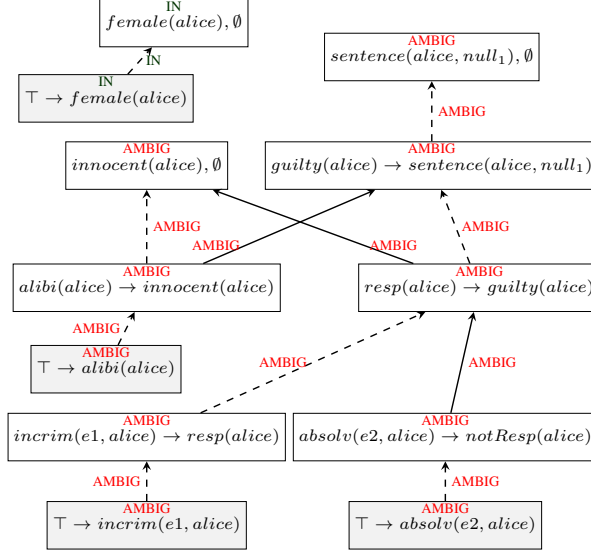Figure 2: IAR applied to Example 1's Statement Graph.

2. $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle \neq$ IN means that either:

   (a) $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle =$ OUT which is impossible given 1. (i.e. $\text{PDL}(f,\emptyset) =$ IN)

   (b) or $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle =$ AMBIG which means either:

      i. $\text{PDL}(f,\emptyset) =$ AMBIG (impossible given 1.),

      ii. or $\exists e \in \mathcal{E}^+_S(\mathbb{s}) \cup \mathcal{E}^+_A(\mathbb{s})$ such that $\text{IAR}(Target(e)) =$ AMBIG which means that f is used to generate ambiguous atoms (impossible given 1.).

Now we prove by contradiction that if $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle =$ IN then $\mathcal{KB} \vDash_{IAR} f$: Suppose we have a fact f such that $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle =$ IN and $\mathcal{KB} \nvDash_{IAR} f$:

1. $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f,\emptyset)\rangle =$ IN means that $\text{IAR}(f,\emptyset) \neq$ AMBIG and $\text{PDL}(f,\emptyset) =$ IN, which means that $(f,\emptyset)$ is not attacked (i.e. there is no derivation for an atom conflicting with f) and is not used to generate conflicting atoms (no outgoing edge leads to an AMBIG statement).

2. $\mathcal{KB} \nvDash_{IAR} f$ means that either f is generated by conflicting atoms (impossible given 1.) or is used to generate conflicting atoms (impossible given 1.).

From (1.) the proposition (2.) directly holds ($\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f \to \emptyset)\rangle \neq$ IN means $\mathbb{SG}^{\text{IAR}}_{\mathcal{KB}}\langle(f \to \emptyset)\rangle \in \{$AMBIG, OUT$\}$ given that IAR is a function).

4.2 New labeling for ICAR Semantics.

The intuition behind ICAR is to reject any fact that is used to generate conflict, while accepting those that do not (even if they were generated after a conflict). From an SG

7

point of view, any statement that is attacked or that supports statements that lead to an attack is considered "ambiguous". This is done by first applying PDL to detect ambiguous and accepted statements then the ICAR labeling starts from query statements and propagates downward towards fact statements (cf. Figure 3). We use the labeling function '**ICAR**' to obtain entailment results equivalent to ICAR [16]. ICAR is defined as follows: given an edge e, $ICAR(e) = ICAR(Source(e))$. Given a statement $\mathbb{s}$:

(a) **$ICAR(\mathbb{s}) = IN$** *iff $ICAR(\mathbb{s}) \neq AMBIG$ and $PDL(\mathbb{s}) \in \{IN, AMBIG\}$.*

(b) **$ICAR(\mathbb{s}) = AMBIG$** *iff*

1. *either $PDL(\mathbb{s}) = AMBIG$ and $\exists e \in \mathcal{E}_A^-(\mathbb{s})$ s.t. $PDL(e) \in \{IN, AMBIG\}$,*
2. *or $\exists e \in \mathcal{E}_S^+(\mathbb{s}) \cup \mathcal{E}_A^+(\mathbb{s})$ such that $ICAR(Target(e)) = AMBIG$.*

(c) **$ICAR(\mathbb{s}) = OUT$** *iff $PDL(\mathbb{s}) = OUT$.*

A statement is labeled AMBIG if it was labeled ambiguous by PDL and it is attacked, or if it leads to an ambiguous statement. It is labeled IN if it was labeled IN or AMBIG by PDL and does not lead to an ambiguous statement.
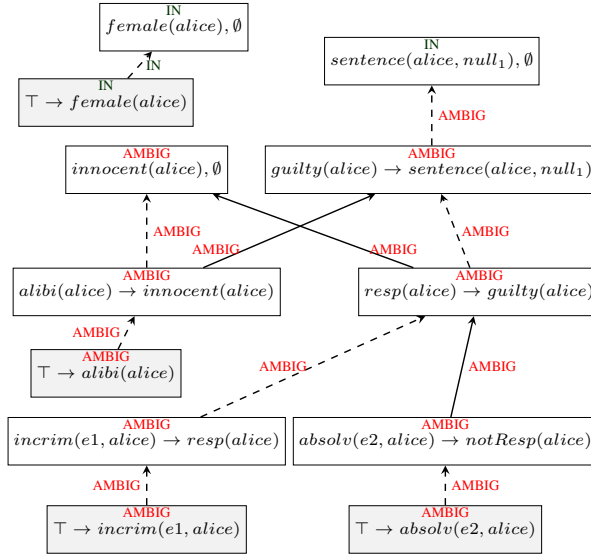
Figure 3: ICAR applied to Example 1's Statement Graph.

In the following proposition, we will denote by $\mathbb{SG}_{\mathcal{KB}}^{ICAR}$ an SG built on the KB $\mathcal{KB}$ that uses the ICAR labeling function and by $\mathbb{SG}_{\mathcal{KB}}^{ICAR}\langle \mathbb{s} \rangle$ the label of a statement $\mathbb{s}$.

Proposition 2 *Let f be a fact in a $\mathcal{KB}$ that contains only defeasible facts and strict rules: $\mathcal{KB} \vDash_{ICAR} f$ iff $\mathbb{SG}_{\mathcal{KB}}^{ICAR}\langle(f, \emptyset)\rangle = IN$ and $\mathcal{KB} \nvDash_{ICAR} f$ iff $\mathbb{SG}_{\mathcal{KB}}^{ICAR}\langle(f, \emptyset)\rangle \in \{AMBIG, OUT\}$.*

We split the proof of (1.) in two parts, first we prove by contradiction that if $\mathcal{KB} \vDash_{ICAR} f$ then $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{IN}$. Suppose we have a fact f such that $\mathcal{KB} \vDash_{ICAR} f$ and $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle \neq \text{IN}$:

1. $\mathcal{KB} \vDash_{ICAR} f$ means that there is a derivation for f and there is no consistent set of facts $S \subseteq \mathcal{F}^*$ such that $S \cup \{f\}$ is inconsistent ($models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$ and $models(S \cup \{f\}, \mathcal{R} \cup \mathcal{N}) = \emptyset$) i.e f is not used to generate ambiguous atoms.

2. $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle \neq \text{IN}$ means that either:

    (a) $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{OUT}$ which is impossible given 1. (there is a derivation for f i.e. $\text{PDL}(f, \emptyset) \in \{\text{IN}, \text{AMBIG}\}$).

    (b) or $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{AMBIG}$ which means either:

        i. $\text{PDL}(f, \emptyset) = \text{AMBIG}$ and there is an edge attacking it (impossible given 1. i.e. there is no derivable conflicting atom with f).

        ii. or $\exists e \in \mathcal{E}_S^+(\mathbb{s}) \cup \mathcal{E}_A^+(\mathbb{s})$ such that $\text{ICAR}(Target(e)) = \text{AMBIG}$ which means that f is used to generate ambiguous atoms (impossible given 1.).

Now we prove by contradiction that if $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{IN}$ then $\mathcal{KB} \vDash_{ICAR} f$: Suppose we have a fact f such that $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{IN}$ and $\mathcal{KB} \nvDash_{ICAR} f$:

1. $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle = \text{IN}$ means that $\text{ICAR}(f, \emptyset) \neq \text{AMBIG}$ and $\text{PDL}(f, \emptyset) \in \{\text{IN}, \text{AMBIG}\}$, which means that $(f, \emptyset)$ is not attacked (i.e. there is no derivation for an atom conflicting with f) and it is used to generate ambiguous atoms (no outgoing edge leads to an AMBIG statement).

2. $\mathcal{KB} \nvDash_{ICAR} f$ means that either f is not derivable (impossible given 1. since $\text{PDL}(f, \emptyset) \in \{\text{IN}, \text{AMBIG}\}$), or there is a derivation for an atom conflicting with f, or f is used to generate ambiguous atoms (impossible given 1.).

From (1.) the proposition (2.) directly holds ($\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle \neq \text{IN}$ means $\mathbb{SG}_{\mathcal{KB}}^{\text{ICAR}}\langle(f, \emptyset)\rangle \in \{\text{AMBIG}, \text{OUT}\}$ given that ICAR is a function).

5 Human intuitions for conflict management

The contribution of the paper is two fold. On one hand we have provided new labelings for Statement Graphs shown to capture repair semantics. In this section we go one step further and show (1) there is practical value into combining defeasible reasoning and repair semantics and (2) provide a Statement Graph labeling for this new semantics.

In order to get an idea of what intuitions humans follow in an abstract context, we ran an experiment with 41 participants in which they were told to place themselves in the shoes of an engineer trying to analyze a situation based on a set of sensors. These sensors (with unknown reliability) give information about the properties of an object called "o", e.g. "Object 'o' has the property P" (which could be interpreted for example as 'o' is red). Also, as an engineer, they have a knowledge that is always true about the relations between these properties, e.g. "All objects that have the property P, also have

the property Q". Some of the properties cannot be true at the same time on the same object, e.g. "An object cannot have the properties P and T at the same time". Using abstract situations allowed us to avoid unwanted effects of *a priori* knowledge while at the same time representing formal concepts (facts, rules and negative constraints) in a textual simplified manner. A transcript of the original text that the experiment participants have received is shown in the following Example 3.

Example 3 (Situation 1) *Textual representation: Three sensors are respectively indicating that "o" has the properties S, Q, and T. We know that any object that has the property S also has the property V. Moreover, an object cannot have the properties S and Q at the same time, nor the properties V and T at the same time.* **Question:** *Can we say that the object "o" has the property T?*

Let us also provide here the logical representation of the above text. Please note that the participants have not also received the logical transcript.

- $\mathcal{F} = \{s(o), q(o), t(o)\}$

- $\mathcal{N} = \{\forall X\ s(X) \wedge q(X) \rightarrow \bot, \forall X\ v(X) \wedge t(X) \rightarrow \bot\}$

- $\mathcal{R} = \{\forall X\ s(X) \rightarrow v(X)\}$

- *Query* $Q = t(o)$

Participants were shown in a random order 5 situations containing inconsistencies. For each situation, the participant was presented with a textual description of an inconsistent knowledge base and a query. Possible answers for a query is "Yes" (entailed) or "No" (not entailed). The 41 participants are second year university students in computer science, 12 female and 29 male aged between 17 and 46 years old.

Table 1 presents the situations and the semantics under which their queries are entailed ($\checkmark$) or not entailed ($-$). The "% of Yes" column indicates the percentage of participants that answered "Yes". The aim of each situation is to identify if a set of semantics coincides with the majority, for example the query in Situation 1 (Example 3) is only entailed under $\models_{block}$ [2]. Not all cases can be represented, for example $\models_{IAR} f$ and $\nvDash_{prop} f$, due to *productivity* (c.f. Section 4.2).

Table 1: Situations Entailment and Results.

Situations	$\models_{block}$	$\models_{prop}$	$\models_{IAR}$	$\models_{ICAR}$	% of "Yes"	$\models_{IAR}^{block}$
#1	$\checkmark$	-	-	-	73.17%	$\checkmark$
#2	$\checkmark$	$\checkmark$	-	-	21.95%	-
#3	$\checkmark$	$\checkmark$	-	$\checkmark$	21.95%	-
#4	-	-	-	$\checkmark$	4.87%	-
#5	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	78.04%	$\checkmark$

From the results in Table 1, we observe that blocking and IAR are the most intuitive (Situations 1 and 5), however blocking alone is not sufficient as shown by Situations

---

[2]Situations and detailed results are available at https://www.dropbox.com/s/4wkblgdx7hzj7s8/situations.pdf

2 and 3, and IAR alone is not sufficient either (Situation 1). One possible explanation is that participants are using a semantics that is a mix of IAR and ambiguity blocking ($\models_{IAR}^{block}$). Such a semantics is absent from the literature as it is interestingly in between Repair Semantics and Defeasible Logics. Please note that we do not argue that this particular semantics is better than existing ones, our aim is to bridge Defeasible Logics and Repair Semantics by defining new inference relations based on intuitions coming from both of them, and then studying properties of these new inference relations.

5.1 New semantics for reasoning in presence of conflict

The intuition behind this new semantics is to apply IAR or ICAR on an SG which has been labeled using BDL rather than PDL. This would amount to replacing PDL by BDL in the definitions of IAR and ICAR. As it turns out, IAR with BDL fully coincides with the answers given by the majority of the participants in our experiment. To illustrate this semantics, consider Example 4.

Example 4 *Applying IAR with ambiguity blocking on Example 1's SG gives $\mathcal{KB} \models_{IAR}^{block}$ $female(alice) \wedge alibi(alice) \wedge innocent(alice)$. Note that the difference with BDL is that $\mathcal{KB} \not\models_{IAR}^{block} incrim(e1, alice)$ and $\mathcal{KB} \not\models_{IAR}^{block} absolv(e2, alice)$. The difference with IAR is that $\mathcal{KB} \not\models_{IAR} alibi(alice)$ and $\mathcal{KB} \not\models_{IAR} innocent(alice)$.*

Let us now analyse the productivity and complexity of new semantics. We say that a semantics $\models_1$ is less productive than $\models_2$ (represented as $\models_1 \rightarrow \models_2$) if, for every $\mathcal{KB}$ and every f, it results in fewer conclusions being drawn (i.e. if $\mathcal{KB} \models_1 f$ then $\mathcal{KB} \models_2 f$). Productivity comparison of Repair Semantics has been discussed in [6] while the productivity between Defeasible Logics semantics can be extracted from the inclusion theorem in [8]. It can be seen that $\models_{IAR} \rightarrow \models_{prop}$ since $\models_{prop}$ only rejects facts that are challenged or generated from challenged facts, while $\models_{IAR}$ also rejects facts that would lead to a conflict.

Proposition 3 *Let $\mathcal{KB}$ be a knowledge base with only defeasible facts and strict rules. Given a fact f:*

1. *if $\mathcal{KB} \models_{IAR} f$ then $\mathcal{KB} \models_{prop} f$*

2. *if $\mathcal{KB} \models_{IAR} f$ then $\mathcal{KB} \models_{IAR}^{block} f$*

3. *if $\mathcal{KB} \models_{IAR}^{block} f$ then $\mathcal{KB} \models_{block} f$*

4. *if $\mathcal{KB} \models_{IAR}^{block} f$ then $\mathcal{KB} \models_{ICAR}^{block} f$*

5. *if $\mathcal{KB} \models_{ICAR} f$ then $\mathcal{KB} \models_{ICAR}^{block} f$*

We prove (1.) by contradiction. Suppose there is a fact f such that $\mathcal{KB} \models_{IAR} f$ and $\mathcal{KB} \not\models_{prop} f$. $\mathcal{KB} \models_{IAR} f$ means that there is a derivation for f from an initial set of facts $T \subseteq \mathcal{F}$ and there is no consistent set of initial facts $S \subseteq \mathcal{F}$ such that $S \cup T$ is inconsistent (i.e $models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$ and $models(S \cup T, \mathcal{R} \cup \mathcal{N}) = \emptyset$) [16]. This means that f is derivable and does not rely conflicting facts, therefore the

statement $(f \to \emptyset)$ has a complete IN support and no IN or AMBIG attack edges, i.e. $\mathbb{SG}_{\mathcal{KB}}^{\text{PDL}}\langle(f,\emptyset)\rangle = \text{IN}$, thus $\mathcal{KB} \models_{prop} f$ which is a contradiction.

We prove (2.) by contradiction, suppose we have $\mathcal{KB} \models_{IAR} f$ and $\mathcal{KB} \not\models_{IAR}^{block} f$:

1. $\mathcal{KB} \models_{IAR} f$ means that there is a derivation for f from an initial set of facts $T \subseteq \mathcal{F}$ and there is no consistent set of initial facts $S \subseteq \mathcal{F}$ such that $S \cup T$ is inconsistent (i.e $models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$ and $models(S \cup T, \mathcal{R} \cup \mathcal{N}) = \emptyset$), which means that f is not generated by conflicting atoms and is not used to generate conflicting atoms i.e. $\text{PDL}(f, \emptyset) = \text{IN}$ which implies that $\text{BDL}(f, \emptyset) = \text{IN}$ [8].

2. $\mathcal{KB} \not\models_{IAR}^{block} f$ means that either $\text{BDL}(f, \emptyset) \neq \text{IN}$ (impossible given 1.) or f is used to generate conflicting atoms (impossible given 1.)

We prove (3.) by contradiction, suppose we have $\mathcal{KB} \models_{IAR}^{block} f$ and $\mathcal{KB} \not\models_{block} f$:

1. $\mathcal{KB} \models_{IAR}^{block} f$ means that $\text{BDL}(f, \emptyset) = \text{IN}$ and f is not used to generate conflicting atoms.

2. $\mathcal{KB} \not\models_{block} f$ means that $\text{BDL}(f, \emptyset) \neq \text{IN}$ (impossible given 1.).

We prove (4.) by contradiction, suppose we have $\mathcal{KB} \models_{IAR}^{block} f$ and $\mathcal{KB} \not\models_{ICAR}^{block} f$:

1. $\mathcal{KB} \models_{IAR}^{block} f$ means that $\text{BDL}(f, \emptyset) = \text{IN}$ and f is not used to generate conflicting atoms.

2. $\mathcal{KB} \not\models_{ICAR}^{block} f$ means that either $\text{BDL}(f, \emptyset) \neq \text{IN}$ (impossible given 1.) or there is a derivable atom conflicting with f or f is used to generate conflicting atoms (impossible given 1.)

We prove (5.) by contradiction, suppose we have $\mathcal{KB} \models_{ICAR} f$ and $\mathcal{KB} \not\models_{ICAR}^{block} f$:

1. $\mathcal{KB} \models_{ICAR} f$ means that $\text{PDL}((f \to \emptyset)) \in \{\text{IN}, \text{AMBIG}\}$, there is no derivable fact conflicting with f, and f is not used to derive conflicting atoms.

2. $\mathcal{KB} \not\models_{ICAR}^{block} f$ means that $\text{BDL}((f \to \emptyset)) \neq \text{IN}$ and $\text{BDL}((f \to \emptyset)) = \text{AMBIG}$ and either there is a derivable fact that is conflicting with f (impossible given 1.) or f is used to generate conflicting atoms (impossible given 1.).
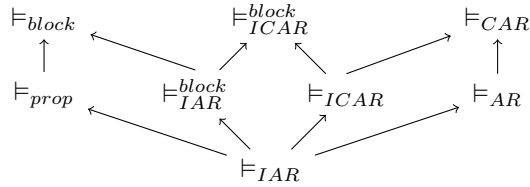

Figure 4: Productivity and complexity of different semantics under Skolem-FES fragment of existential rules.

6   Discussion

In this paper we build upon Statement Graphs for existential rules and their labeling functions for ambiguity blocking, ambiguity propagating [13], and provide custom labeling functions for IAR, and ICAR. These labelings explicitly show how to transition from ambiguity propagating to IAR and ICAR, and how to obtain a combination of blocking with IAR and ICAR. Using an experiment, we have shown that bringing together Defeasible reasoning and Repair semantics allows for the definition of new and potentially interesting semantics with respect to human reasoning. Implementing these new labelings, for instance in the platform presented in [14], would allow to study further the links between labeling functions and human reasoning.

The modeling choices used in this paper (the use of forward chaining, specific Repair Semantics, particular intuitions of Defeasible Logics, and no account for preferences) stem from several rationale. More precisely:

- *Skolem chase*: the focus on the forward chaining mechanism is due to its ability to handle transitive rules [19] contrary to backward chaining [3]. Regarding the choice of chase, we focused on the Skolem chase given its relatively low cost and its ability to stay decidable for all known concrete classes of the FES fragment [5].

- *Language*: Repair Semantics make the assumption of a coherent set of rules because incoherence might yield to the trivial solution of an empty set of repairs [10]. Therefore allowing defeasible rules with the restriction of coherence defeats the purpose of having defeasible rules in the first place.

- *Considered Repair Semantics*: the appeal of IAR and ICAR is in their simplicity and low complexity. Considering other Repair Semantics such as ICR, AR, etc. would require using and defining a more complex version of SG and defeasible reasoning such as well founded semantics which is one favored future research avenue.

- *Defeasible reasoning intuitions*: ambiguity handling is, of course, not the only intuition in defeasible reasoning, however other intuitions such as team defeat, handling of strict rules, etc. are meaningless in this context given the absence of preferences and defeasible rules. However, floating conclusions are applicable in the considered language, nevertheless, neither Defeasible Logics nor IAR/ ICAR accept floating conclusion.

Acknowledgement

References

[1] Antoniou, G., Billington, D., Governatori, G., Maher, M.J., Rock, A.: A Family of Defeasible Reasoning Logics and its Implementation. In: Proceedings of the 14th European Conference on Artificial Intelligence. pp. 459–463 (2000)

[2] Bacchus, F., Chen, X., van Beek, P., Walsh, T.: Binary vs. non-binary constraints. Artif. Intell. **140**(1/2), 1–37 (2002). https://doi.org/10.1016/S0004-3702(02)00210-2, https://doi.org/10.1016/S0004-3702(02)00210-2

[3] Baget, J.F., Garreau, F., Mugnier, M.L., Rocher, S.: Extending Acyclicity Notions for Existential Rules. In: ECAI. pp. 39–44 (2014)

[4] Baget, J.F., Garreau, F., Mugnier, M.L., Rocher, S.: Revisiting chase termination for existential rules and their extension to nonmonotonic negation. arXiv preprint arXiv:1405.1071 (2014)

[5] Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artificial Intelligence **175**(9-10), 1620–1654 (2011)

[6] Benferhat, S., Bouraoui, Z., Croitoru, M., Papini, O., Tabia, K.: Non-Objection Inference for Inconsistency-Tolerant Query Answering. In: IJCAI. pp. 3684–3690 (2016)

[7] Billington, D.: Defeasible Logic is Stable. Journal of logic and computation **3**(4), 379–400 (1993)

[8] Billington, D., Antoniou, G., Governatori, G., Maher, M.: An inclusion theorem for defeasible logics. ACM Transactions on Computational Logic (TOCL) **12**(1), 6 (2010)

[9] Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. Web Semantics: Science, Services and Agents on the World Wide Web **14**, 57–83 (2012)

[10] Deagustini, C.A., Martinez, M.V., Falappa, M.A., Simari, G.R.: On the Influence of Incoherence in Inconsistency-tolerant Semantics for Datalog+-. In: JOWO@ IJCAI (2015)

[11] Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proceedings of the National Conference on Artificial Intelligence. vol. 21, p. 1295. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2006)

[12] Governatori, G., Maher, M.J., Antoniou, G., Billington, D.: Argumentation Semantics for Defeasible Logic. Journal of Logic and Computation **14**(5), 675–702 (2004). https://doi.org/10.1093/logcom/14.5.675.1

[13] Hecham, A., Bisquert, P., Croitoru, M.: On a Flexible Representation of Defeasible Reasoning Variants. In: Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems. pp. 1123–1131 (2018)

[14] Hecham, A., Croitoru, M., Bisquert, P.: DAMN: Defeasible Reasoning Tool for Multi-Agent Reasoning. In: AAAI 2020 - 34th AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence, New York, United States (Feb 2020), `https://hal-lirmm.ccsd.cnrs.fr/lirmm-02393877`

[15] Horty, J.F., Thomason, R.H., Touretzky, D.S.: A skeptical theory of inheritance in nonmonotonic semantic networks. Artificial intelligence **42**(2-3), 311–348 (1990)

[16] Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: International Conference on Web Reasoning and Rule Systems. pp. 103–117. Springer (2010)

[17] Marnette, B.: Generalized schema-mappings: from termination to tractability. In: Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 13–22. ACM (2009)

[18] Nute, D.: Defeasible reasoning: a philosophical analysis in prolog. Springer (1988)

[19] Rocher, S.: Querying Existential Rule Knowledge Bases: Decidability and Complexity. Ph.D. thesis, Université de Montpellier (2016)