Defeasible RDFS via rational closure

Giovanni Casini [a,b,*], Umberto Straccia [a]

[a] *CNR - ISTI, Pisa, Italy*
[b] *CAIR - University of Cape Town, Cape Town, South Africa*

ARTICLE INFO

ABSTRACT

In the field of non-monotonic logics, the notion of Rational Closure (RC) is acknowledged as a notable approach. In recent years, RC has gained popularity in the context of Description Logics (DLs), the logic underpinning the standard semantic Web Ontology Language OWL 2, whose main ingredients are classes, the relationship among classes and roles, which are used to describe the properties of classes.

In this work, we show instead how to integrate RC within the triple language RDFS (Resource Description Framework Schema), which together with OWL 2 is a major standard semantic web ontology language.

To do so, we start from ρdf, a minimal, but significant RDFS fragment that covers the essential features of RDFS, and then extend it to $\rho df_\perp$, allowing to state that two entities are incompatible/disjoint with each other. Eventually, we propose defeasible $\rho df_\perp$ via a typical RC construction allowing to state default class/property inclusions.

Furthermore, to overcome the main weaknesses of RC in our context, *i.e.*, the "drowning problem" (*viz.* the "inheritance blocking problem"), we further extend our construction by leveraging Defeasible Inheritance Networks (DIN) defining a new non-monotonic inference relation that combines the advantages of both (RC and DIN). To the best of our knowledge this is the first time of such an attempt.

In summary, the main features of our approach are: (i) the defeasible $\rho df_\perp$ we propose here remains syntactically a triple language by extending it with new predicate symbols with specific semantics; (ii) the logic is defined in such a way that any RDFS reasoner/store may handle the new predicates as ordinary terms if it does not want to take account of the extra non-monotonic capabilities; (iii) the defeasible entailment decision procedure is built on top of the $\rho df_\perp$ entailment decision procedure, which in turn is an extension of the one for ρdf via some additional inference rules favouring a potential implementation; (iv) the computational complexity of deciding entailment in ρdf and $\rho df_\perp$ are the same; and (v) defeasible entailment can be decided via a polynomial number of calls to an oracle deciding ground triple entailment in $\rho df_\perp$ and, in particular, deciding defeasible entailment can be done in polynomial time.

* Corresponding author at: CNR - ISTI, Pisa, Italy.
  E-mail addresses: giovanni.casini@isti.cnr.it (G. Casini), umberto.straccia@isti.cnr.it (U. Straccia).

1. Introduction

Description Logics (DLs) [4] under *Rational Closure* (RC) [35] is a well-known framework for non-monotonic reasoning in DLs, which has gained rising attention in the last decade [7,10,12–16,22,25,26,41,43].

We recall that a typical problem that can be addressed using non-monotonic formalisms is reasoning with ontologies in which some classes are exceptional w.r.t. some properties of their super classes, as illustrated with the following example.

Example 1.1. Consider the following facts.

(1) Young people are usually happy.
(2) Young people are usually students.
(3) Drug users are usually unhappy.
(4) Drug users use drugs.
(5) Drugs are chemical substances.
(6) Drug users are usually young.
(7) Controlled drug users are usually happy.

We may consider then reasonable to conclude:

(8) Young drug users are usually unhappy.
(9) Controlled young drug users are usually happy.
(10) Drug users are usually students. □

While DLs provide the logical foundation of formal ontologies of the semantic *Web Ontology Language* (OWL) family[1] and endowing them with non-monotonic features is still a main issue, as documented by the past 20 years of technical development [8,19,20,24,38], addressing non-monotonicity in the context of the triple language RDFS (*Resource Description Framework Schema*),[2] which together with OWL 2 is a major standard semantic web ontology language, has attracted in comparison little attention so far. Moreover, almost all approaches we are aware of consider a so-called rule-layer on top of RDFS; see *e.g.*, [2,3,19,31] and Section 6.

In this paper, we will show instead how to integrate RC directly within the triple language RDFS. To do so, we start from ρdf [27,39], a minimal, but significant RDFS fragment that covers the essential features of RDFS, and then extend it to $\rho df_\perp$, allowing to state that two entities are incompatible/disjoint with each other. So, for instance, by referring to Example 1.1, we may represent that happy beings (h) and unhappy beings (u) are incompatible with each other via the $\rho df_\perp$ triple $(h, \perp_c, u)$ (see also Example 3.2 later on), where $\perp_c$ is a new predicate added to the ρdf vocabulary to model an incompatibility relation. Roughly, you may think of being a triple $(c, \perp_c, d)$ as a $\rho df_\perp$ counterpart of a disjointness axiom in OWL (*viz.* DLs) of the form $\texttt{DisjointClasses}(C, D)$ (in DL syntax, $C \sqcap D \sqsubseteq \perp$).

Remark 1.1. Note that, I'm 'unhappy' tells exactly what you are, while I'm 'not happy' only tells what you are not, and, thus, I'm 'unhappy' does not mean the same as I'm 'not happy'. Nevertheless, both are incompatible with the state of being 'happy'. Of course, the incompatibility/disjointness relation is weaker than the negation/complementing relation in the sense that the latter implies the former, but not vice-versa. We introduce the notion of incompatibility/disjointness as the 'weakest' relation we are aware of to address defeasible non-monotonic reasoning, that requires a notion of conflict between pieces of information.[3] This, however, does not prevent our approach to be extended with stronger notions as well, such as involving forms of 'negation' (see, *e.g.*, [2,3,19,31]). However, these have been come so far at the price of an increase of the computational complexity (see Section 6). We will not address such extensions here and leave them for future work. We also refer the reader to [16] for a similar argument related to the use of $\perp$ to express disjointness among DL classes to implement a computationally tractable RC for the DL $\mathcal{ELO}_\perp$.

Now, based on $\rho df_\perp$, we will then propose defeasible $\rho df_\perp$ via a typical RC construction, allowing to state, *e.g.*, "young people (y) are usually happy" via the defeasible triple $\langle y, \text{sc}, h \rangle$, alongside classical triples such as (du, use, d) ("drug users (du) use drugs (d)") and (d, sc, c) ("drugs are chemical substances (c)").[4]

However, despite RC having nice behavioural properties [35], the main weaknesses of RC in our context is the so-called "drowning problem" (also known as the "inheritance blocking problem"), which essentially states that if some defeasible property of a class C contradicts any defeasible property of a superclass D of C, then C inherits none of the defeasible properties of D, including those that are consistent with the properties of C. So, for example, again by referring to Example 1.1, since drug users are atypical young people

---

[1] https://www.w3.org/TR/owl2-profiles/.

[2] http://www.w3.org/TR/rdf-schema/.

[3] Informally, defeasible rules allow for the presence of exceptional cases, that have properties in conflict with respect to the typical situation (*e.g.*, in Example 1.1 we say that drug users are usually unhappy (3), but controlled drug users, despite being drug users, are usually happy (7)).

[4] We recall that according to ρdf, sc stands for "is subclass of".

(they are unhappy, while young people are usually happy), RC wouldn't allow us to associate with drug users any typical property of young people, for example, to conclude $\langle du, \mathsf{sc}, s \rangle$ ("drug users (du) are usually students (s)"). To overcome the "drowning problem", we further extend our construction by leveraging *Defeasible Inheritance Networks* (DIN) [30] defining a new non-monotonic inference relation combining the advantages of both (RC and DIN) and, thus, *e.g.*, allowing us eventually to conclude (10) in Example 1.1.

In summary, the main features of our approach are:

- defeasible $\rho df_\perp$ remains syntactically a triple language and is a simple extension of the ρdf vocabulary by introducing some new predicate symbols with specific semantics;
- any RDFS reasoner/store and SPARQL[5] query answering tool may handle the new types of triples as ordinary ones if it does not want to take account of the extra non-monotonic capabilities;
- the defeasible entailment decision procedure is built on top of the $\rho df_\perp$ entailment decision procedure, which in turn is an extension of the one for ρdf via some additional inference rules favouring a potential implementation; and
- the computational complexity of deciding entailment in ρdf and $\rho df_\perp$ are the same; and
- defeasible entailment can be decided via a polynomial number of calls to an oracle deciding ground triple entailment in $\rho df_\perp$. In particular, deciding defeasible entailment can be done in polynomial time.

In the following, we will proceed as follows. In the next section, we will introduce $\rho df_\perp$, by defining its syntax, semantics and entailment decision procedure. In Section 3 we extend $\rho df_\perp$ towards defeasible $\rho df_\perp$ via an RC construction, defining syntax, semantics, entailment decision procedure and address its computational complexity. In Section 4, we encapsulate DINs into our framework, define the entailment decision procedure and address its computational complexity. All the previous sections address the case of ground graphs only. The non-ground case is addressed in Section 5. Eventually, Section 6 addresses related work and Section 7 concludes with a brief summary of our contribution and highlights future work. Some poofs of lemmas and theorems are in the appendixes.

2. $\rho df_\perp$ graphs

RC is a popular non-monotonic approach in conditional reasoning: given a set of defeasible conditionals $\alpha \rightsquigarrow \beta$ (if α holds, then presumably β holds), we use them like classical monotonic conditionals until we have a conflict in our information, that triggers the non-monotonic reasoning machinery. Example 1.1 shows such a case: we have a conflict between classical reasoning, suggesting that drug users, being young, should be happy, and an exceptional more specific information, stating that drug users are unhappy. Faced with such a conflict, the non-monotonic engine solves it giving, in this case, precedence to the more specific information ("drug users are unhappy"). However, in order to implement this kind of defeasible reasoning in the RDFS framework, we need to introduce some notion of informational incompatibility, such as "someone can not be happy and unhappy at the same time". Such incompatibilities cannot be represented in ρdf: to do so, we extend the ρdf language to $\rho df_\perp$ allowing to express such forms of conflicts (but, see also Remark 1.1). We are going to briefly introduce the ρdf formalism, in order to introduce the $\rho df_\perp$ language. We are going to characterise $\rho df_\perp$ by appropriately extending the ρdf semantics and the ρdf deductive system.

2.1. Syntax

To start with, we rely on a fragment of RDFS, called *minimal ρdf* [39, Def. 15], that covers all main features of RDFS, is essentially the formal logic behind RDFS and, in fact, suffices to illustrate the main concepts and algorithms we will consider in this work and ease the presentation. Specifically, minimal ρdf has the following *vocabulary* (reserved keywords/predicates) whose meaning will be clear within few paragraphs:

$$\rho df = \{ \mathsf{sp}, \mathsf{sc}, \mathsf{type}, \mathsf{dom}, \mathsf{range} \} \ . \tag{1}$$

Moreover, we recall that minimal ρdf does not consider so-called *blank* nodes (see, *e.g.*, [29,39] for more on this theme) and, thus, in what follows, triples and graphs will be *ground*. We will come back to deal with non-ground graphs in Section 5 later on. Also, to avoid unnecessary redundancy, we will just drop the term 'minimal' in what follows.

So, consider pairwise disjoint alphabets **U** and **L** denoting, respectively, *URI references* and *literals*. With **UL** we will denote the union of these sets. Moreover, a *vocabulary* is a subset of **UL**. Specifically, we assume that **U** contains the ρdf vocabulary (see Equation (1)). A literal may be a *plain literal* (*e.g.*, a string) or a *typed literal* (*e.g.*, a boolean value).[6] We call the elements in **UL** *terms*. Terms are denoted with lower case letters $a, b, \dots$ with optional super/lower script.

A *triple* is of the form $\tau = (s, p, o) \in \mathbf{UL} \times \mathbf{U} \times \mathbf{UL}$,[7] where $s, o \notin \rho df$. We call s the *subject*, p the *predicate*, and o the *object*. A *graph* G is a set of triples, the *universe* of G, denoted $\mathsf{uni}(G)$, is the set of terms in **UL** that occur in the triples of G.

We recall that informally (i) (p, sp, q) means that property p is a *subproperty* of property q; (ii) (c, sc, d) means that class c is a *subclass* of class d; (iii) (a, type, b) means that a is of *type* b; (iv) (p, dom, c) means that the *domain* of property p is c; and (v) (p, range, c) means that the *range* of property p is c.
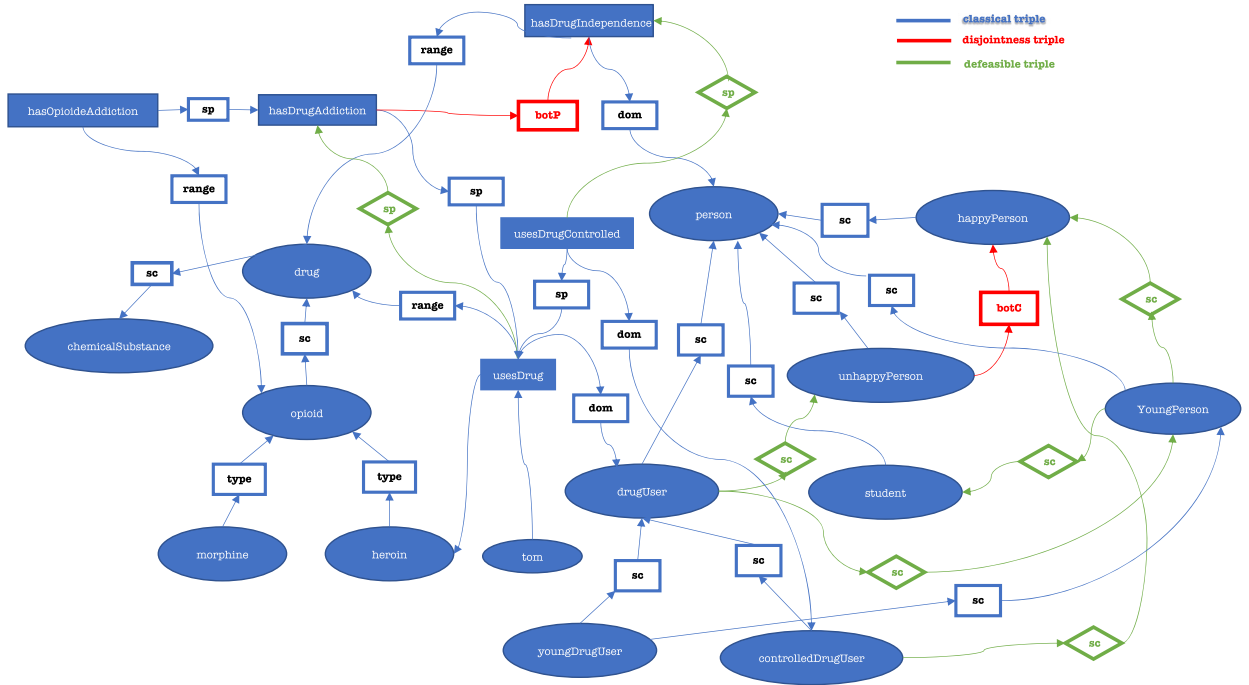

Fig. 1. The drug ontology graph. Rectangular/diamond nodes are properties/relations. Vocabulary properties are in bold. A triple has the pattern $\star \to \boxdot \to \bullet$, where $\bullet$ is an oval node, $\boxdot$ a rectangular or diamond node, and $\star$ is either an oval or rectangular node. Neither $\star$ nor $\bullet$ can involve vocabulary properties. Blue triples are classical ρdf triples, red triples are disjointness triples, while green ones are defeasible.

Example 2.1 *(Running example)*. The following graph is an excerpt of a drug ontology (see also Fig. 1 for a pictorial representation, neglecting for the moment the non-blue elements).

$$G = \{\ (morphine, \mathsf{type}, opioid), (heroin, \mathsf{type}, opioid), (opioid, \mathsf{sc}, drug),$$

$$(drug, \mathsf{sc}, chemicalSubstance), (usesDrug, \mathsf{dom}, drugUser), (usesDrug, \mathsf{range}, drug),$$

$$(tom, usesDrug, heroin), (hasDrugIndependence, \mathsf{dom}, person),$$

$$(hasDrugIndependence, \mathsf{range}, drug), (hasOpioidAddiction, \mathsf{sp}, hasDrugAddiction),$$

$$(hasDrugAddiction, \mathsf{sp}, usesDrug), (happyPerson, \mathsf{sc}, person), (unhappyPerson, \mathsf{sc}, person),$$

$$(hasOpioidAddiction, \mathsf{range}, opioid), (drugUser, \mathsf{sc}, person), (uses, \mathsf{dom}, person),$$

$$(controlledDrugUser, \mathsf{sc}, drugUser), (student, \mathsf{sc}, person), (youngPerson, \mathsf{sc}, person),$$

$$(youngDrugUser, \mathsf{sc}, drugUser), (youngDrugUser, \mathsf{sc}, youngPerson),$$

$$(usesDrugControlled, \mathsf{sp}, usesDrug), (usesDrugControlled, \mathsf{dom}, controlledDrugUser)\} \quad \square$$

Remark 2.1. Let us recap that ρdf has been introduced in [39] to cover the salient features of RDFS from an inference point of view. In [39, Table A.1] the whole RDFS vocabulary is enumerated.[8] There you may find, *e.g.*, the keyword **rdfs:Class**, to indicate that a given term is a class. Besides, in full RDFS one has also to consider many axioms such as $(\mathsf{sc}, \mathsf{dom}, class)$ indicating that "the first argument of the property sc is a class" (see [39, Table A.2]).[9] We believe that, as in [39], such ingredients are of limited importance from a theoretical point of view. Extending our framework to whole RDFS is somewhat straightforward and tedious and, thus, we will not address it here.

We extend the vocabulary of ρdf with a new pair of predicates, $\bot_c$ and $\bot_p$, representing incompatible information:

- $(c, \bot_c, d)$ indicates that the classes c and d are disjoint; analogously,
- $(p, \bot_p, q)$ indicates that the properties p and q are disjoint.

---

[8] See also http://www.w3.org/TR/rdf-schema/.
[9] Instead, we will have no axioms in our setting, see also [39, Corollary 14].

We call $\rho df_\perp$ the vocabulary obtained from ρdf by adding $\perp_c$ and $\perp_p$, that is,

$$\rho df_\perp = \{\mathsf{sp}, \mathsf{sc}, \mathsf{type}, \mathsf{dom}, \mathsf{range}, \perp_c, \perp_p\} . \tag{2}$$

Like for ρdf, we assume that $\mathbf{U}$ contains the $\rho df_\perp$ vocabulary and that all triples $(s, p, o) \in \mathbf{UL} \times \mathbf{U} \times \mathbf{UL}$ are such that $s, o \notin \rho df_\perp$.

Remark 2.2. Please, note that we allow the $\rho df_\perp$ predicates to occur only as second elements of the triples, that is, we allow triples (p, sp, q), but not triples such as *e.g.*, (sp, p, o) or $(\perp_p, p, o)$, which is in line with the notion of *minimal ρdf triple* [39, Def. 15].

Example 2.2 *(Example 2.1 cont.)*. By referring to Example 2.1, the graph

$$G^{str} = G \ \cup \{ (unhappyPerson, \perp_c, happyPerson),$$

$$(hasDrugAddiction, \perp_p, hasDrugIndependence) \ \}$$

is a $\rho df_\perp$ graph. $\square$

2.2. Semantics

Our semantics for $\rho df_\perp$ extends the one for ρdf [39] in the following way.

An *interpretation* $\mathcal{I}$ over a vocabulary V is a tuple $\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$, where $\Delta_R, \Delta_P, \Delta_C, \Delta_L$ are the interpretation domains of $\mathcal{I}$, which are finite non-empty sets, and $\mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}}$ are the interpretation functions of $\mathcal{I}$. In particular:

1. Δ_R are the resources (the domain or universe of $\mathcal{I}$);
2. Δ_P are property names (not necessarily disjoint from Δ_R);
3. $\Delta_C \subseteq \Delta_R$ are the classes;
4. $\Delta_L \subseteq \Delta_R$ are the literal values and contains $\mathbf{L} \cap V$;
5. $\mathfrak{P}[\![\cdot]\!]$ is a function $\mathfrak{P}[\![\cdot]\!] : \Delta_P \to 2^{\Delta_R \times \Delta_R}$;
6. $\mathfrak{C}[\![\cdot]\!]$ is a function $\mathfrak{C}[\![\cdot]\!] : \Delta_C \to 2^{\Delta_R}$;
7. $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^{\mathcal{I}} \in \Delta_R \cup \Delta_P$, and such that $\cdot^{\mathcal{I}}$ is the identity for plain literals and assigns an element in Δ_R to each element in $\mathbf{L}$.

As next, we are going to build our semantics starting from the so-called *reflexive-relaxed* ρdf semantics [39], in which the predicates sc and sp are *not* assumed to be reflexive in accordance to the notion of minimal graph [39, Def. 15]. Our additional semantic constraints w.r.t. [39, Def. 15] are those of condition **Disjointness I** onwards.[10]

Definition 2.1 *(Satisfaction)*. An interpretation $\mathcal{I}$ is a *model* of a graph G, denoted $\mathcal{I} \Vdash_{\rho df_\perp} G$, if and only if $\mathcal{I}$ is an interpretation over the vocabulary $\rho df_\perp \cup \mathtt{uni}(G)$ that satisfies the following conditions:

Simple:
    1. for each $(s, p, o) \in G$, $p^{\mathcal{I}} \in \Delta_P$ and $(s^{\mathcal{I}}, o^{\mathcal{I}}) \in \mathfrak{P}[\![p^{\mathcal{I}}]\!]$;
Subproperty:
    1. $\mathfrak{P}[\![\mathsf{sp}^{\mathcal{I}}]\!]$ is transitive over Δ_P;
    2. if $(p, q) \in \mathfrak{P}[\![\mathsf{sp}^{\mathcal{I}}]\!]$ then $p, q \in \Delta_P$ and $\mathfrak{P}[\![p]\!] \subseteq \mathfrak{P}[\![q]\!]$;
Subclass:
    1. $\mathfrak{P}[\![\mathsf{sc}^{\mathcal{I}}]\!]$ is transitive over Δ_C;
    2. if $(c, d) \in \mathfrak{P}[\![\mathsf{sc}^{\mathcal{I}}]\!]$ then $c, d \in \Delta_C$ and $\mathfrak{C}[\![c]\!] \subseteq \mathfrak{C}[\![d]\!]$;
Typing I:
    1. $x \in \mathfrak{C}[\![c]\!]$ if and only if $(x, c) \in \mathfrak{P}[\![\mathsf{type}^{\mathcal{I}}]\!]$;
    2. if $(p, c) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}}]\!]$ and $(x, y) \in \mathfrak{P}[\![p]\!]$ then $x \in \mathfrak{C}[\![c]\!]$;
    3. if $(p, c) \in \mathfrak{P}[\![\mathsf{range}^{\mathcal{I}}]\!]$ and $(x, y) \in \mathfrak{P}[\![p]\!]$ then $y \in \mathfrak{C}[\![c]\!]$;
Typing II:
    1. For each $\mathsf{e} \in \rho df_\perp$, $\mathsf{e}^{\mathcal{I}} \in \Delta_P$;
    2. if $(p, c) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}}]\!]$ then $p \in \Delta_P$ and $c \in \Delta_C$;
    3. if $(p, c) \in \mathfrak{P}[\![\mathsf{range}^{\mathcal{I}}]\!]$ then $p \in \Delta_P$ and $c \in \Delta_C$;
    4. if $(x, c) \in \mathfrak{P}[\![\mathsf{type}^{\mathcal{I}}]\!]$ then $c \in \Delta_C$;
Disjointness I:
    1. if $(c, d) \in \mathfrak{P}[\![\perp_c^{\mathcal{I}}]\!]$ then $c, d \in \Delta_C$;
    2. $\mathfrak{P}[\![\perp_c^{\mathcal{I}}]\!]$ is symmetric, *sc*-transitive and *c*-exhaustive over Δ_C (see below);

---

[10] An interesting property of *reflexive-relaxed* ρdf is that the proof system is axiom-free [39, Corollary 14].

3. if $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$ then $p, q \in \Delta_\mathsf{P}$;

4. $\mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$ is symmetric, *sp*-transitive and *p*-exhaustive over Δ_P (see below);

Disjointness II:

1. If $(p,c) \in \mathfrak{P}[\![\mathsf{dom}^\mathcal{I}]\!]$, $(q,d) \in \mathfrak{P}[\![\mathsf{dom}^\mathcal{I}]\!]$, and $(c,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$, then $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$;

2. If $(p,c) \in \mathfrak{P}[\![\mathsf{range}^\mathcal{I}]\!]$, $(q,d) \in \mathfrak{P}[\![\mathsf{range}^\mathcal{I}]\!]$, and $(c,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$, then $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$;

Symmetry:

1. If $(c,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$, then $(d,c) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$;

2. If $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$, then $(q,p) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$;

sc-Transitivity:

1. If $(c,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ and $(e,c) \in \mathfrak{P}[\![\mathsf{sc}^\mathcal{I}]\!]$, then $(e,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$;

sp-Transitivity:

1. If $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$ and $(r,p) \in \mathfrak{P}[\![\mathsf{sp}^\mathcal{I}]\!]$, then $(r,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$;

c-Exhaustive:

1. If $(c,c) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ and $d \in \Delta_\mathsf{C}$ then $(c,d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$;

p-Exhaustive:

1. If $(p,p) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$ and $q \in \Delta_\mathsf{P}$ then $(p,q) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^\mathcal{I}]\!]$.

A graph G is *satisfiable* if it has a model $\mathcal{I}$ (denoted $\mathcal{I} \Vdash_{\rho df_\perp} G$).

Remark 2.3. Note that the presence of *e.g.*, $(a, \mathsf{type}, b), (a, \mathsf{type}, c)$ and $(b, \perp_\mathsf{c}, c)$ in a graph does not preclude its satisfiability. In fact, a graph is always satisfiable (see Corollary 2.5 later on) avoiding, thus, the possibility of unsatisfiability and the *ex falso quodlibet* principle. This is in line with the ρdf semantics [39].

On top of the notion of satisfaction we define a notion of entailment between graphs.

Definition 2.2 (*Entailment $\vDash_{\rho df_\perp}$*). Given two graphs G and H, we say that G entails H, denoted $G \vDash_{\rho df_\perp} H$, if and only if every model of G is also a model of H.[11]

Example 2.3 (*Example 2.2 cont.*). Consider Example 2.2. It can be verified that, *e.g.*, the following hold:

$$G^{str} \vDash (tom, \mathsf{type}, person)$$

$$G^{str} \vDash (hasDrugIndependence, \perp_\mathsf{p}, hasOpioidAddiction) \ . \quad \square$$

2.3. Deductive system

In what follows, we provide a sound and complete deductive system for our language. Our system extends the classical *minimal ρdf* system as by Muñoz and others [39, Proposition 17]. The rules for ρdf correspond to rules (1) - (4) below. The system is arranged in groups of rules that capture the semantic conditions of models. In every rule, $A, B, C, D, E, X,$ and Y are meta-variables representing elements in **UL**. An instantiation of a rule is obtained by replacing those meta-variables with actual terms. The rules are as follows:

1. Simple:

   $\frac{G}{G'}$ for $G' \subseteq G$

2. Subproperty:

   (a) $\quad \frac{(A,\mathsf{sp},B),(B,\mathsf{sp},C)}{(A,\mathsf{sp},C)} \qquad$ (b) $\quad \frac{(D,\mathsf{sp},E),(X,D,Y)}{(X,E,Y)}$

3. Subclass:

   (a) $\quad \frac{(A,\mathsf{sc},B),(B,\mathsf{sc},C)}{(A,\mathsf{sc},C)} \qquad$ (b) $\quad \frac{(A,\mathsf{sc},B),(X,\mathsf{type},A)}{(X,\mathsf{type},B)}$

4. Typing:

   (a) $\quad \frac{(D,\mathsf{dom},B),(X,D,Y)}{(X,\mathsf{type},B)} \qquad$ (b) $\quad \frac{(D,\mathsf{range},B),(X,D,Y)}{(Y,\mathsf{type},B)}$

5. Conceptual Disjointness:

   (a) $\quad \frac{(A,\perp_\mathsf{c},B)}{(B,\perp_\mathsf{c},A)} \qquad$ (b) $\quad \frac{(A,\perp_\mathsf{c},B),(C,\mathsf{sc},A)}{(C,\perp_\mathsf{c},B)} \qquad$ (c) $\quad \frac{(A,\perp_\mathsf{c},A)}{(A,\perp_\mathsf{c},B)}$

6. Predicate Disjointness:

   (a) $\quad \frac{(A,\perp_\mathsf{p},B)}{(B,\perp_\mathsf{p},A)} \qquad$ (b) $\quad \frac{(A,\perp_\mathsf{p},B),(C,\mathsf{sp},A)}{(C,\perp_\mathsf{p},B)} \qquad$ (c) $\quad \frac{(A,\perp_\mathsf{p},A)}{(A,\perp_\mathsf{p},B)}$

---

[11] For ease of presentation, if a graph is a singleton, we will omit the curly braces.

7. Crossed Disjointness:

$$(a) \quad \frac{(A, \mathsf{dom}, C), (B, \mathsf{dom}, D), (C, \perp_{\mathsf{c}}, D)}{(A, \perp_{\mathsf{p}}, B)} \qquad (b) \quad \frac{(A, \mathsf{range}, C), (B, \mathsf{range}, D), (C, \perp_{\mathsf{c}}, D)}{(A, \perp_{\mathsf{p}}, B)}$$

As anticipated, please note that the rules that extend minimal ρdf to $\rho df_\perp$ are rules (5) - (7).

Now, using these rules we define a derivation relation in a similar way as in [39].

Definition 2.3 *(Derivation $\vdash_{\rho df_\perp}$).* Let G and H be $\rho df_\perp$-graphs. $G \vdash_{\rho df_\perp} H$ iff there exists a sequence of graphs $P_1, P_2, \ldots, P_k$ with $P_1 = G$ and $P_k = H$ and for each j ($2 \leqslant j \leqslant k$) one of the following cases holds:

- $P_j \subseteq P_{j-1}$ (rule (1));
- there is an instantiation R/R' of one of the rules (2)-(7), such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.

Such a sequence of graphs is called a proof of $G \vdash_{\rho df_\perp} H$. Whenever $G \vdash_{\rho df_\perp} H$, we say that the graph H is derived from the graph G. Each pair (P_{j-1}, P_j), $1 \leqslant j \leqslant k$ is called a step of the proof which is labelled by the respective instantiation R/R' of the rule applied at the step.

Example 2.4 *(Example 2.3 cont.).* Consider the entailments in Example 2.3. The following are the proofs of them.

Case $G^{str} \vdash_{\rho df_\perp} (tom, \mathsf{type}, person)$.

$$(4a) \frac{(usesDrug, \mathsf{dom}, drugUser) \qquad (tom, usesDrug, heroin)}{(3b) \frac{(tom, \mathsf{type}, drugUser) \qquad (drugUser, \mathsf{sc}, person)}{(tom, \mathsf{type}, person)}}$$

Case $G^{str} \vdash_{\rho df_\perp} (hasDrugIndependence, \perp_{\mathsf{p}}, hasOpioidAddiction)$.

$$(6b) \frac{(hasOpioidAddiction, \mathsf{sp}, hasAddiction) \qquad (hasAddiction, \perp_{\mathsf{p}}, hasDrugIndependence)}{(5a) \frac{(hasOpioidAddiction, \perp_{\mathsf{p}}, hasDrugIndependence)}{(hasDrugIndependence, \perp_{\mathsf{p}}, hasOpioidAddiction)}}$$

□

We are going now to prove the following soundness and completeness theorem (recall that proofs can be found in the appendixes).

Theorem 2.1 *(Soundness & completeness).* Let G and H be $\rho df_\perp$-graphs.

$$G \vdash_{\rho df_\perp} H \text{ iff } G \vDash_{\rho df_\perp} H .$$

We divide the proof into lemmas. The following one is needed for soundness.

Lemma 2.2. *Let G and H be $\rho df_\perp$-graphs, let G be satisfiable, and let one of the following statements hold:*

- *$H \subseteq G$;*
- *there is an instantiation R/R' of one of the rules (2)-(7), such that $R \subseteq G$ and $H = G \cup R'$.*

Then, $G \vDash_{\rho df_\perp} H$.

The following lemma defines the construction of the *canonical model* for $\rho df_\perp$ graphs. Let $\mathsf{Cl}(G)$ be the closure of G under the application of rules $(2) - (7)$.

Lemma 2.3. *Given a $\rho df_\perp$-graph G, define an interpretation $\mathcal{I}_G$ as a tuple*

$$\mathcal{I}_G = \langle \Delta_\mathsf{R}, \Delta_\mathsf{P}, \Delta_\mathsf{C}, \Delta_\mathsf{L}, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}_G} \rangle$$

such that:

1. $\Delta_\mathsf{R} := \mathtt{uni}(G) \cup \rho df_\perp$;
2. $\Delta_\mathsf{P} := \{p \in \mathtt{uni}(G) \mid (s, p, o) \in \mathsf{Cl}(G)\} \cup \rho df_\perp \cup \{p \in \mathtt{uni}(G) \mid either\ (p, \mathsf{sp}, q),\ (q, \mathsf{sp}, p),\ (p, \mathsf{dom}, c),\ (p, \mathsf{range}, d),\ (p, \perp_\mathsf{p}, q)\ or\ (q, \perp_\mathsf{p}, p) \in \mathsf{Cl}(G)\}$;

3. $\Delta_{\mathsf{C}} := \{c \in \mathrm{uni}(G) \mid (x, \mathrm{type}, c) \in \mathrm{Cl}(G)\} \cup \{c \in \mathrm{uni}(G) \mid \textit{either } (c, \mathrm{sc}, d), (d, \mathrm{sc}, c), (p, \mathrm{dom}, c), (p, \mathrm{range}, c), (c, \perp_{\mathsf{c}}, d) \textit{ or } (d, \perp_{\mathsf{c}}, c) \in \mathrm{Cl}(G)\}$;

4. $\Delta_{\mathsf{L}} := \mathrm{uni}(G) \cap \mathbf{L}$;

5. $\mathfrak{P}[\![\cdot]\!]$ *is an extension function* $\mathfrak{P}[\![\cdot]\!] : \Delta_{\mathsf{P}} \to 2^{\Delta_{\mathsf{R}} \times \Delta_{\mathsf{R}}}$ *s.t.* $\mathfrak{P}[\![p]\!] := \{(s, o) \mid (s, p, o) \in \mathrm{Cl}(G)\}$;

6. $\mathfrak{C}[\![\cdot]\!]$ *is an extension function* $\mathfrak{C}[\![\cdot]\!] : \Delta_{\mathsf{C}} \to 2^{\Delta_{\mathsf{R}}}$ *s.t.* $\mathfrak{C}[\![c]\!] := \{x \in \mathrm{uni}(G) \mid (x, \mathrm{type}, c) \in \mathrm{Cl}(G)\}$;

7. $\cdot^{\mathcal{I}_G}$ *is an identity function over* Δ_{R}.

Then, for every $\rho df_\perp$*-graph* G, $\mathcal{I}_G \Vdash_{\rho df_\perp} G$.

Now, we have:

Lemma 2.4. *Let* G *and* H *be* $\rho df_\perp$*-graphs. If* $G \vDash_{\rho df_\perp} H$ *then* $H \subseteq \mathrm{Cl}(G)$.

Finally, we can prove the main theorem of this section.

Proof of Theorem 2.1. The proof mirrors the proof of Theorem 8 in [39]. From Lemma 2.4, $G \vDash_{\rho df_\perp} H$ implies that H can be obtained from $\mathrm{Cl}(G)$ using rule (1). Thus, since $G \vdash_{\rho df_\perp} \mathrm{Cl}(G)$, it follows that $G \vdash_{\rho df_\perp} H$. Therefore Theorem 2.1 follows from Lemmas 2.2 and 2.4. □

Please note that, like in classical ρdf, $\rho df_\perp$-graphs are always satisfiable.

Corollary 2.5. *A* $\rho df_\perp$*-graph* G *is always satisfiable.*

Proof. This is an immediate consequence of Lemma 2.3. □

Example 2.5. Consider the simple graph $G = \{(a, \mathrm{type}, c), (a, \mathrm{type}, d), (c, \perp_{\mathsf{c}}, d)\}$. The interpretation $\mathcal{I}_G$ is defined as follows:

- $\Delta_{\mathsf{R}} = \{a, c, d\} \cup \rho df_\perp$;
- $\Delta_{\mathsf{P}} = \{\mathrm{type}, \perp_{\mathsf{c}}\}$; $\Delta_{\mathsf{C}} = \{c, d\}$; $\Delta_{\mathsf{L}} = \emptyset$;
- $\mathfrak{P}[\![\mathrm{type}]\!] = \{\langle a, c \rangle, \langle a, d \rangle\}$; $\mathfrak{P}[\![\perp_{\mathsf{c}}]\!] = \{\langle c, d \rangle, \langle d, c \rangle\}$;
- $\mathfrak{C}[\![c]\!] = \{a\}$; $\mathfrak{C}[\![d]\!] = \{a\}$;
- $x^{\mathcal{I}_G} = x$, for every $x \in \Delta_{\mathsf{R}}$. □

2.4. Some interesting derived inference rules

In what follows, we illustrate the derivation of some interesting rules of inference. To start with, note that the triples $(d, \perp_{\mathsf{c}}, d)$ and $(q, \perp_{\mathsf{p}}, q)$ are particularly significant: indeed, the intended meaning of *e.g.*, $(d, \perp_{\mathsf{c}}, d)$ is that 'concept/class d is empty'.

Some derived inference rules that will turn out to be useful are the following:

Empty Subclass:

$$\frac{(A, \mathrm{sc}, B) \qquad (A, \mathrm{sc}, C) \qquad (B, \perp_{\mathsf{c}}, C)}{(A, \perp_{\mathsf{c}}, A)} \text{ (EmptySC)}$$

Here is the derivation:

$$\text{(5b)} \ \frac{\dfrac{(A, \mathrm{sc}, B) \qquad (B, \perp_{\mathsf{c}}, C)}{\text{(5a)} \ \dfrac{(A, \perp_{\mathsf{c}}, C)}{(C, \perp_{\mathsf{c}}, A)} \qquad (A, \mathrm{sc}, C)}}{(A, \perp_{\mathsf{c}}, A)} \text{ (5b)}$$

A special case of rule (EmptySC) is obtained by imposing $B = C$, shows that if a class is empty, also all its subclasses are empty.

$$\frac{(A, \mathrm{sc}, B) \qquad (B, \perp_{\mathsf{c}}, B)}{(A, \perp_{\mathsf{c}}, A)} \text{ (EmptySC')}$$

Empty Subpredicate:

$$\frac{(A, \mathrm{sp}, B) \qquad (A, \mathrm{sp}, C) \qquad (B, \perp_{\mathsf{p}}, C)}{(A, \perp_{\mathsf{p}}, A)} \text{ (EmptySP)}$$

Here is the derivation:

$$(6b) \frac{(A, \mathsf{sp}, B) \qquad (B, \perp_\mathsf{p}, C)}{(6a) \frac{(A, \perp_\mathsf{p}, C)}{\frac{(C, \perp_\mathsf{p}, A)}{(A, \perp_\mathsf{p}, A)} \qquad (A, \mathsf{sp}, C)} (6b)}$$

As for (EmptySC), by imposing $B = C$ we obtain a special case of the rule (EmptySP) showing that if a property is empty, so are all its subproperties.

$$\frac{(A, \mathsf{sp}, B) \qquad (B, \perp_\mathsf{p}, B)}{(A, \perp_\mathsf{p}, A)} \text{ (EmptySP')}$$

Conflicting Domain:

$$\frac{(A, \mathsf{dom}, C) \qquad (A, \mathsf{dom}, X) \qquad (C, \perp_\mathsf{c}, X)}{(A, \perp_\mathsf{p}, A)} \text{ (7a')}$$

This is a special case of the rule (7a) in which $B = A$.
Conflicting Range:

$$\frac{(A, \mathsf{range}, C) \qquad (A, \mathsf{range}, X) \qquad (C, \perp_\mathsf{c}, X)}{(A, \perp_\mathsf{p}, A)} \text{ (7b')}$$

Similarly as above, this is a special case of the rule (7b) in which $B = A$.

3. Defeasible $\rho df_\perp$

We now introduce the possibility of modelling defeasible information in the RDFS framework.

3.1. Syntax

We will consider defeasibility w.r.t. the predicates sc and sp only, and introduce the notion of *defeasible triple* defined next.

Definition 3.1 (*Defeasible triple*). A *defeasible triple* is of the form

$$\delta = \langle s, p, o \rangle \in \mathbf{UL} \times \{\mathsf{sc}, \mathsf{sp}\} \times \mathbf{UL} \,,$$

where $s, o \notin \rho df_\perp$.

The intended meaning of *e.g.*, $\langle c, \mathsf{sc}, d \rangle$ is "Typically, an instance of c is also an instance of b". So, for instance, by referring to Example 1.1, the defeasible statement (3) can be represented as the defeasible triple $\langle drugUser, \mathsf{sc}, unhappyPerson \rangle$. Analogously, $\langle p, \mathsf{sp}, q \rangle$ is read as "Typically, a pair related by p is also related by q". So, for instance, by referring to Fig. 1, the defeasible triple $\langle usesDrug, \mathsf{sp}, hasDrugAddiction \rangle$ aims at representing the fact that "typically, if someone uses some drug then it may become addicted to that drug".

A *defeasible graph* is a set $G = G^{str} \cup G^{def}$, where G^{str} is a $\rho df_\perp$-graph and G^{def} is a set of defeasible triples.

Example 3.1 (*Running example cont.*). The following set G^{def} is a set of defeasible triples (*cf.* green triples within Fig. 1)

$$G^{def} = \{ \langle drugUser, \mathsf{sc}, unhappyPerson \rangle, \langle drugUser, \mathsf{sc}, youngPerson \rangle,$$

$$\langle youngPerson, \mathsf{sc}, student \rangle, \langle controlledDrugUser, \mathsf{sc}, happyPerson \rangle,$$

$$\langle youngPerson, \mathsf{sc}, happyPerson \rangle, \langle usesDrug, \mathsf{sp}, hasDrugAddiction \rangle,$$

$$\langle usesDrugControlled, \mathsf{sp}, hasDrugIndependence \rangle \} \,.$$

Therefore, the whole graph in Fig. 1 corresponds to the defeasible graph

$$G = G^{str} \cup G^{def} \,,$$

where G^{str} is defined in Example 2.2. □

Given two defeasible graphs G and G', G is a sub-graph of G' iff $G \subseteq G'$. In the following, we use the notation $[\![\cdot,\cdot,\cdot]\!]$ to indicate either $(\cdot,\cdot,\cdot)$ or $\langle\cdot,\cdot,\cdot\rangle$ (that is, $[\![\cdot,\cdot,\cdot]\!] \in \{(\cdot,\cdot,\cdot),\langle\cdot,\cdot,\cdot\rangle\}$).

Remark 3.1. Note that in practice a defeasible triple $\langle c, \mathsf{sc}, d\rangle$ may be represented as (c, sc_t, d), where sc_t is a new symbol indicating defeasible class inclusion. Similarly, $\langle p, \mathsf{sp}, q\rangle$ may be represented as (p, sp_t, q), where sp_t is a new symbol indicating defeasible property inclusion. Therefore, both defeasible triples could have been represented in the RDF language with vocabulary $\rho df_\perp \cup \{\mathsf{sc}_t, \mathsf{sp}_t\}$. While certainly this is an option for a practical implementation, for ease of presentation, we prefer to stick to the former notation.

Given a defeasible graph $G = G^{str} \cup G^{def}$, its *strict counterpart* is the graph

$$G^s := G^{str} \cup \{(s,p,o) \mid \langle s,p,o\rangle \in G^{def}\} . \tag{3}$$

Generally speaking, given a defeasible graph G, we need to define how to reason with it. As mentioned above, in presumptive reasoning it is considered rational to reason classically with defeasible information in case no conflicts arise; on the other hand, if we have to deal with conflictual information we need to introduce some form of defeasible reasoning to resolve such conflicts.

To do so, first of all, we need to define the notion of *conflict* in our framework.

Definition 3.2 (conflict). Let G be a defeasible graph. G has a *conflict* if, for some term t, either $G^s \vdash_{\rho df_\perp} (t, \perp_{\mathsf{c}}, t)$ or $G^s \vdash_{\rho df_\perp} (t, \perp_{\mathsf{p}}, t)$ holds.

Informally, we consider that there is a conflict in a defeasible graph if, treating the defeasible triples as classical triples, we have that some term t must be interpreted as being 'empty'.[12]

To give a sense of the rationale behind this definition, let's see how it behaves w.r.t. our running example.

Example 3.2 (Running example cont.). Consider the defeasible graph G defined in Example 3.1. Let us prove that G has a conflict. To do so, note that the strict counterpart of G is

$$G^s = G^{str} \cup \{(drugUser, \mathsf{sc}, unhappyPerson), (drugUser, \mathsf{sc}, youngPerson),$$

$$(youngPerson, \mathsf{sc}, student), (controlledDrugUser, \mathsf{sc}, happyPerson),$$

$$(youngPerson, \mathsf{sc}, happyPerson), (usesDrug, \mathsf{sp}, hasDrugAddiction),$$

$$(usesDrugControlled, \mathsf{sp}, hasDrugIndependence)\} .$$

Now, let us prove that $G^s \vdash_{\rho df_\perp} (drugUser, \perp_{\mathsf{c}}, drugUser)$, *i.e.*, from a "strict" point of view, a $drugUser$ cannot exist. For ease of presentation, we will use the following abbreviations:

$$unhappyPerson \mapsto uhP$$
$$happyPerson \mapsto hP$$
$$drugUser \mapsto dU$$
$$youngPerson \mapsto yP .$$

Now, by rule (5a) applied to $(uhP, \perp_{\mathsf{c}}, hP)$ we immediately have $G^s \vdash_{\rho df_\perp} (hP, \perp_{\mathsf{c}}, uhP)$. Therefore, we have the following proof of $G^s \vdash_{\rho df_\perp} (dU, \perp_{\mathsf{c}}, dU)$:

$$(3a) \quad \frac{\dfrac{(dU, \mathsf{sc}, yP) \qquad (yP, \mathsf{sc}, hP)}{(dU, \mathsf{sc}, hP)} \qquad (dU, \mathsf{sc}, uhP) \qquad (hP, \perp_{\mathsf{c}}, uhP)}{(dU, \perp_{\mathsf{c}}, dU)} \text{ (EmptySC)}$$

As a consequence, G has a conflict. Note that in a similar way we may prove that

$$G^s \vdash_{\rho df_\perp} (usesDrugControlled, \perp_{\mathsf{p}}, usesDrugControlled) . \quad \square$$

3.2. Semantics

An interpretation for a defeasible graph G is composed by a set of $\rho df_\perp$ interpretations, ranked accordingly to how much they conform to our expectations.

---

[12] Note that the concept of conflict that we define here is related to the notion of *incoherence* in OWL formalism [44]: an OWL ontology is incoherent if a concept introduced in the vocabulary turns out to be empty.

Table 1

A ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$, as described in Example 3.3. $\mathcal{M}$ is the set of all the $\rho df_\perp$-interpretations for the given vocabulary. $[\![(b, \perp_c, b) \cup (s, \text{sc}, b)]\!]$ indicates all the $\rho df_\perp$-interpretations in $\mathcal{M}$ that satisfy both $(b, \perp_c, b)$ and (s, sc, b), $[\![(b, \text{sc}, f) \cup (s, \text{sc}, b)]\!]$ all the $\rho df_\perp$-interpretations that satisfy both (b, sc, f) and (s, sc, b), and so on.

rank ∞	$\mathcal{M}^F \setminus (\text{rank } 0 \cup \text{rank } 1 \cup \text{rank } 2)$
rank 2	$[\![(s, \text{sc}, b)]\!] \setminus (\text{rank } 0 \cup \text{rank } 1)$
rank 1	$[\![(b, \text{sc}, f) \cup (s, \text{sc}, b)]\!] \setminus \text{rank } 0$
rank 0	$[\![(b, \perp_c, b) \cup (s, \text{sc}, b)]\!]$

Definition 3.3 (*Ranked $\rho df_\perp$ interpretations*). A ranked interpretation is a pair $\mathcal{R} = (\mathcal{M}, r)$, where $\mathcal{M}$ is the set of all $\rho df_\perp$ interpretations defined on a fixed set of domains $\Delta_R, \Delta_P, \Delta_C, \Delta_L$, and r is a ranking function over $\mathcal{M}$[13]

$$r : \mathcal{M} \mapsto \mathbb{N} \cup \{\infty\}$$

satisfying a convexity property:

- there is an interpretation $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$;
- for each $i > 0$, if there is an interpretation $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = i$, then there is an interpretation $\mathcal{I}' \in \mathcal{M}$ s.t. $r(\mathcal{I}') = (i - 1)$.

Informally, the intuition behind ranking interpretations is that, given two $\rho df_\perp$ interpretations $\mathcal{I}, \mathcal{I}' \in \mathcal{M}$, $r(\mathcal{I}) < r(\mathcal{I}')$ indicates that the interpretation $\mathcal{I}$ is more in line with our expectations than the interpretation $\mathcal{I}'$. Given $\mathcal{R} = (\mathcal{M}, r)$, let $\mathcal{M}_\mathbb{N}$ be the set of elements in $\mathcal{M}$ with rank lower that ∞, that is,

$$\mathcal{M}_\mathbb{N} = \{\mathcal{I} \in \mathcal{M} \mid r(\mathcal{I}) \in \mathbb{N}\} \ .$$

Informally, in $\mathcal{R}$ the $\rho df_\perp$-interpretations with rank infinity are simply considered impossible, and the satisfaction relation, which we will define next, is determined referring only to the $\rho df_\perp$-interpretations in $\mathcal{M}_\mathbb{N}$.

Example 3.3. Consider a vocabulary that allows to talk about birds (b), sparrows (s), and their flying abilities (f). Table 1 represents a ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ in which rank 0 is populated with the $\rho df_\perp$-interpretations that satisfy $(b, \perp_c, b)$ and (s, sc, b); rank 1 with the $\rho df_\perp$-interpretations that satisfy (b, sc, f) and (s, sc, b) and that are not already in rank 0; rank 2 with the $\rho df_\perp$-interpretations that satisfy (s, sc, b) and that are not already in the lower ranks, 0 and 1; finally, rank ∞ is populated by the $\rho df_\perp$-interpretations that do not satisfy (s, sc, b).

Hence, $\mathcal{R}$ represents the following expectations. In all the conceivable situations (that is, in $\mathcal{M}_\mathbb{N}$, from rank 0 to rank 2), sparrows are birds, since (s, sc, b) is satisfied. In the most expected situations, that is, rank 0, birds do not actually exist ($(b, \perp_c, b)$ holds). Then, in rank 1, there are the situations in which birds exist, and they fly, since (b, sc, f) is satisfied. Rank 2 represents the situations in which birds exist, but they do not necessarily fly ((b, sc, f) is not satisfied); such situations are conceivable, but less expected. Eventually we have the situations that are considered impossible, represented by rank ∞, in which sparrows are not necessarily birds ((s, sc, b) does not hold).

Given a ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ and a term t, let $\text{c\_min}(t, \mathcal{R})$ be the set of the most expected interpretations in $\mathcal{M}$ in which t (interpreted as class) is not empty, that is,

$$\text{c\_min}(t, \mathcal{R}) = \{\mathcal{I} \in \mathcal{M}_\mathbb{N} \mid \mathcal{I} \not\Vdash_{\rho df_\perp} (t, \perp_c, t) \text{ and there is no } \mathcal{I}' \in \mathcal{M}_\mathbb{N} \text{ s.t.}$$
$$\mathcal{I}' \not\Vdash_{\rho df_\perp} (t, \perp_c, t) \text{ and } r(\mathcal{I}') < r(\mathcal{I})\} \ .$$

Analogously, for a term t interpreted as predicate, we define

$$\text{p\_min}(t, \mathcal{R}) = \{\mathcal{I} \in \mathcal{M}_\mathbb{N} \mid \mathcal{I} \not\Vdash_{\rho df_\perp} (t, \perp_p, t) \text{ and there is no } \mathcal{I}' \in \mathcal{M}_\mathbb{N} \text{ s.t.}$$
$$\mathcal{I}' \not\Vdash_{\rho df_\perp} (t, \perp_p, t) \text{ and } r(\mathcal{I}') < r(\mathcal{I})\} \ .$$

Definition 3.4 (*Ranked satisfaction*). For every triple (s, p, o), a ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ *satisfies* (s, p, o) if (s, p, o) is satisfied by every $\rho df_\perp$-interpretation in $\mathcal{M}$, that is,

$$\mathcal{R} \Vdash_{\rho df_\perp} (s, p, o) \text{ iff } \mathcal{I} \Vdash_{\rho df_\perp} (s, p, o) \text{ for every } \mathcal{I} \in \mathcal{M}_\mathbb{N} \ .$$

---

[13] We will assume that $0 \in \mathbb{N}$.

For every defeasible triple of the form $\langle s, p, o \rangle$, the notion of a ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ *satisfying* $\langle s, p, o \rangle$, denoted $\mathcal{R} \Vdash_{\rho df_\perp} \langle s, p, o \rangle$, is defined as follows:

$$\mathcal{R} \Vdash_{\rho df_\perp} \langle c, \mathsf{sc}, d \rangle \text{ iff } \mathcal{I} \Vdash_{\rho df_\perp} (c, \mathsf{sc}, d) \text{ for every } \mathcal{I} \in \mathsf{c\_min}(c, \mathcal{R})$$

$$\mathcal{R} \Vdash_{\rho df_\perp} \langle p, \mathsf{sp}, q \rangle \text{ iff } \mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sp}, q) \text{ for every } \mathcal{I} \in \mathsf{p\_min}(p, \mathcal{R}) \,.$$

Given a defeasible $\rho df_\perp$-graph $G = G^{str} \cup G^{def}$, a ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ is a *model* of G (denoted $\mathcal{R} \Vdash_{\rho df_\perp} G$) if $\mathcal{R} \Vdash_{\rho df_\perp} [\![s, p, o]\!]$ for every $[\![s, p, o]\!] \in G$.

The intuition behind this definition is that the triple $\langle c, \mathsf{sc}, s \rangle$ holds in a ranked interpretation if (c, sc, s) holds in the most expected $\rho df_\perp$-interpretations in which c is not an empty class. The intuition follows a similar line of the original propositional construction [35], and its DL reformulations [10,12,26].

Example 3.4. Consider the ranked interpretation $\mathcal{R} = (\mathcal{M}, r)$ in Example 3.3. $\mathcal{M}_\mathbb{N}$ corresponds to all the $\rho df_\perp$-interpretations in ranks from 0 to 2. Hence we have

$$\mathcal{R} \Vdash_{\rho df_\perp} (s, \mathsf{sc}, b),$$

since all the $\rho df_\perp$-interpretations in $\mathcal{M}_\mathbb{N}$ satisfy (s, sc, b). That is, according to $\mathcal{R}$ sparrows are a subclass of birds. Moreover, we also have

$$\mathcal{R} \Vdash_{\rho df_\perp} \langle b, \mathsf{sc}, f \rangle.$$

That is, typical birds fly. To see this, we have to look at the most expected $\rho df_\perp$-interpretations in which birds exist, that is, $\mathsf{c\_min}(b, \mathcal{R})$. Now, $\mathsf{c\_min}(b, \mathcal{R})$ corresponds to the $\rho df_\perp$-interpretations in rank 1, since in rank 0 all the interpretations satisfy $(b, \perp_c, b)$. As all the $\rho df_\perp$-interpretations in rank 1 satisfy (b, sc, f), we have that $\mathcal{R}$ satisfies $\langle b, \mathsf{sc}, f \rangle$.

As in the propositional and DL constructions, once we have defined the notion of ranked interpretation to model defeasible information, the problem is to decide which kind of entailment relation, that is, what kind of defeasible reasoning, we would like to model. Despite it is recognised that there are multiple available options according to the kind of properties we want to satisfy [14,15,26,33–35,40], it is generally recognised that RC [35] is the fundamental construction in the area, and most of the other proposed systems can be built as refinements of it. We recall that RC models the so-called *Presumption of Typicality* [34, p. 4], that is the reasoning principle imposing that, if we are not informed about any exceptional property, we presume that we are dealing with a typical situation. The essential behaviour characterising the presumption of typicality is that a subclass that does not show any exceptional property inherits all the typical properties of the superclass. From a semantic point of view the definition of RC can be obtained via various equivalent definitions [28,35]: here we opt for the characterisation of RC using the minimal ranked model [9,40] and, in particular, we consider the characterisation given in [26], which, once adjusted accordingly, we believe appropriate for our defeasible RDF framework.

In the following, given a defeasible graph $G = G^{str} \cup G^{def}$ and its strict counterpart G^s (see Equation (3)), let the interpretation $\mathcal{I}_{G^s}$

$$\mathcal{I}_{G^s} := \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}_{G^s}} \rangle \tag{4}$$

be defined from G^s as in Lemma 2.3, and consider the domains $\Delta_R, \Delta_P, \Delta_C, \Delta_L$ in it. With $\mathcal{M}^G$ we denote the set of all $\rho df_\perp$-interpretations defined over such domains, and we indicate a ranked interpretation built over $\mathcal{M}^G$ as $\mathcal{R}^G = (\mathcal{M}^G, r)$. If clear from the context, we may omit the superscript G in $\mathcal{R}^G = (\mathcal{M}^G, r)$.[14]

Now, given a defeasible graph G, let $\mathfrak{I}_G$ be the set of the ranked interpretations $\mathcal{R}^G$ and let $\mathfrak{R}_G$ be the elements of $\mathfrak{I}_G$ that are also models for G, that is,

$$\mathfrak{R}_G := \{ \mathcal{R} \in \mathfrak{I}_G \mid \mathcal{R} \Vdash_{\rho df_\perp} G \} \,. \tag{5}$$

Note that $\mathfrak{R}_G$ is not empty, since for any defeasible graph G it is easy to define a trivial ranked model: using the construction in Lemma 2.3 we define the $\rho df_\perp$-interpretation $\mathcal{I}_{G^s}$, that is a model of G^s, the strict counterpart of G (see above); consider the ranked interpretation $\mathcal{R}^{tr} = (\mathcal{M}^G, r)$, where $r(\mathcal{I}_{G^s}) = 0$ and $r(\mathcal{I}) = \infty$ for any other $\mathcal{I} \in \mathcal{M}^G$. It is easy to check that $\mathcal{R}^{tr}$ is a model of the defeasible graph G.

Moreover, since the domains in Equation (4) are finite, $\mathcal{M}^G$ is finite as well. Additionally, as a ranked interpretation $\mathcal{R}^G = (\mathcal{M}^G, r)$ built over $\mathcal{M}^G$ has to satisfy the convexity property (see Definition 3.3), the ranking function r is bounded by $|\mathcal{M}^G|$.[15] Therefore, $\mathfrak{I}_G$ is finite and, thus, so is $\mathfrak{R}_G$.

Please note that by construction $\mathcal{R}, \mathcal{R}' \in \mathfrak{R}_G$ differ only w.r.t. the involved ranking functions r, r', respectively, which induces the following order over the ranked models in $\mathfrak{R}_G$.

---

[14] Please note that the domains of all interpretations occurring in $\mathcal{R}^G$ are defined as in Equation (4).

[15] That is, the maximal rank of any interpretation in $\mathcal{M}^G$ cannot exceed $|\mathcal{M}^G|$.

Definition 3.5 (*Presumption ordering $\preceq$*). Let $\mathcal{R} = (\mathcal{M}, r)$, $\mathcal{R}' = (\mathcal{M}, r')$, and $\mathcal{R}, \mathcal{R}' \in \mathfrak{R}_G$. We define

1. $\mathcal{R} \preceq \mathcal{R}'$ iff for every $\mathcal{I} \in \mathcal{M}$, $r(\mathcal{I}) \leqslant r'(\mathcal{I})$.
2. $\mathcal{R} \prec \mathcal{R}'$ iff $\mathcal{R} \preceq \mathcal{R}'$ and $\mathcal{R}' \npreceq \mathcal{R}$.
3. $\min_{\preceq}(\mathfrak{R}_G) := \{\mathcal{R} \in \mathfrak{R}_G \mid \text{there is no } \mathcal{R}' \in \mathfrak{R}_G \text{ s.t. } \mathcal{R}' \prec R\}$.

The set $\min_{\preceq}(\mathfrak{R}_G)$ contains the ranked models of G in which the $\rho df_\perp$-interpretations are "pushed down" as much as possible in the ranking, that is, they are considered as typical as possible.

We next show that actually there is a unique minimal ranked model for a defeasible graph.

Proposition 3.1. *For every defeasible graph G, $|\min_{\preceq}(\mathfrak{R}_G)| = 1$.*

In the following, the unique $\preceq$-minimal ranked model in $\mathfrak{R}_G$ is called the *minimal G-model* and is denoted with $\mathcal{R}_{\min G}$, *i.e.*, $\min_{\preceq}(\mathfrak{R}_G) = \{\mathcal{R}_{\min G}\}$.

Hence, Definition 3.5 and Proposition 3.1 together tell us that the only minimal G-model $\mathcal{R}_{\min G}$ of a graph G is the ranked model of G in which every $\rho df_\perp$-interpretation is positioned in the lowest possible rank, modulo the satisfaction of the graph G. That is, it is the model of G in which we consider every $\rho df_\perp$-interpretation as much typical as possible (that is, in the lowest possible rank), given the information at our disposal (the graph G).

Definition 3.6 (*Minimal entailment*). Given a defeasible graph G and the corresponding minimal G-model $\mathcal{R}_{\min G}$ of G. A defeasible graph G *minimally entails* a triple $[\![s, p, o]\!]$, denoted $G \vDash_{\min} [\![s, p, o]\!]$, iff $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} [\![s, p, o]\!]$.

Using the minimal ranking in $\mathcal{R}_{\min G}$ we can also define the *height* of a term, indicating at which level of exceptionality a term t is not necessarily empty, as a class or as a predicate. This corresponds to the minimal rank, that is, the rank in $\mathcal{R}_{\min G}$, in which we encounter a $\rho df_\perp$-interpretation that does not satisfy $(t, \perp_c, t)$, or, respectively, $(t, \perp_p, t)$.

Definition 3.7 (*Height*). Let $G = G^{str} \cup G^{def}$ be a defeasible graph, with $\mathcal{R}_{\min G} = \{\mathcal{M}, r\}$ being its minimal model, and let t be a term in G. The c-*height* of t, indicated as $h_G^c(t)$, corresponds to the lowest rank $r(\mathcal{I})$ of some $\mathcal{I} \in \mathcal{M}$ s.t. $\mathcal{I} \nVdash_{\rho df_\perp} (t, \perp_c, t)$, that is,

$$h_G^c(t) = \begin{cases} \infty, & \text{if } \mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_c, t) \text{ for every } \mathcal{I} \in \mathcal{M}_{\mathbb{N}} \\ \min\{r(\mathcal{I}) \mid \mathcal{I} \in \mathcal{M} \text{ and } \mathcal{I} \nVdash_{\rho df_\perp} (t, \perp_c, t)\}, & \text{otherwise.} \end{cases}$$

Analogously, the p-*height* of t, indicated as $h_G^p(t)$, corresponds to the lowest rank $r(\mathcal{I})$ of some $\mathcal{I} \in \mathcal{M}$ s.t. $\mathcal{I} \nVdash_{\rho df_\perp} (t, \perp_p, t)$

$$h_G^p(t) = \begin{cases} \infty, & \text{if } \mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_p, t) \text{ for every } \mathcal{I} \in \mathcal{M}_{\mathbb{N}} \\ \min\{r(\mathcal{I}) \mid \mathcal{I} \in \mathcal{M} \text{ and } \mathcal{I} \nVdash_{\rho df_\perp} (t, \perp_p, t)\}, & \text{otherwise.} \end{cases}$$

Also, we define the *height $h(\mathcal{R})$* of a ranked interpretation $\mathcal{R}$ as the highest finite rank of the $\rho df_\perp$-interpretations in it. That is, let $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model, then

$$h(\mathcal{R}) := \max\{r(\mathcal{I}) \mid \mathcal{I} \in \mathcal{M}_{\mathbb{N}}\} .$$

The c-*height* of a class c indicates how exceptional the objects in it are w.r.t. a graph G. We consider the minimal model $\mathfrak{R}_G$ of the graph G, in which every $\rho df_\perp$-interpretation is associated to the lowest possible rank. If there is some interpretation $\mathcal{I}$ in rank 0 such that $\mathcal{I} \nVdash_{\rho df_\perp} (c, \perp_c, c)$, then it is reasonable to expect a situation in which c is a non-empty class. In such a case $h_G^c(c) = 0$, that indicates that c is a class that is compatible with all our expectations (represented by the defeasible part of the graph G). If, instead, the first $\rho df_\perp$-interpretations not satisfying $(c, \perp_c, c)$ appear higher in the ranks, then the class c is exceptional, that is, it can be populated only by objects that do not fully satisfy our expectations. The higher the height, the more exceptional the population of c must be w.r.t. the expectations formalised by the graph G. Analogously for the p-*height* of a predicate p. The following example uses the classical simple penguin scenario to show that the c-*height* of the class of penguins is higher than the c-*height* of the class of birds, since penguins are exceptional non-flying birds.

Example 3.5 (*Penguin example*). Consider the typical *penguin example* from the non-monotonic reasoning literature. That is, we have a set F^{str} of classical triples and the set F^{def} containing defeasible triples:

$$F^{str} = \{(p, \mathsf{sc}, b), (s, \mathsf{sc}, b), (p, \mathsf{sc}, e), (e, \perp_c, f)\} \text{ and } F^{def} = \{\langle b, \mathsf{sc}, f \rangle\} ,$$

where p, b, f, e, s stand for *penguins, birds, flying creatures, non-flying creatures, sparrow*, respectively. Together they define the graph F:

$$F = F^{str} \cup F^{def} = \{(p, \mathsf{sc}, b), (s, \mathsf{sc}, b), (p, \mathsf{sc}, e), (e, \perp_c, f), \langle b, \mathsf{sc}, f \rangle\} .$$

Its strict counterpart is

Table 2

The minimal model of graph F from Example 3.5. $[\![F^s]\!]$ indicates all the $\rho df_\perp$-interpretations in $\mathcal{M}^F$ that satisfy the graph F^s, $[\![F^{str} \cup (b, \perp_c, b)]\!]$ all the $\rho df_\perp$-interpretations in $\mathcal{M}^F$ that satisfy the graph $F^{str} \cup (b, \perp_c, b)$, and so on.

rank ∞	$\mathcal{M}^F \setminus (\text{rank } 0 \cup \text{rank } 1)$
rank 1	$[\![F^{str}]\!] \setminus \text{rank } 0$
rank 0	$[\![F^s]\!] \cup [\![F^{str} \cup (b, \perp_c, b)]\!]$

$$F^s = \{(p, \mathsf{sc}, b), (s, \mathsf{sc}, b), (p, \mathsf{sc}, e), (e, \perp_c, f), (b, \mathsf{sc}, f)\} \ .$$

Given F^s, we define the domains used for the models in $\mathfrak{R}_F$ following the construction in Lemma 2.3. The minimal model $\mathcal{R}_{\min F} \in \mathfrak{R}_F$ will be a model of height 1: all the $\rho df_\perp$-interpretations that satisfy F^s will have rank 0, all the $\rho df_\perp$-interpretations that satisfy $F^{str} \cup \{(b, \perp_c, b)\}$ will have rank 0, all the $\rho df_\perp$-interpretations that satisfy F^{str} but neither (b, sc, f) nor $(b, \perp_c, b)$ will have rank 1, and all the $\rho df_\perp$-interpretations that do not satisfy F^{str} will have infinite rank. The model is shown in Table 2.

It is easily verified that F^s implies $(p, \perp_c, p)$. Also, it is easy to check that F^s does not imply $(b, \perp_c, b)$ and does not imply $(s, \perp_c, s)$, since we are informed that penguins do not fly, but we are not informed that sparrows do not fly (there is no (s, sc, e) in our graph); also, F^{str} does not imply $(p, \perp_c, p)$, since F^{str} does not contain (b, sc, f) anymore. The resulting configuration of $\mathcal{R}_{\min F}$ is such that: all the $\rho df_\perp$-interpretations with height 0 satisfy $(p, \perp_c, p)$, but not all of them satisfy $(b, \perp_c, b)$ or $(s, \perp_c, s)$; not all the $\rho df_\perp$-interpretations with height 1 satisfy $(p, \perp_c, p)$.

It can also be verified that $\mathcal{R}_{\min F}$ is a model of F: all the $\rho df_\perp$-interpretations with finite height satisfy the strict part F^{str}. The minimal interpretations that do not satisfy $(b, \perp_c, b)$ have height 0 and satisfy F^s, and consequently satisfy (b, sc, f). Therefore, the defeasible triple $\langle b, \mathsf{sc}, f \rangle$ is satisfied by $\mathcal{R}_{\min F}$.

Note that there cannot be a model of F that is preferred to $\mathcal{R}_{\min F}$. If we move any $\rho df_\perp$-interpretation from rank 1 to rank 0, the resulting model would not satisfy $\langle b, \mathsf{sc}, f \rangle$ anymore: every interpretation in rank 1 does not satisfy neither $(b, \perp_c, b)$ nor (b, sc, f), hence, by moving one of them to rank 0 we obtain an interpretation $\mathcal{R}$ that would not satisfy $\langle b, \mathsf{sc}, f \rangle$, since we would have in $\mathsf{c\_min}(b, \mathcal{R})$ a $\rho df_\perp$-interpretation not satisfying (b, sc, f). Instead, if we move any $\rho df_\perp$-interpretation from rank ∞ to any finite rank, the resulting model would not satisfy F^{str}.

Being a minimal model of F, $\mathcal{R}_{\min F}$ satisfies the presumption of typicality. To check this, it suffices to determine what we can derive about sparrows: since sparrows do not have any exceptional property, they should inherit all the typical properties of birds, and we should be able to derive that they presumably fly. We need to check the $\rho df_\perp$ interpretations in $\mathsf{c\_min}(s, \mathcal{R}_{\min F})$, that must be in rank 0, since $F^s \not\vdash_{\rho df_\perp} (s, \perp_c, s)$. In particular, the $\rho df_\perp$ interpretations at rank 0 satisfy F^s or $F^{str} \cup \{(b, \perp_c, b)\}$, and the interpretations in $\mathsf{c\_min}(s, \mathcal{R}_{\min F})$ must be among those satisfying F^s, since $F^{str} \cup \{(b, \perp_c, b)\} \vdash_{\rho df_\perp} (s, \perp_c, s)$. This implies that all interpretations in $\mathsf{c\_min}(s, \mathcal{R}_{\min F})$ satisfy (s, sc, b) and (b, sc, f), that is, they satisfy (s, sc, f). Consequently, according to Definition 3.4, $\mathcal{R}_{\min F} \Vdash_{\rho df_\perp} \langle s, \mathsf{sc}, f \rangle$, that is, $F \vDash_{\min} \langle s, \mathsf{sc}, f \rangle$, as desired. $\square$

3.3. Exceptionality

Minimal entailment defines the semantics. We next define a decision procedure for it. To do so, we define the notion of *exceptionality*, a reformulation in our context of a property that is fundamental for RC [35]. Informally, a class t (or, respectively, a predicate t) is exceptional w.r.t. a defeasible graph if there is no typical situation in which t can be populated with some instance. Formally it corresponds to saying that in every ranked model of the graph, all the $\rho df_\perp$-interpretations with height 0 satisfy $(t, \perp_c, t)$ (respectively, $(t, \perp_p, t)$).

Definition 3.8 (*Exceptionality*). Let G be a defeasible $\rho df_\perp$-graph, $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model in $\mathfrak{R}_G$ and let t be a term.

1. We say that t is c-*exceptional* (resp. p-*exceptional*) w.r.t. $\mathcal{R}$ if for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, we have that $\mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_c, t)$ (resp. $\mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_p, t)$).
2. We say that t is c-*exceptional* (resp. p-*exceptional*) w.r.t. G if it is c-exceptional (resp. p-*exceptional*) w.r.t. all $\mathcal{R} \in \mathfrak{R}_G$.

In Example 3.5 p is c-exceptional w.r.t. $\mathcal{R}_{\min F}$. It turns out that in order to check exceptionality w.r.t. a graph it is sufficient to refer to the minimal model of the graph.

Proposition 3.2. *A term t is c-exceptional (resp. p-exceptional) w.r.t. a defeasible graph G iff it is c-exceptional (resp. p-exceptional) w.r.t. $\mathcal{R}_{\min G}$.*

Hence, in Example 3.5 p is c-exceptional w.r.t. the graph F, since it is exceptional w.r.t. the minimal model $\mathcal{R}_{\min F}$. We now prove that *a term t is c-exceptional (resp., p-exceptional) w.r.t. a defeasible graph G iff $G^s \vdash_{\rho df_\perp} (t, \perp_c, t)$ (resp., $G^s \vdash_{\rho df_\perp} (t, \perp_p, t)$)*, which will provide us a decision procedure to decide exceptionality.

In order to prove the above claim, we introduce the notion of proof tree and prove some lemmas beforehand. To start with, we reformulate the classical notion of proof tree for $\rho df_\perp$.

Definition 3.9 *($\rho df_\perp$ proof tree).* A $\rho df_\perp$ proof tree is a finite tree in which

- each node is a $\rho df_\perp$ triple;
- every node is connected to the node(s) immediately above through one of the inference rules (1)-(7) presented in Section 2.3.

Let T be a $\rho df_\perp$ proof tree, H be the set of the triples appearing as top nodes (called *leaves*) and let (s, p, o) be the triple appearing in the unique bottom node (called *root*). Then T is a $\rho df_\perp$ proof tree from H to (s, p, o).

For instance, in Example 3.2 we have a $\rho df_\perp$ proof tree from $\{(dU, \mathsf{sc}, yP), (yP, \mathsf{sc}, hP), (dU, \mathsf{sc}, uhP), (hP, \perp_{\mathsf{c}}, uhP)\}$ to $(dU, \perp_{\mathsf{c}}, dU)$. By Definitions 2.3 and 3.9, the following proposition is immediate to prove.

Proposition 3.3. *Let G be a $\rho df_\perp$ graph and (s, p, o) be a $\rho df_\perp$ triple. Then $G \vdash_{\rho df_\perp} (s, p, o)$ iff there is a $\rho df_\perp$ proof tree from H to (s, p, o) for some $H \subseteq G$.*

As next, we define a depth function on the trees.

Definition 3.10 *(Immediate subtree and depth function d).* Let T be a $\rho df_\perp$ proof tree. The set of the *immediate subtrees* of T, $\mathfrak{T} = \{T_1, \ldots, T_n\}$, are the trees T_i obtained from T by eliminating the root node of T.

The *depth* $d(T) \in \mathbb{N}$ of T is defined inductively the following way:

- if T is a single node then $d(T) = 0$;
- else, $d(T) = 1 + \max\{d(T') \mid T' \in \mathfrak{T}\}$.

Now we prove the following lemmas that will be used to establish that a term t is c-exceptional (resp., p-exceptional) w.r.t. a defeasible graph G iff $G^s \vdash_{\rho df_\perp} (t, \perp_{\mathsf{c}}, t)$ (resp., $G^s \vdash_{\rho df_\perp} (t, \perp_{\mathsf{p}}, t)$) (see Proposition 3.12 later on).

Lemma 3.4. *Let T be a $\rho df_\perp$ proof tree from H to (p, sc, q). Then T contains only triples of the form (A, sc, B).*

Lemma 3.5. *Let T be a $\rho df_\perp$ proof tree from H to $(p, \perp_{\mathsf{c}}, q)$. Then T contains only triples of the form (A, sc, B) or $(A, \perp_{\mathsf{c}}, B)$.*

Note that a tree proving $(t, \perp_{\mathsf{c}}, t)$ is just a particular case of Lemma 3.5.

Lemma 3.6. *Let G be a defeasible $\rho df_\perp$-graph, $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model in $\mathfrak{R}_G$ and $\langle p, \mathsf{sc}, q \rangle \in G^{def}$. For every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{c}}, p)$.*
Analogously, for every $\langle p, \mathsf{sp}, q \rangle \in G^{def}$ and every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{p}}, p)$.

We can extend the above lemma to derived subclass and subproperty triples.

Lemma 3.7. *Let G be a defeasible $\rho df_\perp$-graph, $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model in $\mathfrak{R}_G$ and let $G^s \vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ for some terms p, q. For every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{c}}, p)$.*
Analogously, if $G^s \vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ for some terms p, q, then for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{p}}, p)$.

Lemma 3.8. *Let G be a defeasible graph, p and q terms, and let $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ be G's minimal model. If $G^s \vdash_{\rho df_\perp} (p, \perp_{\mathsf{c}}, q)$ then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{c}}, q)$ for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$.*

The following is an immediate corollary of Lemma 3.8.

Corollary 3.9. *For any defeasible graph G and any term t, if $G^s \vdash_{\rho df_\perp} (t, \perp_{\mathsf{c}}, t)$ then t is c-exceptional w.r.t. G.*

Now we prove the analogous result for p-exceptionality.

Lemma 3.10. *Let G be a defeasible graph, p, q be any pair of terms, and let $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ be G's minimal model. If $G^s \vdash_{\rho df_\perp} (p, \perp_{\mathsf{p}}, q)$ then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathsf{p}}, q)$ for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$.*

An immediate corollary of Lemma 3.10 is the following.

Procedure ExceptionalC(G).

Input: Defeasible graph $G = G^{str} \cup G^{def}$
Output: Set $\epsilon^c(G)$ of c-exceptional triples w.r.t. G
  1: $\epsilon^c(G) := \emptyset$
  2: **for all** $\langle p, \mathrm{sc}, q \rangle \in G^{def}$ **do**
  3:     **if** $G^s \vdash_{\rho df_\perp} (p, \perp_c, p)$ **then**
  4:         $\epsilon^c(G) := \epsilon^c(G) \cup \{\langle p, \mathrm{sc}, q \rangle\}$
  5: **return** $\epsilon^c(G)$

Corollary 3.11. *For any defeasible graph G and any term t, if $G^s \vdash_{\rho df_\perp} (t, \perp_p, t)$ then t is p-exceptional w.r.t. G.*

Now we are ready to prove (see appendix) the main proposition of this section.

Proposition 3.12. *A term t is c-exceptional (resp., p-exceptional) w.r.t. a defeasible graph G iff $G^s \vdash_{\rho df_\perp} (t, \perp_c, t)$ (resp., $G^s \vdash_{\rho df_\perp} (t, \perp_p, t)$).*

Proposition 3.12 gives us a correct and complete correspondence between the semantic notions of c-exceptionality and p-exceptionality with the $\rho df_\perp$ decision procedure $\vdash_{\rho df_\perp}$. This correspondence allows us then to compute all the c-exceptional triples and all the p-exceptional ones, as illustrated by the procedures ExceptionalC(G) and ExceptionalP(G), respectively, where the notion of exceptional triple is defined in the obvious way:

Definition 3.11 *(Exceptional triple).* Let G be a defeasible $\rho df_\perp$-graph. We say that a defeasible triple $\langle p, \mathrm{sc}, q \rangle \in G^{def}$ (resp. $\langle p, \mathrm{sp}, q \rangle \in G^{def}$) is c-*exceptional* (resp. p-*exceptional*) w.r.t. G if p is c-exceptional (resp. p-*exceptional*) w.r.t. G.

Procedure ExceptionalP(G).

Input: Defeasible graph $G = G^{str} \cup G^{def}$
Output: Set $\epsilon^p(G)$ of p-exceptional triples w.r.t. G
  1: $\epsilon^p(G) := \emptyset$
  2: **for all** $\langle p, \mathrm{sp}, q \rangle \in G^{def}$ **do**
  3:     **if** $G^s \vdash_{\rho df_\perp} (p, \perp_p, p)$ **then**
  4:         $\epsilon^p(G) := \epsilon^c(G) \cup \{\langle p, \mathrm{sp}, q \rangle\}$
  5: **return** $\epsilon^p(G)$

Procedures ExceptionalC(G) and ExceptionalP(G) correctly model exceptionality, as proved by the following immediate corollary of Proposition 3.12.

Corollary 3.13. *Given a defeasible graph G and a defeasible triple $\langle p, \mathrm{sc}, q \rangle \in G^{def}$ (resp., $\langle p, \mathrm{sp}, q \rangle \in G^{def}$), $\langle p, \mathrm{sc}, q \rangle \in \epsilon^c(G)$ (resp., $\langle p, \mathrm{sp}, q \rangle \in \epsilon^p(G)$) iff it is c-exceptional (resp. p-exceptional) w.r.t. G.*

Example 3.6 *(Running example cont.).* Like in Example 3.2, it is easily verified that, besides $G^s \vdash_{\rho df_\perp} (dU, \perp_c, dU)$ and $G^s \vdash_{\rho df_\perp} (uDC, \perp_p, uDC)$, we also have $G^s \vdash_{\rho df_\perp} (cDU, \perp_c, cDU)$.[16] As a consequence,

$$\mathtt{ExceptionalC}(G) = \{\langle dU, \mathrm{sc}, uhP \rangle, \langle dU, \mathrm{sc}, yP \rangle, \langle cDU, \mathrm{sc}, hP \rangle\}$$

$$\mathtt{ExceptionalP}(G) = \{\langle uDC, \mathrm{sp}, hDI \rangle\} . \quad \square$$

3.4. The ranking procedure

Iteratively applied, the notions of c-exceptionality and p-exceptionality allow us to associate to every term, *i.e.*, to every defeasible triple, a rank value w.r.t. a defeasible graph G. Specifically, we introduce a ranking procedure, called Ranking(G), that orders the defeasible information in G^{def} into a sequence $\mathsf{D}_0, \ldots, \mathsf{D}_n, \mathsf{D}_\infty$ of sets D_i of defeasible triples, with $n \geqslant 0$ and D_∞ possibly empty. The procedure is shown below.

The ranking procedure is built on top of the ExceptionalC and ExceptionalP procedures, using $\rho df_\perp$ decision steps only.

Example 3.7 *(Running example cont.).* Consider Example 3.6. Let us order G^{def} by running Ranking(G). So, consider $\mathsf{D}_0 := G^{def}$. From Example 3.6 we immediately have

$$\mathtt{ExceptionalC}(G^{str} \cup \mathsf{D}_0) := \mathtt{ExceptionalC}(G)$$

$$\mathtt{ExceptionalP}(G^{str} \cup \mathsf{D}_0) := \mathtt{ExceptionalP}(G)$$

$$\mathsf{D}_1 := \{\langle dU, \mathrm{sc}, uhP \rangle, \langle dU, \mathrm{sc}, yP \rangle, \langle cDU, \mathrm{sc}, hP \rangle, \langle uDC, \mathrm{sp}, hDI \rangle\} .$$

---

[16] $uDC \mapsto usesDrugControlled$, $cDU \mapsto controlledDrugUser$ and $hDI \mapsto hasDrugIndependence$.

Procedure Ranking(G).

Input: Defeasible graph $G = G^{str} \cup G^{def}$
Output: ranking $r(G) = \{D_0, \dots, D_n, D_\infty\}$
1: $D_0 := G^{def}$
2: $i := 0$
3: **repeat**
4:    $D_{i+1} := \texttt{ExceptionalC}(G^{str} \cup D_i) \cup \texttt{ExceptionalP}(G^{str} \cup D_i)$
5:    $i := i + 1$
6: **until** $D_i = D_{i+1}$
7: $D_\infty := D_i$
8: $r(G) := \{D_0, \dots, D_{i-1}, D_\infty\}$
9: **return** $r(G)$

Now, it can be verified that w.r.t. $G^{str} \cup D_1$, dU and uDC are not exceptional, while cDU is, as

$$G^{str} \cup (D_1)^s \nvdash_{\rho df_\perp} (dU, \perp_c, dU)$$

$$G^{str} \cup (D_1)^s \nvdash_{\rho df_\perp} (uDC, \perp_p, uDC)$$

$$G^{str} \cup (D_1)^s \vdash_{\rho df_\perp} (cDU, \perp_c, cDU) .$$

Therefore,

$$D_2 := \{\langle cDU, \mathsf{sc}, hP \rangle\} .$$

Finally, it is easy to check that

$$D_3 := \emptyset$$

that is,

$$D_\infty := \emptyset$$

and, thus,

$$r(G) := \{D_0, D_1, D_2, D_\infty\} . \quad \square$$

The following example shows an example of a triple that turns out to have infinite rank.

Example 3.8. Let L be the graph $\{(p, \mathsf{dom}, r), (q, \mathsf{dom}, t), (r, \perp_c, t), \langle q, \mathsf{sc}, p \rangle\}$. The only defeasible triple is $\langle q, \mathsf{sc}, p \rangle$, hence $D_0 = \{\langle q, \mathsf{sc}, p \rangle\}$. Applying the procedure $\texttt{ExceptionalC}(L)$, from the graph L^s we conclude the triple $(q, \perp_p, q)$ with the following derivation:

$$\cfrac{\cfrac{(p, \mathsf{dom}, r) \quad (q, \mathsf{dom}, t) \quad (r, \perp_c, t)}{(p, \perp_p, q)} (7a') \quad (q, \mathsf{sp}, p)}{(q, \perp_p, q)} (6b)$$

Hence we have $D_1 = D_0$, that is, we have a fixed point, and we can conclude $D_\infty = D_0 = \{\langle q, \mathsf{sc}, p \rangle\}$. $\quad \square$

Now we prove that the ranking procedure $\texttt{Ranking}(G)$ correctly mirrors the ranking of the defeasible information w.r.t. the height functions h_G^c and h_G^p. Specifically, we want to show (see Proposition 3.19 later on) that

- for $i < n$, $\langle p, \mathsf{sc}, q \rangle \in D_i \setminus D_{i+1}$ iff $h_G^c(p) = i$;
- for $i = n$, $\langle p, \mathsf{sc}, q \rangle \in D_n \setminus D_\infty$ iff $h_G^c(p) = n$,

and an analogous result w.r.t. h_G^p.

To do so, we need to introduce some preliminary constructions and lemmas. To start with, let us note that the information in a graph $G = G^{str} \cup G^{def}$ is ranked from the semantical point of view by the minimal model $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ and the height functions h_G^c and h_G^p defined on it. Let $\mathcal{R}_{\min G}^i = (\mathcal{M}^i, r^i)$ be the submodel of $\mathcal{R}_{\min G}$ obtained by eliminating all the $\rho df_\perp$ interpretations in $\mathcal{M}$ whose height is strictly less than i ($i \geqslant 0$). Specifically, given a graph $G = G^{str} \cup G^{def}$ and its minimal model $\mathcal{R}_{\min G} = (\mathcal{M}, r)$, $\mathcal{R}_{\min G}^i = (\mathcal{M}^i, r^i)$ is defined as follows:

- $\mathcal{M}^i := \{I \in \mathcal{M} \mid r(I) \geqslant i\}$;
- $r^i(I) := r(I) - i$, for every $I \in \mathcal{M}^i$.

That is, $\mathcal{R}_{\min G}^i$ is obtained from $\mathcal{R}_{\min G}$ by eliminating all the interpretations that have a rank of $i-1$ or less. We next show that the interpretation $\mathcal{R}_{\min G}^i$ models the defeasible information in the graph that has a higher level of exceptionality: the higher the value of i, the higher the level of exceptionality. That is, given a defeasible graph $G = G^{str} \cup G^{def}$, let

$$G^i := G^{str} \cup G_i^{def} \ , \tag{6}$$

where

$$G_i^{def} := \{\langle p, \mathtt{sc}, q \rangle \in G^{def} \mid h_G^{\mathtt{c}}(p) \geqslant i\} \cup \{\langle p, \mathtt{sp}, q \rangle \in G^{def} \mid h_G^{\mathtt{p}}(p) \geqslant i\} \ . \tag{7}$$

Then the following proves that $\mathcal{R}_{\min G}^i$ is indeed a model of G^i.

Lemma 3.14. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ its minimal ranked model. Then $\mathcal{R}_{\min G}^i = (\mathcal{M}^i, r^i)$ is a model of the subgraph $G^i = G^{str} \cup G_i^{def}$.*

Now, it turns out that the minimal model for G^i, that is built using the set of $\rho df_\perp$-interpretations $\mathcal{M}$, can easily be defined by extending $\mathcal{R}_{\min G}^i$. That is, let $\mathcal{R}_i^* = (\mathcal{M}, r_i^*)$ be a ranked interpretation where r_i^* is defined as

$$r_i^*(\mathcal{I}) = \begin{cases} r^i(\mathcal{I}) & \text{if } \mathcal{I} \in \mathcal{M}^i \\ 0 & \text{otherwise .} \end{cases}$$

The following holds.

Lemma 3.15. *Given a defeasible graph G, $\mathcal{R}_i^*$ is the minimal model of the subgraph G^i.*

The following lemma connects the height of a term with the notion of exceptionality in the models $\mathcal{R}_i^*$.

Lemma 3.16. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and let $\mathcal{R}_{\min G}$ be its minimal model, with $h(\mathcal{R}_{\min G}) = n$. For every $i \leqslant n$ and term p s.t. $h_G^{\mathtt{c}}(p) \geqslant i$ (resp., $h_G^{\mathtt{p}}(p) \geqslant i$) p is c-exceptional (resp., p-exceptional) w.r.t. $\mathcal{R}_{\min G}^i = (\mathcal{M}^i, r^i)$ iff it is c-exceptional (resp., p-exceptional) w.r.t. $\mathcal{R}_i^* = (\mathcal{M}, r_i^*)$.*

The following lemma connects the height to the computation of exceptionality.

Lemma 3.17. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, and $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ its minimal model, with $h(\mathcal{R}_{\min G}) = n$, and $\langle p, \mathtt{sc}, q \rangle \in G^{def}$. Then,*

- *for every $i < n$, $h_G^{\mathtt{c}}(p) \geqslant i + 1$ iff $\langle p, \mathtt{sc}, q \rangle \in \mathtt{ExceptionalC}(G^i)$;*
- *for $i = n$, $h_G^{\mathtt{c}}(p) = \infty$ iff $\langle p, \mathtt{sc}, q \rangle \in \mathtt{ExceptionalC}(G^i)$.*

Analogously, let $\langle p, \mathtt{sp}, q \rangle \in G^{def}$. Then,

- *for every $i < n$, $h_G^{\mathtt{p}}(p) \geqslant i + 1$ iff $\langle p, \mathtt{sp}, q \rangle \in \mathtt{ExceptionalP}(G^i)$;*
- *for $i = n$, $h_G^{\mathtt{p}}(p) = \infty$ iff $\langle p, \mathtt{sp}, q \rangle \in \mathtt{ExceptionalP}(G^i)$.*

We now move on to prove the correspondence between the ranking procedure and the height (the semantic ranking of the defeasible information).

Given a defeasible graph $G = G^{str} \cup G^{def}$, $G_i^{\mathtt{D}}$ ($i \geqslant 0$) is the subgraph of G defined as follows:

$$G_i^{\mathtt{D}} := G^{str} \cup \mathtt{D}_i \ , \tag{8}$$

where $\mathtt{D}_i$ is in the output of $\mathtt{Ranking}(G)$, i.e., $\mathtt{D}_i \in \mathtt{r}(G)$. Our objective now is to prove that $G^i = G_i^{\mathtt{D}}$. To prove that for every i, $G^i = G_i^{\mathtt{D}}$, it is sufficient to prove that for every rank value $i \neq \infty$ a term t is exceptional w.r.t. $G_i^{\mathtt{D}}$ iff it is exceptional w.r.t. $\mathcal{R}_{\min G}^i$.

The following can be shown.

Lemma 3.18. *Let G be a defeasible graph, with n being the height of its minimal model. Then, for every $i \leqslant n$, $G^i = G_i^{\mathtt{D}}$.*

Now we can state the main proposition for our ranking procedure.

Proposition 3.19. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, $\langle p, \mathtt{sc}, q \rangle$ (resp., $\langle p, \mathtt{sp}, q \rangle$) be in G^{def}, and $\mathtt{r}(G) = \{\mathtt{D}_0, \ldots, \mathtt{D}_n, \mathtt{D}_\infty\}$ be the ranking obtained by $\mathtt{Ranking}(G)$. The following statements hold:*

- *for $i < n$, $\langle p, \mathtt{sc}, q \rangle \in \mathtt{D}_i \setminus \mathtt{D}_{i+1}$ iff $h_G^{\mathtt{c}}(p) = i$;*

- *for $i = n$, $\langle p, \mathrm{sc}, q \rangle \in \mathrm{D}_n \setminus \mathrm{D}_\infty$ iff $h_G^c(p) = n$.*

Proof. Immediate consequence of Lemma 3.18. $\square$

Another consequence of Lemma 3.18 is the following corollary, that will be useful later on.

Corollary 3.20. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, with $\mathrm{r}(G) = \{\mathrm{D}_0, \ldots, \mathrm{D}_n, \mathrm{D}_\infty\}$ being the ranking obtained by* $\mathtt{Ranking}(G)$*, and let p be a term. The following statements hold:*

- $h_G^c(p) = 0$ *if and only if* $G_0^{\mathrm{D}} \nvdash_{\rho df_\perp} (p, \perp_c, p)$*;*
- *for every i, $0 < i \leqslant n$, $h_G^c(p) = i$ if and only if $G_{i-1}^{\mathrm{D}} \vdash_{\rho df_\perp} (p, \perp_c, p)$ and $G_i^{\mathrm{D}} \nvdash_{\rho df_\perp} (p, \perp_c, p)$;*
- $h_G^c(p) = \infty$ *if and only if* $G_n^{\mathrm{D}} \vdash_{\rho df_\perp} (p, \perp_c, p)$*.*

Analogously,

- $h_G^p(p) = 0$ *if and only if* $G_0^{\mathrm{D}} \nvdash_{\rho df_\perp} (p, \perp_p, p)$*;*
- *for every i, $0 < i \leqslant n$, $h_G^p(p) = i$ if and only if $G_{i-1}^{\mathrm{D}} \vdash_{\rho df_\perp} (p, \perp_p, p)$ and $G_i^{\mathrm{D}} \nvdash_{\rho df_\perp} (p, \perp_p, p)$;*
- $h_G^c(p) = \infty$ *if and only if* $G_n^{\mathrm{D}} \vdash_{\rho df_\perp} (p, \perp_p, p)$*.*

Proof. Immediate from Proposition 3.12, Lemma 3.17 and Lemma 3.18. $\square$

3.5. Decision procedures for defeasible ρdf

In this section we present our main decision procedures. That is, given triples $\langle p, \mathrm{sc}, q \rangle$, $\langle p, \mathrm{sp}, q \rangle$ and (s, p, o) as queries, the procedures below decide whether they are or not minimally entailed by a defeasible graph G.

Remark 3.2. Given a fixed defeasible graph G, we assume that its ranking $\mathrm{r}(G) = \{\mathrm{D}_0, \ldots, \mathrm{D}_n, \mathrm{D}_\infty\} = \mathtt{Ranking}(G)$ has already been computed.

We start with the case of $\rho df_\perp$-triples of the form (s, p, o). The procedure $\mathtt{StrictMinEntailment}$ below decides whether $G \vDash_{\min} (s, p, o)$ holds. Essentially, given G, we take the strict part G^{str} to which we add disjointness triples related to defeasible ones with infinite rank and then test for $\rho df_\perp$ entailment.

Procedure $\mathtt{StrictMinEntailment}(G, \mathrm{r}(G), (s, p, o))$.

Input: Graph $G = G^{str} \cup G^{def}$, ranking $\mathrm{r}(G) = \{\mathrm{D}_0, \ldots, \mathrm{D}_n, \mathrm{D}_\infty\}$, a $\rho df_\perp$-triple (s, p, o)
Output: $\mathtt{true}$ if $G \vDash_{\min} (s, p, o)$; $\mathtt{false}$ otherwise
1: $G' := G^{str} \cup \{(p, \perp_c, p) \mid \langle p, \mathrm{sc}, q \rangle \in \mathrm{D}_\infty\} \cup \{(p, \perp_p, p) \mid \langle p, \mathrm{sp}, q \rangle \in \mathrm{D}_\infty\}$
2: **return** $G' \vdash_{\rho df_\perp} (s, p, o)$

The following lemma can be proved, which motivates the construction of graph G' in the procedure $\mathtt{StrictMinEntailment}$.

Lemma 3.21. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, $\mathrm{r}(G) = \{\mathrm{D}_0, \ldots, \mathrm{D}_n, \mathrm{D}_\infty\}$ its ranking, and $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ its minimal model, with $\mathcal{I} \in \mathcal{M}$. Then $\mathcal{I} \in \mathcal{M}_{\mathbb{N}}$ iff $\mathcal{I} \Vdash_{\rho df_\perp} G^{str} \cup \{(p, \perp_c, p) \mid \langle p, \mathrm{sc}, q \rangle \in \mathrm{D}_\infty\} \cup \{(p, \perp_p, p) \mid \langle p, \mathrm{sp}, q \rangle \in \mathrm{D}_\infty\}$.*

The following theorem establishes correctness and completeness of the $\mathtt{StrictMinEntailment}$ procedure.

Theorem 3.22. *Consider a defeasible graph G and a $\rho df_\perp$-triple (s, p, o). Then*

$$G \vDash_{\min} (s, p, o) \text{ iff } \mathtt{StrictMinEntailment}(G, \mathrm{r}(G), (s, p, o)) .$$

Proof. By Lemma 3.21 and Theorem 2.1 we have that $G \vDash_{\min} (s, p, o)$ if and only if $G^{str} \cup \{(p, \perp_c, p) \mid \langle p, \mathrm{sc}, q \rangle \in \mathrm{D}_\infty\} \cup \{(p, \perp_p, p) \mid \langle p, \mathrm{sp}, q \rangle \in \mathrm{D}_\infty\} \vDash_{\rho df_\perp} (s, p, o)$. Therefore, the procedure $\mathtt{StrictMinEntailment}$ is correct and complete. $\square$

Example 3.9 (*Running example cont.*)**.** Consider Example 3.7. We have seen that $\mathrm{r}(G) = \{\mathrm{D}_0, \mathrm{D}_1, \mathrm{D}_2, \mathrm{D}_\infty\}$, with $\mathrm{D}_\infty = \emptyset$. Let us show now that "tom is a person". To do so, let us show that

$$\mathtt{StrictMinEntailment}(G, \mathrm{r}(G), (tom, \mathsf{type}, person)) = \mathtt{true} .$$

As $\mathrm{D}_\infty = \emptyset$, we have that $G' = G^{str}$. Now, in Example 2.4 we have seen that $G^{str} \vdash_{\rho df_\perp} (tom, \mathsf{type}, person)$. Therefore, $\mathtt{StrictMinEntailment}(G, \mathrm{r}(G), (tom, \mathsf{type}, person))$ returns $\mathtt{true}$ and by Theorem 3.22 we can conclude that $G \vDash_{\min} (tom, \mathsf{type}, person)$. $\square$

Procedure DefMinEntailmentC$(G, \mathrm{r}(G), \langle p, \mathrm{sc}, q \rangle)$.

Input: Graph $G = G^{str} \cup G^{def}$, ranking $\mathrm{r}(G) = \{\mathrm{D}_0, \dots, \mathrm{D}_n, \mathrm{D}_\infty\}$, defeasible triple $\langle p, \mathrm{sc}, q \rangle$

Output: true if $G \vDash_{\min} \langle p, \mathrm{sc}, q \rangle$; false otherwise

  1:   $i := 0$
  2:   $\mathrm{D}_{n+1} := \mathrm{D}_\infty$
  3:   **repeat**
  4:      **if** $i \leqslant n$ **then**
  5:         $G' := G^{str} \cup (\mathrm{D}_i)^s$
  6:         $j := i$
  7:         $i := i + 1$
  8:      **else**
  9:         **return** true
10:   **until** $G' \nvdash_{\rho df_\perp} (p, \perp_{\mathrm{c}}, p)$
11:   $\mathrm{D}^p := \{\langle r, \mathrm{sc}, s \rangle \mid \langle r, \mathrm{sc}, s \rangle \in \mathrm{D}_j \setminus \mathrm{D}_{j+1}\}$
12:   **return** $G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$

We next consider triples of the form $\langle p, \mathrm{sc}, q \rangle$. The decision procedure DefMinEntailmentC decides whether $G \vDash_{\min} \langle p, \mathrm{sc}, q \rangle$ holds (we will later consider also an analogous procedure for triples of form $\langle p, \mathrm{sp}, q \rangle$). Roughly, given a graph $G = G^{str} \cup G^{def}$, the ranking $\mathrm{r}(G) = \{\mathrm{D}_0, \dots, \mathrm{D}_n, \mathrm{D}_\infty\}$ of defeasible triples in G^{def} and a query triple $\langle p, \mathrm{sc}, q \rangle$, we first search for the lowest ranked D_j (smallest j) such that p is not c-exceptional w.r.t. $G^{str} \cup (\mathrm{D}_j)^s$, i.e., $G^{str} \cup (\mathrm{D}_j)^s \nvdash_{\rho df_\perp} (p, \perp_{\mathrm{c}}, p)$. Then, from such a set D_j we remove all higher ranked defeasible triples, and we call D^p the resulting set. We conclude by checking whether the query triple $\langle p, \mathrm{sc}, q \rangle$ is entailed by $G^{str} \cup (\mathrm{D}^p)^s$, i.e., $G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$.

To establish the correctness and completeness of procedure DefMinEntailmentC we prove some lemmas beforehand. Let us recall that, by Lemma 3.4, every proof tree with a triple (p, sc, q) as root contains only triples of the form (A, sc, B). Using such a lemma, the following can be proven.

Lemma 3.23. *Let T be a proof tree from a graph H to a triple (p, sc, q), and let $\mathcal{I}$ be a model of H. If $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_{\mathrm{c}}, s)$ for some triple $(s, \mathrm{sc}, o) \in H$, then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_{\mathrm{c}}, p)$.*

Lemma 3.24. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, and let $\mathrm{r}(G) = \{\mathrm{D}_0, \dots, \mathrm{D}_n, \mathrm{D}_\infty\}$ be its ranking. For any pair of terms p, q s.t. $h_G^c(p) \leqslant n$,*

$$G \vDash_{\min} \langle p, \mathrm{sc}, q \rangle \text{ iff } G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q),$$

where D^p is defined as in the DefMinEntailmentC procedure.

The following theorem establishes correctness and completeness of the DefMinEntailmentC procedure.

Theorem 3.25. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and let $\langle p, \mathrm{sc}, q \rangle$ be a defeasible triple. Then,*

$$G \vDash_{\min} \langle p, \mathrm{sc}, q \rangle \text{ iff } \texttt{DefMinEntailmentC}(G, \mathrm{r}(G), \langle p, \mathrm{sc}, q \rangle).$$

Example 3.10 (*Running example cont.*). Consider again Example 3.7. Let us show that "young drug users are usually unhappy" (*cf.* statement (8) in Example 1.1). To do so, let us show that

$$\texttt{DefMinEntailmentC}(G, \mathrm{r}(G), \langle yDU, \mathrm{sc}, uhP \rangle) = \texttt{true}.$$

For $i = 0$, we have $G' = G^s$, and $G^s \vdash_{\rho df_\perp} (yDU, \perp_{\mathrm{c}}, yDU)$, since $G^s \vdash_{\rho df_\perp} (dU, \perp_{\mathrm{c}}, dU)$, that together with (yDU, sc, dU) implies $(yDU, \perp_{\mathrm{c}}, yDU)$ by the rule ($EmptySP'$).

For $i = 1$, we have $G' = G \cup (\mathrm{D}_1)^s$, and $G' \nvdash_{\rho df_\perp} (yDU, \perp_{\mathrm{c}}, yDU)$, since $G' \nvdash_{\rho df_\perp} (dU, \perp_{\mathrm{c}}, dU)$. Therefore, according to line 11 we have

$$\mathrm{D}^p := \mathrm{D}_1 \setminus \mathrm{D}_2 = \{\langle dU, \mathrm{sc}, uhP \rangle, \langle dU, \mathrm{sc}, yP \rangle, \langle uDC, \mathrm{sp}, hDI \rangle\}.$$

Finally, it is easy to check that:

$$G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (yDU, \mathrm{sc}, uhP)$$

since $(dU, \mathrm{sc}, uhP) \cup (yDU, \mathrm{sc}, dU) \subseteq G^{str} \cup (\mathrm{D}^{yDU})^s$. Therefore,

$$\texttt{DefMinEntailmentC}(G, \mathrm{r}(G), \langle yDU, \mathrm{sc}, uhP \rangle) = \texttt{true}$$

and by Theorem 3.25 we can conclude that $G \vDash_{\min} \langle yDU, \mathrm{sc}, uhP \rangle$.

Please note that $G \nvDash_{\min} \langle yDU, \mathrm{sc}, hP \rangle$ as $(yP, \mathrm{sc}, hP) \notin G^{str} \cup (\mathrm{D}^p)^s$. $\square$

Procedure $\texttt{DefMinEntailmentP}(G, \text{r}(G), \langle p, \text{sp}, q \rangle)$.

Input: Graph $G = G^{str} \cup G^{def}$, ranking $\text{r}(G) = \{D_0, \ldots, D_n, D_\infty\}$, defeasible triple $\langle p, \text{sp}, q \rangle$
Output: $\texttt{true}$ if $G \vDash_{\min} \langle p, \text{sp}, q \rangle$; $\texttt{false}$ otherwise

1: $i := 0$
2: $D_{n+1} := D_\infty$
3: **repeat**
4:    **if** $i \leqslant n$ **then**
5:        $G' := G^{str} \cup (D_i)^s$
6:        $j := i$
7:        $i := i + 1$
8:    **else**
9:        **return** $\texttt{true}$
10: **until** $G' \nvdash_{\rho df_\perp} (p, \perp_p, p)$
11: $D^p := \{\langle r, \text{sp}, s \rangle \mid \langle r, \text{sp}, s \rangle \in D_j \setminus D_{j+1}\}$
12: **return** $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \text{sp}, q)$

Remark 3.3 *(Drowning problem)*. We take the opportunity, using Example 3.10, to show that our entailment relation presents the same inferential limit that characterises Rational Closure, and that motivates the entailment relation we are going to present in Section 4 later on. By referring to Example 3.10, please note that indeed we also have

$$G \nvDash_{\min} \langle dU, \text{sc}, s \rangle$$

as $\texttt{DefMinEntailmentC}(G, \text{r}(G), \langle dU, \text{sc}, s \rangle)$ returns $\texttt{false}$, *i.e.*, we cannot derive that "drug users are usually students" (*cf.* statement (10) in Example 1.1).

This latter example shows a well-known behaviour of RC: in the case a class is exceptional w.r.t. a super-class, it does not inherit any of the typical properties of the super-class. In this specific case, since drug users are exceptional young people (they are not happy), in RC they do not inherit any of the typical properties of young people: in particular, we cannot conclude that drug users are students. This behaviour, called the *drowning problem* [5], may not be desirable in some applications and various RC extensions have been developed to overcome this problem [14,15,34]. In this work, we will address it in Section 4 via DINs.

Eventually, an analogous procedure to $\texttt{DefMinEntailmentC}$ can be defined for the case of defeasible triples of the form $\langle p, \text{sp}, q \rangle$, as illustrated by the $\texttt{DefMinEntailmentP}$ procedure.

The proof that procedure $\texttt{DefMinEntailmentP}$ is correct and complete w.r.t. $\vDash_{\min}$ proceeds similarly to the one for $\texttt{DefMinEntailmentC}$.

Specifically, we first prove the analogues of Lemma 3.4.

Lemma 3.26. *Let T be a $\rho df_\perp$ proof tree from H to (p, sp, q). Then T contains only triples of the form (A, sp, B).*

Proof. The proof is similar to the proof of Lemma 3.4, we just need to refer to rule (2a) instead of (3a). $\square$

Also for the other propositions the proof is analogous to the correspondent propositions for procedure $\texttt{DefMinEntailmentC}$. It suffices to change every instance of sc with sp, $\perp_c$ with $\perp_p$, every reference to Lemma 3.4 with Lemma 3.26, and so on. Specifically, we have

Lemma 3.27. *Let T be a proof tree from a graph H to a triple (p, sp, q), and let $\mathcal{I}$ be a model of H. If $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_p, s)$ for some triple $(s, \text{sp}, o) \in H$, then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, p)$.*

Lemma 3.28. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, and let $\text{r}(G) = \{D_0, \ldots, D_n, D_\infty\}$ be its ranking. For any pair of terms p, q s.t. $h_G^p(p) \leqslant n$,*

$$G \vDash_{\min} \langle p, \text{sp}, q \rangle \text{ iff } G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \text{sp}, q) ,$$

where D^p is defined as in the $\texttt{DefMinEntailmentP}$ procedure.

Eventually, we conclude with the following theorem establishing correctness and completeness of the $\texttt{DefMinEntailmentP}$ procedure.

Theorem 3.29. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and let $\langle p, \text{sp}, q \rangle$ be a defeasible triple. Then,*

$$G \vDash_{\min} \langle p, \text{sp}, q \rangle \text{ iff } \texttt{DefMinEntailmentP}(G, \text{r}(G), \langle p, \text{sp}, q \rangle) .$$

Example 3.11 (*Running example cont.*). Consider again Example 3.7. Likewise Example 3.10, it is not difficult to show that indeed

$$\text{DefMinEntailmentP}(G, \text{r}(G), \langle uDC, \text{sp}, hDI \rangle) = \texttt{true} .$$

Therefore, by Theorem 3.29 we can conclude that $G \models_{\min} \langle uDC, \text{sp}, hDI \rangle$. That is, "someone that uses some drug in a controlled way has usually drug independence to that drug". However, please also note that $G \nvDash_{\min} \langle uDC, \text{sp}, hDA \rangle$ instead[17] as one may check that

$$\text{DefMinEntailmentP}(G, \text{r}(G), \langle uDC, \text{sp}, hDA \rangle) = \texttt{false} . \quad \square$$

For completeness, we conclude this section with some examples relying on the "penguin" Example 3.5.

Example 3.12. Consider the graph H, similar to the graph F from Example 3.5, but with the following changes:

- the triple $\langle b, \text{sc}, hf \rangle$ has been added, where hf is read as "having feathers";
- the information that penguins do not fly has been made defeasible, that is, the triple (p, sc, e) has been substituted by $\langle p, \text{sc}, e \rangle$;
- the triples $\langle pj, \text{sc}, f \rangle$ and (pj, sc, p) have been added, where pj is read as "penguins with jet-packs".

That is, $H = H^{str} \cup H^{def}$, with

$$H^{str} = \{(p, \text{sc}, b), (s, \text{sc}, b), (e, \bot_c, f), (pj, \text{sc}, p)\}$$

$$H^{def} = \{\langle b, \text{sc}, f \rangle, \langle p, \text{sc}, e \rangle, \langle b, \text{sc}, hf \rangle, \langle pj, \text{sc}, f \rangle\} .$$

Given $H^s = \{(p, \text{sc}, b), (s, \text{sc}, b), (b, \text{sc}, f), (p, \text{sc}, e), (e, \bot_c, f), (b, \text{sc}, hf), (pj, \text{sc}, f), (pj, \text{sc}, p)\}$, it is easy to check that:

$$H^s \nvdash_{\rho df_\bot} (b, \bot_c, b)$$

$$H^s \nvdash_{\rho df_\bot} (s, \bot_c, s)$$

$$H^s \vdash_{\rho df_\bot} (p, \bot_c, p)$$

$$H^s \vdash_{\rho df_\bot} (pj, \bot_c, pj) .$$

Therefore, within procedure $\texttt{Ranking}(H)$, we have

$$D_1 = \{\langle p, \text{sc}, e \rangle, \langle pj, \text{sc}, f \rangle\} .$$

Since

$$H^{str} \cup (D_1)^s \nvdash_{\rho df_\bot} (p, \bot_c, p)$$

$$H^{str} \cup (D_1)^s \vdash_{\rho df_\bot} (pj, \bot_c, pj)$$

the procedure $\texttt{Ranking}(H)$ determines

$$D_2 = \{\langle pj, \text{sc}, f \rangle\} ,$$

and terminates with

$$D_\infty = \emptyset ,$$

as $H^{str} \cup (D_2)^s \nvdash_{\rho df_\bot} (pj, \bot_c, pj)$.

Having computed $\texttt{Ranking}(H)$, we can now check what is minimally entailed.

At first, we can check that penguins do not fly, that is, we can make the following queries:

Triple $\langle p, \text{sc}, e \rangle$**.** We apply the procedure $\texttt{DefMinEntailmentC}(H, \text{r}(H), \langle p, \text{sc}, e \rangle)$ and have: $G' \nvdash_{\rho df_\bot} (p, \bot_c, p)$ for $i = 1$, that implies $D^p = \{\langle p, \text{sc}, e \rangle\}$. $G^{str} \cup (D^p)^s \vdash_{\rho df_\bot} (p, \text{sc}, e)$, since $(D^p)^s = \{(p, \text{sc}, e)\}$, and $\texttt{DefMinEntailmentC}(H, \text{r}(H), \langle p, \text{sc}, e \rangle)$ returns $\texttt{true}$.

Triple $\langle p, \text{sc}, f \rangle$**.** We apply the procedure $\texttt{DefMinEntailmentC}(H, \text{r}(H), \langle p, \text{sc}, f \rangle)$ and have: $G^{str} \cup (D^p)^s \nvdash_{\rho df_\bot} (p, \text{sc}, f)$, and $\texttt{DefMinEntailmentC}(H, \text{r}(H), \langle p, \text{sc}, f \rangle)$ returns $\texttt{false}$.

The procedure $\texttt{DefMinEntailmentC}$ allows us to correctly derive that "penguins do not fly", while it avoids to derive that "penguins fly". What about the other typical property of birds in our graph, that is, having feathers?

---

[17] $hDA \mapsto hasDrugAddiction$.

Triple $\langle p, \mathsf{sc}, hf \rangle$**.** We apply the procedure $\mathtt{DefMinEntailmentC}(H, r(H), \langle p, \mathsf{sc}, hf \rangle)$ and have: $G^{str} \cup (\mathrm{D}^p)^s \nvdash_{\rho df_\perp} (p, \mathsf{sc}, hf)$, and

$$\mathtt{DefMinEntailmentC}(H, r(H), \langle p, \mathsf{sc}, hf \rangle) = \mathtt{false} \ .$$

This latter example shows again the drowning problem (see Remark 3.3). In this specific case, since penguins are exceptional birds (they do not fly), in RC they do not inherit any of the typical properties of birds: that is, we cannot conclude that penguins have feathers.

Now we move to check the behaviour of sub-classes that do not show any exceptional behaviour. From our graph we know that sparrows are birds, and we have no information about any unusual property associated to sparrows. As a consequence, reasoning on the base of the principle of 'presumption of typicality' (see Section 3.2), we would like sparrows to inherit all the typical properties of birds. In fact, we have:

Triple $\langle s, \mathsf{sc}, f \rangle$**.** We apply the procedure $\mathtt{DefMinEntailmentC}(H, r(H), \langle s, \mathsf{sc}, f \rangle)$ and get: $G' \nvdash_{\rho df_\perp} (s, \perp_c, s)$ for $i = 0$, that implies $\mathrm{D}^p = \{\langle b, \mathsf{sc}, f \rangle, \langle b, \mathsf{sc}, hf \rangle\}$. Now, $G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (s, \mathsf{sc}, f)$, since $(b, \mathsf{sc}, f) \in (\mathrm{D}^p)^s$ and $(s, \mathsf{sc}, b) \in G^{str}$. Therefore,

$$\mathtt{DefMinEntailmentC}(H, r(H), \langle s, \mathsf{sc}, f \rangle) = \mathtt{true} \ .$$

Triple $\langle s, \mathsf{sc}, hf \rangle$**.** We apply the procedure $\mathtt{DefMinEntailmentC}(H, r(H), \langle s, \mathsf{sc}, hf \rangle)$ and get: $G' \nvdash_{\rho df_\perp} (s, \perp_c, s)$ for $i = 0$. $G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (s, \mathsf{sc}, hf)$, since $(b, \mathsf{sc}, hf) \in (\mathrm{D}^p)^s$ and $(s, \mathsf{sc}, b) \in G^{str}$, it follows that

$$\mathtt{DefMinEntailmentC}(H, r(H), \langle s, \mathsf{sc}, hf \rangle) = \mathtt{true} \ .$$

Finally, we check what happens with an extra exceptional level. Do penguins with jet-packs fly or not?

Triple $\langle pj, \mathsf{sc}, f \rangle$**.** We apply the procedure $\mathtt{DefMinEntailmentC}(H, r(H), \langle pj, \mathsf{sc}, f \rangle)$ and have: $G' \nvdash_{\rho df_\perp} (pj, \perp_c, pj)$ for $i = 2$, that implies $\mathrm{D}^p = \{\langle pj, \mathsf{sc}, f \rangle\}$. $G^{str} \cup (\mathrm{D}^p)^s \vdash_{\rho df_\perp} (pj, \mathsf{sc}, f)$, since $(\mathrm{D}^p)^s = \{(pj, \mathsf{sc}, f)\}$, and

$$\mathtt{DefMinEntailmentC}(H, r(H), \langle pj, \mathsf{sc}, f \rangle) = \mathtt{true} \ .$$

Triple $\langle pj, \mathsf{sc}, e \rangle$**.** We apply the procedure $\mathtt{DefMinEntailmentC}(H, r(H), \langle pj, \mathsf{sc}, e \rangle)$ and have:
$G^{str} \cup (\mathrm{D}^p)^s \nvdash_{\rho df_\perp} (pj, \mathsf{sc}, e)$, and

$$\mathtt{DefMinEntailmentC}(H, r(H), \langle pj, \mathsf{sc}, e \rangle) = \mathtt{false} \ .$$

Therefore, correctly RC does not allow penguins with jet-packs to inherit the property of not-flying from typical penguins. $\quad\square$

The next example shows a case in which we have some information with infinite rank, and how that indicates the presence of some conflict.

Example 3.13. Let the graph L contain the following information

$$L = \{(b, \mathsf{sc}, ba), (mb, \mathsf{sc}, b), (ba, \perp_c, bw)\langle mb, \mathsf{sc}, bw\rangle\} \ ,$$

where ba is read "breaths air", bw is read "breaths underwater", and mb is the class "marsh bird".

When we apply the $\mathtt{Ranking}(L)$ procedure, as

$$L^s \vdash_{\rho df_\perp} (mb, \perp_c, mb)$$

we obtain $\mathrm{D}_0 = \mathrm{D}_1 = \mathrm{D}_\infty = \{\langle mb, \mathsf{sc}, bw\rangle\}$. Now, we can check, using the $\mathtt{StrictMinEntailment}$ procedure, that *e.g.*, L minimally entails $(mb, \perp_c, mb)$: since $\langle mb, \mathsf{sc}, bw\rangle \in \mathrm{D}_\infty$, we have $(mb, \perp_c, mb) \in L'$, that obviously implies

$$L' \vdash_{\rho df_\perp} (mb, \perp_c, mb) \ .$$

Therefore, as expected,

$$\mathtt{StrictMinEntailment}(L, r(L), (mb, \perp_c, mb)) = \mathtt{true} \ . \quad\square$$

The outcomes in Example 3.13 are reasonable and desirable: despite we are dealing with defeasible information, we are facing an unsolvable conflict: we are informed that birds breath air, (b, sc, ba), without exceptions, since it is a strict $\rho df_\perp$-triple, while marsh birds usually breath underwater, $\langle mb, \mathsf{sc}, bw\rangle$. The triple (b, sc, ba), not being defeasible, does not allow the existence of birds breathing underwater, and from the information at our disposal it is reasonable to conclude that marsh birds cannot exist, that is, $(mb, \perp_c, mb)$.

Triples with infinite rank appear when there is some unsolvable conflict in the graph. Another example comes when we have in the graph pieces of information that are in direct conflict with each other. For example, if we add to the graph H in Example 3.12 the two defeasible triples $\langle mb, \mathsf{sc}, f \rangle, \langle mb, \mathsf{sc}, e \rangle$ ('marsh birds typically fly' and 'marsh birds typically do not fly'), that are in direct conflict with each other, we end up again concluding $(mb, \perp_c, mb)$.

3.6. Structural properties

RC, like many other non-monotonic approaches, has also been analysed from a 'structural properties' point of view [37]. For example, in the propositional case, given a knowledge base $\mathcal{K}$ of defeasible conditionals $\alpha \rightsquigarrow \beta$ (read 'if α holds, typically β holds too'), where α and β are propositions, RC satisfies a particular form of constrained monotonicity, called *Rational Monotonicity* (RM) [35]:

$$(RM) \quad \frac{\mathcal{K} \vDash \alpha \rightsquigarrow \beta, \quad \mathcal{K} \not\vdash \alpha \rightsquigarrow \neg\gamma}{\mathcal{K} \vDash \alpha \wedge \gamma \rightsquigarrow \beta}$$

The intended meaning of an instance of the structural property above is the following: if we know that typical birds fly ($\mathcal{K} \vDash \alpha \rightsquigarrow \beta$), and we are not aware that all typical birds are not black ($\mathcal{K} \not\vdash \alpha \rightsquigarrow \neg\gamma$), then we may conclude that, typically, black birds fly ($\mathcal{K} \vDash \alpha \wedge \gamma \rightsquigarrow \beta$).

In our framework, a propositional defeasible conditional $\alpha \rightsquigarrow \beta$ correspond to defeasible triples of the form $\langle p, \mathrm{sc}, q\rangle$ and $\langle p, \mathrm{sp}, q\rangle$. Therefore, given a graph G, the property (RM) may take a form like[18]

$$(RM) \quad \frac{G \vDash_{\min} \langle p, \mathrm{sc}, q\rangle, \quad G \not\vDash_{\min} \langle p, \mathrm{sc}, \neg r\rangle}{G \vDash_{\min} \langle p \wedge r, \mathrm{sc}, q\rangle}$$

Such a property is linked to the use of conjunction and negation of terms, which, however, are not supported (so far) in $\rho df_\perp$. It does not seem possible to express (RM) in defeasible $\rho df_\perp$.

However, there are a few basic structural properties that still can be expressed in our framework. One of the simplest one is the property of *Supraclassicality*, as it is called in the propositional setting, that simply indicates that a strict piece of information implies also its own defeasible, weaker formulation, that is,

$$(Supra_c) \quad \frac{G \vDash_{\min} (p, \mathrm{sc}, q)}{G \vDash_{\min} \langle p, \mathrm{sc}, q\rangle} \qquad (Supra_p) \quad \frac{G \vDash_{\min} (p, \mathrm{sp}, q)}{G \vDash_{\min} \langle p, \mathrm{sp}, q\rangle}$$

Proposition 3.30. $\vDash_{\min}$ *satisfies* $(Supra_c)$ *and* $(Supra_p)$.

Proof. Consider $(Supra_c)$, and let $G \vDash_{\min} (p, \mathrm{sc}, q)$. That implies that all $\rho df_\perp$-interpretations in the minimal model $\mathcal{R}_{\min G}$ satisfy (p, sc, q). Consequently, for every $\rho df_\perp$-interpretation $\mathcal{I}$ in $\mathrm{c\_min}(p, \mathcal{R}_{\min G})$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$. That is, according to Definition 3.4, $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q\rangle$, that is to $G \vDash_{\rho df_\perp} \langle p, \mathrm{sc}, q\rangle$. The proof $(Supra_p)$ is analogous. $\square$

Reflexivity, Left Logical Equivalence, and *Right Weakening* are other essential properties. In the propositional case they are of the form:

$$(Ref) \quad \mathcal{K} \vDash \alpha \rightsquigarrow \alpha$$

$$(LLE) \quad \frac{\mathcal{K} \vDash \alpha \rightsquigarrow \beta, \quad \vDash \alpha \equiv \gamma}{\mathcal{K} \vDash \gamma \rightsquigarrow \beta} \qquad (RW) \quad \frac{\mathcal{K} \vDash \alpha \rightsquigarrow \beta, \quad \beta \vDash \gamma}{\mathcal{K} \vDash \alpha \rightsquigarrow \gamma}$$

As for *Supraclassicality*, these properties can also be translated in our system in two versions: one for classes and one for predicates. For (LLE), logical equivalence '$\equiv$' is translated using symmetric pairs of sc- and sp-triples. Specifically, the above axioms are encoded as

$$(Ref_c) \quad G \vDash_{\min} \langle p, \mathrm{sc}, p\rangle$$

$$(LLE_c) \quad \frac{G \vDash_{\min} \langle p, \mathrm{sc}, r\rangle, \quad G \vDash_{\min} (p, \mathrm{sc}, q), \quad G \vDash_{\min} (q, \mathrm{sc}, p)}{G \vDash_{\min} \langle q, \mathrm{sc}, r\rangle} \qquad (RW_c) \quad \frac{G \vDash_{\min} \langle p, \mathrm{sc}, q\rangle, \quad G \vDash_{\min} (q, \mathrm{sc}, r)}{G \vDash_{\min} \langle p, \mathrm{sc}, r\rangle}$$

$$(Ref_p) \quad G \vDash_{\min} \langle p, \mathrm{sp}, p\rangle$$

$$(LLE_p) \quad \frac{G \vDash_{\min} \langle p, \mathrm{sp}, r\rangle, \quad G \vDash_{\min} (p, \mathrm{sp}, q), \quad G \vDash_{\min} (q, \mathrm{sp}, p)}{G \vDash_{\min} \langle q, \mathrm{sp}, r\rangle} \qquad (RW_p) \quad \frac{G \vDash_{\min} \langle p, \mathrm{sp}, q\rangle, \quad G \vDash_{\min} (q, \mathrm{sp}, r)}{G \vDash_{\min} \langle p, \mathrm{sp}, r\rangle}$$

Note that (Ref_c) and (Ref_p) do not hold, as they do not hold in the classical form $G \vDash_{\min} (p, \mathrm{sc}, p)$ and $G \vDash_{\min} (p, \mathrm{sp}, p)$. This is the consequence of having considered minimal ρdf for which reflexivity for sc and sp triples does not hold. If we use (full) ρdf, we will recover reflexivity for the classical triples, and, by Proposition 3.30, we would obtain immediately also (Ref_c) and (Ref_p).

Concerning (LLE) and (RW), they are satisfied.

Proposition 3.31. $\vDash_{\min}$ *satisfies* (LLE_c), (RW_c), (LLE_p), *and* (RW_p).

---

[18] Informally, the extension of $\neg r$ is the complementary of the extension of r over Δ_C.

Remark 3.4. Let us note that, despite (RM) seems not to be syntactically expressible in our framework, we have proposed the same kind of construction that is behind propositional RC, that is, we model a kind of defeasible reasoning implementing the *Presumption of Typicality* [34, p. 4]: if we are not informed of the contrary, we reason assuming that we are dealing with typical behaviours. From a semantic point of view, this "maximisation of typicality" has been modelled considering the *minimal models* of a KB: those models in which the entities are associated to the lowest (*i.e.*, most typical) possible rank value, modulo the satisfaction of the knowledge base. This formal solution has been used also elsewhere in order to give a semantic characterisation of RC [9,26,40]. On the other hand, from the decision procedure point of view, we have defined an algorithm that follows similar ones developed for RC in other formal frameworks; in particular, our decision procedure is built on top of the decision procedure for the monotonic fragment, and it decides RC through a calculation of exceptionalities and rank values, as it is done *e.g.*, for RC for propositional logic and DLs [12,16,21].

Nevertheless, there are some more properties satisfied by our entailment relation $\vDash_{\min}$. In the following, let us consider the closure operation $\mathsf{Cl}_{\min}$ defined as follows: given a graph G,

$$\mathsf{Cl}_{\min}(G) := \{\,[\![s,p,o]\!] \mid G \vDash_{\min} [\![s,p,o]\!]\,\}.$$

As we may expect, $\mathsf{Cl}_{\min}$ is not monotonic, that is, given any graph G and triple $[\![s,p,o]\!]$, the following does not necessarily hold[19]:

$$\mathsf{Cl}_{\min}(G) \subseteq \mathsf{Cl}_{\min}(G \cup \{[\![s,p,o]\!]\})\,.$$

Proposition 3.32. $\mathsf{Cl}_{\min}$ *is not monotonic.*

Proof. Consider the graph F in Example 3.5, and its subgraph

$$F' = \{(p,\mathsf{sc},b),(s,\mathsf{sc},b),\langle b,\mathsf{sc},f\rangle,(e,\perp_{\mathsf{c}},f)\}\,,$$

obtained eliminating the triple (p,sc,e).

It can be easily shown that w.r.t. F' penguins are not exceptional, *i.e.*,

$$\mathtt{DefMinEntailmentP}(F',r(F'),\langle p,\mathsf{sc},f\rangle) = \mathtt{true}\,,$$

while w.r.t. F penguins are exceptional, *i.e.*,

$$\mathtt{DefMinEntailmentP}(F,r(F),\langle p,\mathsf{sc},f\rangle) = \mathtt{false}\,. \quad \square$$

Nonetheless, and not surprising, $\mathsf{Cl}_{\min}(G)$ is monotonic w.r.t. the strict part of the information.

Proposition 3.33. *Let G be any graph, (s,p,o) any $\rho df_\perp$-triple, and $[\![s',p',o']\!]$ be any triple. If $G \vDash_{\min} (s,p,o)$, then $G \cup \{[\![s',p',o']\!]\} \vDash_{\min} (s,p,o)$.*

Proof. This is immediate from Theorem 3.22. Indeed, $G \vDash_{\min} (s,p,o)$ implies $G' \vdash_{\rho df_\perp} (s,p,o)$, with G' defined according to the procedure $\mathtt{StrictMinEntailment}$. Let $F := G \cup \{[\![s',p',o']\!]\}$, and let F' be defined applying the procedure $\mathtt{StrictMinEntailment}$ to F. Clearly $G' \subseteq F'$, and, given that $\vdash_{\rho df_\perp}$ is monotonic, we can conclude $F' \vdash_{\rho df_\perp} (s,p,o)$, that is, $F \vDash_{\min} (s,p,o)$. $\quad \square$

This is clearly a desirable behaviour, since the strict part of our information should be treated as non-defeasible information, and consequently we should reason monotonically about it.

$\mathsf{Cl}_{\min}$ satisfies some other desirable properties: namely, *Inclusion, Cumulativity*, and *Idempotence*.

Proposition 3.34. $\mathsf{Cl}_{\min}$ *satisfies inclusion. That is, for any graph G*

$$G \subseteq \mathsf{Cl}_{\min}(G)\,.$$

Proof. $\mathsf{Cl}_{\min}(G)$ is determined by the model $\mathcal{R}_{\min G}$, and, by definition, $\mathcal{R}_{\min G} \in \mathfrak{R}_G$, that is, $\mathcal{R}_{\min G}$ is a model of G, implying $G \subseteq \mathsf{Cl}_{\min}(G)$. $\quad \square$

Proposition 3.35. $\mathsf{Cl}_{\min}$ *satisfies Cumulativity. That is, for any pair of graphs G, G',*

if $G \subseteq G' \subseteq \mathsf{Cl}_{\min}(G)$, then $\mathsf{Cl}_{\min}(G') = \mathsf{Cl}_{\min}(G)$.

---

[19] Monotonicity at the level of the entailment relation is sometimes called *semi-monotonicity* in the literature on non-monotonic reasoning [40].

Proof. $Cl_{min}(G)$ is determined by the model $\mathcal{R}_{\min G}$, that is, the minimal model of G. Since $G' \subseteq Cl_{min}(G)$, $\mathcal{R}_{\min G}$ is also a model of G'. Therefore, $\mathcal{R}_{\min G}$ must be the minimal model also for G'. Indeed, assume that is not the case, that is, there is a model R of G' such that $R \preceq \mathcal{R}_{\min G}$. Since $G \subseteq G'$, R is also a model of G, and $\mathcal{R}_{\min G}$ would not be the minimal model of G, against the hypothesis. Hence both $Cl_{min}(G)$ and $Cl_{min}(G')$ are determined by the model $\mathcal{R}_{\min G}$. □

Cumulativity has two immediate consequences that are well-known, desirable properties: a constrained form of monotonicity, called *Cautious Monotonicity*, and the classical property of *Cut*.

Proposition 3.36. Cl_{min} *satisfies* Cautious Monotonicity. *That is, for any graph G and any triple $[\![s, p, o]\!]$,*

$$\text{if } G \vDash_{\min} [\![s, p, o]\!] \text{ and } G \vDash_{\min} [\![s', p', o']\!], \text{ then } G \cup [\![s', p', o']\!] \vDash_{\min} [\![s, p, o]\!] .$$

Proposition 3.37. Cl_{min} *satisfies* Cut. *That is, for any graph G and any triple $[\![s, p, o]\!]$,*

$$\text{if } G \cup [\![s', p', o']\!] \vDash_{\min} [\![s, p, o]\!] \text{ and } G \vDash_{\min} [\![s', p', o']\!], \text{ then } G \vDash_{\min} [\![s, p, o]\!] .$$

Please note that these two properties are here analysed at the level of entailment. As discussed above, in the case of the conditional reasoning [35] these structural properties can be expressed at two levels: at the level of conditionals (see the first formulation of (RM) above), and the meta-level of the entailment relation. Here we are looking at the properties (CM) and (Cut) at the meta-level of the entailment relation. Similarly to the (RM) case, if conjunction among terms would have been allowed, such properties at the level of the language may be expressed as

$$(CM) \quad \frac{G \vDash_{\rho df_\perp} \langle p, \text{sc}, q \rangle, \quad G \vDash_{\rho df_\perp} \langle p, \text{sc}, r \rangle}{G \vDash_{\rho df_\perp} \langle p \wedge r, \text{sc}, q \rangle} \qquad (Cut) \quad \frac{G \vDash_{\rho df_\perp} \langle p \wedge r, \text{sc}, q \rangle, \quad G \vDash_{\rho df_\perp} \langle p, \text{sc}, r \rangle}{G \vDash_{\rho df_\perp} \langle p, \text{sc}, q \rangle}$$

Finally, Cl_{min} satisfies also *Idempotence*.

Proposition 3.38. Cl_{min} *satisfies* Idempotence. *That is, for any graph G,*

$$Cl_{min}(Cl_{min}(G)) = Cl_{min}(G) .$$

Proof. *Idempotence* is an immediate consequence of *Inclusion* and *Cumulativity*. It is sufficient to set $G' = Cl_{min}(G)$ in the Cumulativity property, while Inclusion guarantees that $G \subseteq Cl_{min}(G)$ holds. □

The satisfaction of such properties, defined at the level of entailment and also called *global properties* by Lehmann and Magidor, was already proved for the original formulation of RC in the propositional case [35, Sect. 5.5].

3.7. Computational complexity

We now address the computational complexity of the previously defined procedures and show that our entailment decision procedures run in polynomial time.

To start with, let us consider $\vdash_{\rho df_\perp}$. Now, consider a graph G and a $\rho df_\perp$-triple (s, p, o). An easy way to decide whether $G \vdash_{\rho df_\perp} (s, p, o)$ holds is to compute the closure $Cl(G)$ of G and then check whether $(s, p, o) \in Cl(G)$. Now, as G is ground, like for [39], it is easily verified that, both the time to compute the closure of G as well as its size are $O(|G|^2)$ and, thus,

Proposition 3.39. *For a graph G and a $\rho df_\perp$-triple (s, p, o), $G \vdash_{\rho df_\perp} (s, p, o)$ can be decided in time $O(|G|^2)$.*

Remark 3.5. Let us note that [39, Theorem 21] provides also an $O(|G| \log |G|)$ time algorithm to decide the ground ρdf entailment problem.[20] Whether a similar algorithm can be extended also to $\rho df_\perp$ (so including also rules (5)-(7)) while maintaining the same computational complexity is still an open problem.

We next consider the ExceptionalC and ExceptionalP procedures. It is immediately verified that, by Proposition 3.39,

Proposition 3.40. *For a defeasible graph $G = G^{str} \cup G^{def}$, both ExceptionalC(G) as well as ExceptionalP(G) require at most $|G^{def}|$ checks and, thus, both run in time $O(|G^{def}||G|^2)$.*

Consider now the case of the Ranking procedure. It is easily verified that steps 3.-6. may be repeated at most $|G^{def}|$ times and each of which calls once the ExceptionalC and ExceptionalP procedures. Therefore, by Proposition 3.40, we have easily that

---

20 It corresponds here to consider rules (1)-(4).

Proposition 3.41. *For a defeasible graph $G = G^{str} \cup G^{def}$,* $\mathtt{Ranking}(G)$ *runs in time $O(|G^{def}|^2|G|^2)$. Moreover, the number of sets in $\mathtt{r}(G)$ is at most $O(|G^{def}|)$.*

Eventually, let us consider the entailment check procedures in Section 3.5. Let us recall Remark 3.2 and, thus, we assume that the ranking has been computed once and for all. The time required to compute $\mathtt{r}(G)$ is, by Proposition 3.41, $O(|G^{def}|^2|G|^2)$.

To what concerns $\mathtt{StrictMinEntailment}$, the following result is an immediate consequence from Proposition 3.39.

Proposition 3.42. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and a $\rho df_\perp$-triple (s, p, o). Then the procedure $\mathtt{StrictMinEntailment}(G, \mathtt{r}(G), (s, p, o))$ runs in time $O(|G|^2)$.*

Now, consider $\mathtt{DefMinEntailmentC}$. By Proposition 3.41, it is easily verified that the steps 3.-10. may be repeated at most $O(|G^{def}|)$ times and each time we make one $\vdash_{\rho df_\perp}$ check. By considering also the additional $\vdash_{\rho df_\perp}$ check in step 12, by Proposition 3.39 we have

Proposition 3.43. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and a defeasible triple $\langle p, \mathtt{sc}, q \rangle$. Then the procedure $\mathtt{DefMinEntailmentC}(G, \mathtt{r}(G), \langle p, \mathtt{sc}, q \rangle))$ runs in time $O(|G^{def}||G|^2)$.*

The computational complexity of $\mathtt{DefMinEntailmentP}$ is the same as for $\mathtt{DefMinEntailmentC}$ and, thus, we conclude with

Proposition 3.44. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and a defeasible triple $\langle p, \mathtt{sp}, q \rangle$. Then the procedure $\mathtt{DefMinEntailmentP}(G, \mathtt{r}(G), \langle p, \mathtt{sc}, q \rangle))$ runs in time $O(|G^{def}||G|^2)$.*

From Propositions 3.39 - 3.44, it follows immediately that

Corollary 3.45. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and a triple $[\![p, *, q]\!]$. Then $G \vDash_{\min} [\![p, *, q]\!]$ can be decided in polynomial time.*

4. Inheritance networks based closure

As we have seen in Remark 3.3 and Example 3.12, RC suffers from the so-called drowning problem: if a class is exceptional w.r.t. a superclass it does not inherit *any* of the typical property of the superclass (e.g., we can neither derive that "drug users are usually students" nor that "penguins usually have feathers"). Such an inferential behaviour is considered a weakness in many applications. In order to overcome such a limit, some closure operations extending RC have been proposed: for example, the Lexicographic Closure [13,34,36], the Defeasible Inheritance-based approach [14], the Relevant Closure [15], and the Multipreference Closure [23].

Here we propose an adaptation of the Defeasible Inheritance-based approach [14], originally formulated for propositional logic and DLs, to the RDFS framework. We focus on such an approach for two reasons: it is inspired by inheritance nets [48], hence it is based on a formalism that is already particularly close to RDFS; and, it is promising from the computational complexity point of view, as we have already seen in [14] and in the case of the DL $\mathcal{EL}_\perp$ [16, Section 4].

The approach is based on refining RC by using the graph to identify the axioms taking part in each specific conflict between pieces of information. As we are going to see, such a solution will allow us both to overcome the drowning effect and to preserve computational tractability.[21] We also point out that, at the time of writing, we did not find a tractable procedure to decide defeasible subsumption under other approaches such as Lexicographic Closure, Relevant Closure, or Multipreference closure mentioned above. However, some tractable results are known for some specific syntactic cases under Lexicographic Closure. In fact, [18] shows that, while rational closure and lexicographic entailment share the same complexity for deciding entailment of propositional conditional knowledge bases (they are both P^{NP}-complete), when moving to the Horn-case, the complexity of the two approaches diverge in the sense that the former is P-complete, while the latter still remains P^{NP}-complete. [18] also shows that for so-called feedback-free Horn conditional KBs[22] lexicographic Horn entailment is polynomial. Nevertheless, we leave the option to restrict syntactically defeasible RDFS graphs under lexicographic closure (or any other before mentioned closure operation) and how these restrictions impact computationally to future work.

4.1. Decision procedures for inheritance net based closure

In the Defeasible Inheritance-based approach [14,16] the pieces of information in the KB are translated into an inheritance net, which extends the work of Touretzky [48]. Since RDFS is already a graphical formalism, such a translation is not necessary in our framework. Also, in [14,16] we have introduced the notion of *duct* on a graph, which is a generalisation of the classical notion of

---

[21] In [14, Appendix A] it is also shown that the inheritance-base closure behaves well, and better than RC w.r.t. most of the "benchmark" examples.

[22] Very roughly, feedback-free Horn conditional KBs are such that the consequent of a defeasible Horn rule cannot occur in a strict Horn-rule, and we can define a particular partition, the *default partition*, of the set of the defeasible Horn rules based on the vocabulary (see [18, Sect. 6.4.2] for a detailed explanation).

path that allows the formalisation of conjunctions and disjunctions, which we do not need here. In fact, we can revert to the notion of *paths* only, defined inductively as follows.[23]

Definition 4.1 *(path).* Given a graph $G = G^{str} \cup G^{def}$, the *paths* in G are defined as follows:

1. every triple $[\![p, *, q]\!]$ in G corresponds to a path $\pi = \{[\![p, *, q]\!]\}$ in G from p to q;
2. if π is a path from p to q and $[\![q, *, s]\!]$ is a triple in G that does not already occur in π, then $\pi' = \pi \cup \{[\![q, *, s]\!]\}$ is a path in G from p to s.

Using the notion of path we apply the RC procedure locally: that is, if we want to decide whether $\langle p, \mathrm{sc}, q \rangle$ or $\langle p, \mathrm{sp}, q \rangle$ are entailed by a graph, the exceptionality rankings and the RC are calculated considering only the information in the KB that has some connection to p and q (for more details, we refer the reader to [14]): we consider only the defeasible triples in G^{def} corresponding to triples appearing in the paths from p to q, and we compute the RC of only such a portion of the KB.

For the rest of this section, we will assume that we will be working with graphs $G = G^{str} \cup G^{def}$ s.t. G^{str} is closed under StrictMinEntailment, that is, for every $\rho df_\perp$ triple (s, p, o), $(s, p, o) \in G^{str}$ iff StrictMinEntailment$(G, \mathrm{Ranking}(G), (s, p, o))$ returns true.

Now, let $G = G^{str} \cup G^{def}$ be a graph, and consider any triple $\langle p, *, q \rangle$ with $* \in \{\mathrm{sc}, \mathrm{sp}\}$. The procedure for deciding whether $\langle p, *, q \rangle$ is in the closure of a graph G using the Defeasible Inheritance-based approach is as follows [14].

First, we need to proceed with a first closure operation, that needs to be done once and for all: set $G_{in} := G$; then, for every pair of nodes p, q in the graph do

Step 1. Compute the set of all paths starting in p and ending into q, and let $\Delta_{p,q}$ be the set of all the defeasible triples $\langle t, *, z \rangle$ (with $* \in \{\mathrm{sc}, \mathrm{sp}\}$) appearing in such paths.

Step 2. Let $G_{p,q} = G^{str} \cup \Delta_{p,q}$.

Step 3. If $G_{p,q} \models_{\min} \langle p, *, q \rangle$, let $G_{in} := G_{in} \cup \{\langle p, *, q \rangle\}$.

The procedure InheritanceCompletion describes the pseudo-algorithm corresponding to such steps.

Procedure InheritanceCompletion(G).

Input: Graph $G = G^{str} \cup G^{def}$ with G^{str} closed under StrictMinEntailment
Output: Graph G_{in}

1:   $G_{in} := G$
2:   **for all** $\langle p, q \rangle$, s.t. $p, q \in \mathrm{uni}(G)$ **do**
3:      $\Delta_{p,q} := \bigcup \{\sigma \mid \sigma \text{ is a path in } G \text{ from } p \text{ to } q\} \setminus G^{str}$
4:      $G_{p,q} := G^{str} \cup \Delta_{p,q}$
5:      **if** DefMinEntailmentC$(G_{p,q}, \mathrm{Ranking}(G_{p,q}), \langle p, \mathrm{sc}, q \rangle) = \text{true}$ **then**
6:         $G_{in} := G_{in} \cup \langle p, \mathrm{sc}, q \rangle$
7:      **if** DefMinEntailmentP$(G_{p,q}, \mathrm{Ranking}(G_{p,q}), \langle p, \mathrm{sp}, q \rangle) = \text{true}$ **then**
8:         $G_{in} := G_{in} \cup \langle p, \mathrm{sp}, q \rangle$
9:   **return** G_{in}

The expansion of the graph G into the graph G_{in} adds to the graph defeasible connections that are not involved into conflicts, but that would have been lost in RC due to the drowning problem. Example 4.1 below will show it.

Once we have avoided the drowning effect, we define the inheritance based entailment (that we will indicate with the symbol $\vdash_{in}$) by closing G_{in} under RC. That is:

Definition 4.2 *(Inheritance-based entailment).* For any graph $G = G^{str} \cup G^{def}$ and any triple $[\![p, r, q]\!]$,

$$G \vdash_{in} [\![p, r, q]\!] \text{ iff } G_{in} \models_{\min} [\![p, r, q]\!] . \tag{9}$$

Given Theorems 3.25 and 3.29, Equation (9) can be reformulated as follows:

$$G \vdash_{in} (p, r, q) \text{ iff StrictMinEntailment(InheritanceCompletion}(G),$$

$$\mathrm{Ranking(InheritanceCompletion}(G)), (p, r, q)) = \text{true};$$

$$G \vdash_{in} \langle p, \mathrm{sc}, q \rangle \text{ iff DefMinEntailmentC(InheritanceCompletion}(G), \tag{10}$$

$$\mathrm{Ranking(InheritanceCompletion}(G)), \langle p, \mathrm{sc}, q \rangle) = \text{true};$$

$$G \vdash_{in} \langle p, \mathrm{sp}, q \rangle \text{ iff DefMinEntailmentP(InheritanceCompletion}(G),$$

---

[23] We use Greek letters $\pi, \sigma, \dots$ to denote paths.

$$\text{Ranking}(\text{InheritanceCompletion}(G)), \langle p, \mathrm{sp}, q \rangle) = \texttt{true} \ .$$

That is, given a graph G, if we have as query, respectively, a strict triple, a defeasible sc-triple, or a defeasible sp-triple, $\vdash_{in}$ is determined by calling, respectively, `StrictMinEntailment`, `DefMinEntailmentC`, and `DefMinEntailmentP`.

Example 4.1 (*Running example cont.*). We have already seen in Example 3.12 that `DefMinEntailmentC` and `DefMinEntailmentP` suffer from the drowning effect. This also holds for Example 3.10 (see also Remark 3.3). In particular, in Example 3.10 we have seen that, despite young people are usually happy, young drug users are atypical young people that are usually unhappy: *i.e.*, $G \models_{\min} \langle yDU, \mathrm{sc}, uhP \rangle$ and $G \not\models_{\min} \langle yDU, \mathrm{sc}, hP \rangle$. Now, the exceptionality of the class of young drug users w.r.t. the class of young people implies that, due to the drowning effect, young drug users do not inherit any of the defeasible properties associated to young people, in particular we will not be able to derive that typically young drug users are students, that is, $G \not\models_{\min} \langle yDU, \mathrm{sc}, s \rangle$.

We can easily check this. As we have seen in Example 3.10, the defeasible set associated to the query $\langle yDU, \mathrm{sc}, uhP \rangle$ is

$$\mathrm{D}^{yDU} = \{\langle dU, \mathrm{sc}, uhP \rangle, \langle dU, \mathrm{sc}, yP \rangle\} \ .$$

As the defeasible set associated to a query triple depends on its first element, D^{yDU} is associated also to the query $\langle yDU, \mathrm{sc}, s \rangle$. Now, it is easy to check that

$$G^{str} \cup (\mathrm{D}^{yDU})^s \not\vdash_{\rho df_\perp} \langle yDU, \mathrm{sc}, s \rangle$$

as $\langle yP, \mathrm{sc}, s \rangle$ is not in D^{yDU}. As a consequence $G \not\models_{\min} \langle yDU, \mathrm{sc}, s \rangle$.

On the other hand, using $\vdash_{in}$ we avoid the drowning effect, as shown next. Our query triple is $\langle yDU, \mathrm{sc}, s \rangle$. First of all we have to check whether it is in `InheritanceCompletion`(G). Consider all the paths from yDU to s: *i.e.*,

$$\sigma = \{(yDU, \mathrm{sc}, yP), \langle yP, \mathrm{sc}, s \rangle\}$$

$$\sigma' = \{(yDU, \mathrm{sc}, dU), \langle dU, \mathrm{sc}, yP \rangle, \langle yP, \mathrm{sc}, s \rangle\} \ .$$

That is,

$$\Delta_{(yDU,S)} := (\sigma \cup \sigma') \setminus G^{str} = \{\langle dU, \mathrm{sc}, yP \rangle, \langle yP, \mathrm{sc}, s \rangle\}$$

and

$$G_{(yDU,s)} = G^{str} \cup \Delta_{(yDU,s)} \ .$$

Now we have to check whether $G_{(yDU,s)} \models_{\min} \langle yDU, \mathrm{sc}, s \rangle$. We do so by invoking

$$\texttt{DefMinEntailmentC}(G_{(yDU,s)}, \texttt{Ranking}(G_{(yDU,s)}), \langle yDU, \mathrm{sc}, s \rangle) \ .$$

First of all, we need to execute `Ranking`($G_{(yDU,s)}$). As

$$(G_{(yDU,s)})^s \not\vdash_{\rho df_\perp} (dU, \perp_\mathrm{c}, dU)$$

$$(G_{(yDU,s)})^s \not\vdash_{\rho df_\perp} (yP, \perp_\mathrm{c}, yP) \ ,$$

we obtain $r(G_{(yDUr,s)}) = \{\mathrm{D}^0, \mathrm{D}^\infty\}$ with

$$\mathrm{D}^0 = \{\langle yP, \mathrm{sc}, s \rangle, \langle dU, \mathrm{sc}, yP \rangle\} \ ,$$

and $\mathrm{D}^\infty = \emptyset$. That is, $G_{(yDU,s)}$ is a subgraph of G without any exceptionality. It is now easy to check that the defeasible set associated to the query triple $\langle yDU, \mathrm{sc}, s \rangle$ is

$$\mathrm{D}^{yDU} = \mathrm{D}^0$$

and we obtain

$$G^{str} \cup (\mathrm{D}^0)^s \vdash_{\rho df_\perp} \langle yDU, \mathrm{sc}, s \rangle \ ,$$

as we have (yDU, sc, yP) and (yP, sc, s) in $G^{str} \cup (\mathrm{D}^0)^s$. That is,

$$G_{(yDU,s)} \models_{\min} \langle yDU, \mathrm{sc}, s \rangle \ .$$

Therefore, $\langle yDU, \mathrm{sc}, s \rangle \in$ `InheritanceCompletion`(G), that is, $\langle yDU, \mathrm{sc}, s \rangle \in G_{in}$. Given that $G \vdash_{in} \langle yDU, \mathrm{sc}, s \rangle$ iff $G_{in} \models_{\min} \langle yDU, \mathrm{sc}, s \rangle$ or, equivalently,

$$\langle yDU, \mathrm{sc}, s \rangle \in \texttt{DefMinEntailmentC}(\texttt{InhertanceCompletion}(G)) \ ,$$

$\langle yDU, \mathrm{sc}, s \rangle \in G_{in}$, and $\models_{\min}$ satisfies Inclusion (Proposition 3.34), we can conclude that

$$G \vdash_{in} \langle yDU, \mathsf{sc}, s \rangle \ ,$$

as desired.

On the other hand, it is easy to check that in case of conflicting information the inheritance-based approach and rational closure behave alike.

To see this, we can check whether the triple $\langle yDU, \mathsf{sc}, uhP \rangle$ follows from G, as in Example 3.10. The paths from yDU to uhP are:

$$\sigma = \{ (yDU, \mathsf{sc}, yP), \langle yP, \mathsf{sc}, hP \rangle, (hP, \perp_c, uhP) \}$$

$$\sigma' = \{ (yDU, \mathsf{sc}, dU), \langle dU, \mathsf{sc}, yP \rangle, \langle yP, \mathsf{sc}, hP \rangle, (hP, \perp_c, uhP) \}$$

$$\sigma'' = \{ (yDU, \mathsf{sc}, dU), \langle dU, \mathsf{sc}, uhP \rangle \} \ .$$

Therefore,

$$\Delta_{(yDU, uhP)} := (\sigma \cup \sigma' \cup \sigma'') \setminus G^{str} = \{ \langle dU, \mathsf{sc}, yP \rangle, \langle yP, \mathsf{sc}, hP \rangle, \langle dU, \mathsf{sc}, uhP \rangle \}$$

and

$$G_{(yDU, uhP)} = G^{str} \cup \Delta_{(yDU, uhP)} \ .$$

Now we check whether $G_{(yDU, uhP)} \vDash_{\min} \langle yDU, \mathsf{sc}, uhP \rangle$ holds. At first, we rank $\Delta_{(yDU, uhP)}$ via $\mathtt{Ranking}(G_{(yDU, uhP)})$. We have:

$$\mathtt{D}_0 := \Delta_{(yDU, uhP)} \ .$$

We need to check whether dU, yP are 'empty classes' in $(G_{(yDU, uhP)})^s$. Indeed, as for G^s in Example 3.10, we obtain $(G_{(yDU, uhP)})^s \vdash_{\rho df_\perp}$ $(dU, \perp_c, dU)$, while yP does not turn out to be empty in $(G_{(yDU, uhP)})^s$. Consequently, $\mathtt{ExceptionalC}(G^{str} \cup \mathtt{D}_0) := \{ \langle dU, \mathsf{sc}, uhP \rangle,$ $\langle dU, \mathsf{sc}, yP \rangle \}$ and also $\mathtt{ExceptionalP}(G^{str} \cup \mathtt{D}_0) := \emptyset$. Therefore,

$$\mathtt{D}_1 := \{ \langle dU, \mathsf{sc}, uhP \rangle, \langle dU, \mathsf{sc}, yP \rangle \} \ .$$

Next, in $G^{str} \cup \mathtt{D}_1$, dU is not exceptional anymore, and we obtain

$$\mathtt{D}_2 := \emptyset \ .$$

That is,

$$\mathtt{D}_\infty := \emptyset$$

and, thus,

$$\mathtt{r}(G_{(yDU, uhP)}) = \{ \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_\infty \} \ .$$

Now we can check whether $G_{(yDU, uhP)} \vDash_{\min} \langle yDU, \mathsf{sc}, uhP \rangle$. To do so, we call

$$\mathtt{DefMinEntailmentC}(G_{(yDU, uhP)}, \mathtt{r}(G_{(yDU, uhP)}), \langle yDU, uhP \rangle) \ .$$

For $i = 0$, we have $G' = (G_{(yDU, uhP)})^s$, and $G' \vdash_{\rho df_\perp} (yDU, \perp_c, yDU)$, since $(G_{(yDU, uhP)})^s \vdash_{\rho df_\perp} (dU, \perp_c, dU)$, that, together with (yDU, sc, dU), implies $(yDU, \perp_c, yDU)$ by the rule $(EmptySP')$.

For $i = 1$, we have $G' = G^{str} \cup (\mathtt{D}_1)^s$, and $G' \nvdash_{\rho df_\perp} (yDU, \perp_c, yDU)$. Hence we associate the defeasible set $\mathtt{D}_1$ to the query $\langle yDU, \mathsf{sc}, uhP \rangle$. Following the procedure $\mathtt{DefMinEntailmentC}$ we obtain:

$$\mathtt{D}^{yDU} := \mathtt{D}_1 \setminus \mathtt{D}_\infty = \{ \langle dU, \mathsf{sc}, uhP \rangle, \langle dU, \mathsf{sc}, yP \rangle \} \ .$$

Eventually, it is easy to check that:

$$G^{str} \cup (\mathtt{D}^{yDU})^s \vdash_{\rho df_\perp} (yDU, \mathsf{sc}, uhP)$$

as $(dU, \mathsf{sc}, uhP) \cup (yDU, \mathsf{sc}, dU) \subseteq G^{str} \cup (\mathtt{D}^{yDU})^s$.

Therefore, we can conclude $G_{yDU, uhP} \vDash_{\min} \langle yDU, \mathsf{sc}, uhP \rangle$. That implies that $\langle yDU, \mathsf{sc}, uhP \rangle \in G_{in}$, and consequently $G \vdash_{in} \langle yDU, \mathsf{sc}, uhP \rangle$.

We can verify that actually

$$G_{in} := G \cup \{ \langle yDU, \mathsf{sc}, uhP \rangle, \langle yDU, \mathsf{sc}, s \rangle \} \ .$$

For example, $\langle yDU, \mathsf{sc}, hP \rangle \notin G_{in}$: in fact, we have $\Delta_{(yDU, hP)} = \Delta_{(yDU, uhP)}$, that implies that we have the same ranking and the same set $\mathtt{D}^{yDU}$. Moreover, $G^{str} \cup (\mathtt{D}^{yDU})^s \nvdash_{\rho df_\perp} (yDU, \mathsf{sc}, hP)$, since $(yP, \mathsf{sc}, hP) \notin G^{str} \cup (\mathtt{D}^{yDU})^s$, which implies that $G_{(yDU, uhP)} \nvDash_{\min} \langle yDU, \mathsf{sc}, hP \rangle$.

Given G_{in}, and in particular checked that $\langle yDU, \mathsf{sc}, hP \rangle \notin G_{in}$, it is possible to verify that $G_{in} \nvDash_{\min} \langle yDU, \mathsf{sc}, hP \rangle$, that is, $G \nvdash_{in} \langle yDU, \mathsf{sc}, hP \rangle$. $\square$

4.2. Structural properties

Here we check that Inheritance-based entailment $\vdash_{in}$ satisfies some of the properties satisfied also by RC. Most of the proofs are straightforward, since $\vdash_{in}$ corresponds to an RC over the graph G_{in} (see Definition 4.2). Let Cl_{in} be the closure operation corresponding to $\vdash_{in}$:

$$\mathrm{Cl}_{in}(G) := \{ [\![s,p,o]\!] \mid G \vdash_{in} [\![s,p,o]\!] \}.$$

Some properties are the same as in Section 3.6, simply reformulated for $\vdash_{in}$.

$$(Supra_c) \quad \frac{G \vdash_{in} (p,\mathrm{sc},q)}{G \vdash_{in} \langle p,\mathrm{sc},q \rangle} \qquad (Supra_p) \quad \frac{G \vdash_{in} (p,\mathrm{sp},q)}{G \vdash_{in} \langle p,\mathrm{sp},q \rangle}$$

The proof of the next proposition proceeds as in Proposition 3.30, just considering the graph G_{in} instead of the graph G.

Proposition 4.1. $\models_{\rho df_\perp}$ satisfies $(Supra_c)$ and $(Supra_p)$.

Then we have *Reflexivity, Left Logical Equivalence,* and *Right Weakening.*

$$(Ref_c) \quad G \vdash_{in} \langle p,\mathrm{sc},p \rangle$$

$$(LLE_c) \quad \frac{G \vdash_{in} \langle p,\mathrm{sc},r \rangle, \quad G \vdash_{in} (p,\mathrm{sc},q), \quad G \vdash_{in} (q,\mathrm{sc},p)}{G \vdash_{in} \langle q,\mathrm{sc},r \rangle} \qquad (RW_c) \quad \frac{G \vdash_{in} \langle p,\mathrm{sc},q \rangle, \quad G \vdash_{in} (q,\mathrm{sc},r)}{G \vdash_{in} \langle p,\mathrm{sc},r \rangle}$$

$$(Ref_p) \quad G \vdash_{in} \langle p,\mathrm{sp},p \rangle$$

$$(LLE_p) \quad \frac{G \vdash_{in} \langle p,\mathrm{sp},r \rangle, \quad G \vdash_{in} (p,\mathrm{sp},q), \quad G \vdash_{in} (q,\mathrm{sp},p)}{G \vdash_{in} \langle q,\mathrm{sp},r \rangle} \qquad (RW_p) \quad \frac{G \vdash_{in} \langle p,\mathrm{sp},q \rangle, \quad G \vdash_{in} (q,\mathrm{sp},r)}{G \vdash_{in} \langle p,\mathrm{sp},r \rangle}$$

As for $\models_{\min}$, (Ref_c) and (Ref_p) do not hold, since we have used minimal ρdf for which reflexivity of sc and sp triples does not hold. The proof of Proposition 4.2 proceeds as in Proposition 3.31, just considering the graph G_{in} instead of the graph G.

Proposition 4.2. $\vdash_{in}$ satisfies (LLE_c), (RW_c), (LLE_p), and (RW_p).

Proposition 4.3. Cl_{in} is not monotonic.

Once more, it can be proved as for Proposition 3.32.

Proposition 4.4. Cl_{in} satisfies inclusion. That is, for any graph G

$$G \subseteq \mathrm{Cl}_{in}(G).$$

Proof. $G \subseteq G_{in}$. $\mathrm{Cl}_{\min}(G_{in})$ is determined by a model of G_{in}, implying $G \subseteq G_{in} \subseteq \mathrm{Cl}_{\min}(G_{in})$, that is, $G \subseteq \mathrm{Cl}_{in}(G)$. $\square$

However, the operator Cl_{in} does not always satisfy *Idempotence*.

Proposition 4.5. Cl_{in} does not satisfy Idempotence. *That is, there is a graph G such that*

$$\mathrm{Cl}_{in}(\mathrm{Cl}_{in}(G)) \neq \mathrm{Cl}_{in}(G).$$

An immediate consequence of Proposition 4.5 is also the failure of *Cumulativity*.

Proposition 4.6. Cl_{in} does not satisfy Cumulativity. *That is, there are at least two graphs G and G' such that*

$$G \subseteq G' \subseteq \mathrm{Cl}_{in}(G) \text{ and } \mathrm{Cl}_{in}(G') \neq \mathrm{Cl}_{in}(G).$$

Note that the same graph used in the proof of Proposition 4.5 proves also Proposition 4.6, as it is sufficient to set $G' = \mathrm{Cl}_{in}(G)$.

Remark 4.1. As stated in Proposition 4.5, Cl_{in} does not satisfy *Idempotence*. However, it is still possible to refine Cl_{in} so that it does satisfy Idempotence. In fact, it is sufficient to iterate the application of InheritanceCompletion followed by $\mathrm{Cl}_{\min}$, until we do reach a fixed point. Given that we work with finite graphs, such a fixed point must be reached after a finite number of iterations. In particular, given a graph G, the procedure Cl_{in} can be iterated at most twice for each pair of nodes p,q in G, once to obtain a new triple $\langle p,\mathrm{sc},q \rangle$, and once to obtain a new triple $\langle p,\mathrm{sp},q \rangle$. Hence the new procedure would consist in at most $2(|G|^2)$ reiterations of the procedure Cl_{in} and, thus, we still remain polynomial to decide defeasible entailment. We have preferred to introduce only one

iteration of Cl_{in}, since the graph for which Idempotence does not hold appears to be quite artificial and has a complex configuration (see the example used to prove Proposition 4.5). In particular, the inferential advantages given by the reiteration of the Cl_{in} operation do not appear to be worth of the extra computational costs.

However, at the time of writing, we are still missing a proof showing whether by iterating Cl_{in} we obtain or not *Cumulativity*.

4.3. Computational complexity

In this section we prove that deciding $\vdash_{in}$ is still computationally tractable, *i.e.*, it runs in polynomial time. We start with `InheritanceCompletion`.

In line 3 of the `InheritanceCompletion` procedure we need to compute all paths between two nodes.

Lemma 4.7. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and two nodes, p and q, in G. $\Delta_{p,q}$ can be determined in $O(|G|^2)$ time.*

Given Lemma 4.7, we can show that:

Proposition 4.8. *Consider a defeasible graph $G = G^{str} \cup G^{def}$. Assume that G^{str} is closed under* `StrictMinEntailment`. *Then the procedure* `InheritanceCompletion`(G) *runs in time $O(|G^{def}||G|^4)$.*

We conclude with:

Corollary 4.9. *Consider a defeasible graph $G = G^{str} \cup G^{def}$. Then $G \vdash_{in} [\![p, o, q]\!]$ can be decided in polynomial time.*

5. Dealing with non-ground graphs

So far, we assumed graphs to be ground. However, as pointed out in *e.g.*, [29], many ρdf graphs contain so-called *blank nodes* in triples, *i.e.*, in logical terms existentially quantified variables. We next show how our approach may be extended to non-grounded $\rho df_\perp$ graphs as well by preserving the computational complexity of the ground case.

Defeasible $\rho df_\perp$ To start with, again we recap here only the salient notions we will rely on and refer the reader *e.g.*, directly to [27,39] for more details. So, let **B** be a new alphabet, pairwise disjoint to **U** and **L**, denoting *blank nodes*, and let us denote with **UBL** the union of these alphabets. A (possibly non-ground) $\rho df_\perp$ *triple* is now of the form $\tau = (s, p, o) \in \mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}$, where again $s, o \notin \rho df$. We will use symbols x, y, z (with optional sub- or super-scripts) to denote blank nodes in triples. However, *we still assume that defeasible triples remain ground and, thus, e.g., triples of the form $\langle c, \mathrm{sc}, d \rangle$ (indicating that "typically, an instance of c is also an instance of b") remain ground.*[24]

We define a *map* as a function $\mu : \mathbf{UBL} \to \mathbf{UBL}$ preserving URIs and literals, *i.e.*, $\mu(t) = t$, for all $t \in \mathbf{UL}$. Given a graph G, we define $\mu(G) = \{(\mu(s), \mu(p), \mu(o)) \mid (s, p, o) \in G\}$. We speak of a map μ from G_1 to G_2, and write $\mu : G_1 \to G_2$, if μ is such that $\mu(G_1) \subseteq G_2$.

From a semantical point of view, we extend the notion of interpretation to capture the idea of existentiality of blank nodes by imposing the following additional condition to an interpretation $\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$:

8. on **B**, $\cdot^{\mathcal{I}}$ is a function $\cdot^{\mathcal{I}} : \mathbf{B} \to \Delta_R$.

The notion of satisfiability (resp. entailment) is as Definition 2.1 (resp. Definition 2.2).

From a deductive system point of view, as reported in [39], we have to consider some additional rules: namely,

8. Implicit Typing:

$$(a) \quad \frac{(A, \mathrm{dom}, B), (D, \mathrm{sp}, A), (X, D, Y)}{(X, \mathrm{type}, B)} \qquad (b) \quad \frac{(A, \mathrm{range}, B), (D, \mathrm{sp}, A), (X, D, Y)}{(Y, \mathrm{type}, B)}$$

9. Simple':

$$\frac{G}{G'} \text{ for a map } \mu : G' \to G$$

with the convention that in all rules (2) - (8) meta-variables may represent now elements in **UBL**. Please note that, rule (9) captures the semantics of blank nodes.

Finally, the notion of *derivation* (Definition 2.3) is extended in the obvious way:

Definition 5.1 *(Derivation $\vdash_{\rho df_\perp}$).* Let G and H be $\rho df_\perp$-graphs. $G \vdash_{\rho df_\perp} H$ iff there exists a sequence of graphs $P_1, P_2, \ldots, P_k$ with $P_1 = G$ and $P_k = H$ and for each j $(2 \leqslant j \leqslant k)$ one of the following cases holds:

---

[24] In fact, so far we do not envisage meaningful cases in which such triples may be not ground. Nevertheless, we will leave the case in which this type of triples may contain blank nodes as well for future work.

- there is a map $\mu : P_j \to P_{j-1}$ (rule (9));
- $P_j \subseteq P_{j-1}$ (rule (1));
- there is an instantiation R/R' of one of the rules (2)-(8), such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.

Such a sequence of graphs is called a proof of $G \vdash_{\rho df_\perp} H$. Whenever $G \vdash_{\rho df_\perp} H$, we say that the graph H is derived from the graph G. Each pair (P_{j-1}, P_j), $1 \leqslant j \leqslant k$ is called a step of the proof which is labelled by the respective instantiation R/R' of the rule applied at the step.

With $\mathsf{Cl}_B(G)$ we denote the closure of a $\rho df_\perp$ graph G under the application of rules $(2) - (8)$ (see also [39,27]).

Example 5.1. Consider the $\rho df_\perp$ graph G containing the triples

$$G = \{(a, \mathsf{sc}, x), (x, \mathsf{sc}, b), (y, \mathsf{sc}, c),$$

$$(c, \perp_c, d), (p, \mathsf{dom}, y), (q, \mathsf{dom}, d)\} .$$

Then, it is not difficult to verify that $\mathsf{Cl}_B(G)$ contains the following triples:

$$\mathsf{Cl}_B(G) \supseteq \{(a, \mathsf{sc}, b), (y, \perp_c, d), (p, \perp_p, q)\} . \quad \square$$

Now, likewise for Theorem 2.1, by using also [39, Theorem 8] and [39, Theorem 10], we may prove similarly that

Theorem 5.1. *Let G and H be possibly non-ground $\rho df_\perp$-graphs.*

1. $G \vdash_{\rho df_\perp} H$ *iff* $G \vDash_{\rho df_\perp} H$.
2. *If* $G \vdash_{\rho df_\perp} H$ *then there is a proof of H from G such that rule (9) is used at most once and at the last step of the proof.*

Example 5.2 *(Example 5.1 cont.).* By referring to Example 5.1, it is easily verified that indeed

$$G \vDash_{\rho df_\perp} \{(a, \mathsf{sc}, b), (y, \perp_c, d), (p, \perp_p, q)\} .$$

From a computational complexity point of view, from [47] and Theorem 5.1, point 2, it follows immediately that, as for ρdf, also for $\rho df_\perp$ we have that

Theorem 5.2. *Let G and H be possibly non-ground $\rho df_\perp$-graphs. Then*

1. *Deciding $G \vDash_{\rho df_\perp} H$ is an NP-complete problem.*
2. *If H is ground, then $G \vDash_{\rho df_\perp} H$ iff $H \subseteq \mathsf{Cl}_B(G)$.*
3. *The size of $|\mathsf{Cl}_B(G)|$ is $O(|G|^2)$.*

In particular, note that even if G may contain blank nodes, if H is ground, then deciding $G \vDash_{\rho df_\perp} H$ can be done in time $O(|G|^2)$.

Please note also the following: by using Theorem 5.1, point 2, and Theorem 5.2, we may also decide the case $G \vDash_{\rho df_\perp} (s, p, o)$, where s, o may be blank nodes, in time $O(|G|^2)$ by computing $\mathsf{Cl}_B(G)$ incrementally and then

Case $(x, p, o), x \in \mathbf{B}, p, o \in \mathbf{UL}$: check whether there is $(s, p, o) \in \mathsf{Cl}_B(G)$ with $s \in \mathbf{UBL}$;

Case $(s, p, y), y \in \mathbf{B}, s, p \in \mathbf{UL}$: check whether there is $(s, p, o) \in \mathsf{Cl}_B(G)$ with $o \in \mathbf{UBL}$;

Case $(x, p, y), x, y \in \mathbf{B}, p \in \mathbf{UL}$: check whether there is $(s, p, o) \in \mathsf{Cl}_B(G)$ with $s, o \in \mathbf{UBL}$.

Therefore,

Corollary 5.3. *Let G and H be possibly non-ground $\rho df_\perp$-graphs. If H is ground or $|H| = 1$ then $G \vDash_{\rho df_\perp} H$ can be decided in time $O(|G|^2)$.*

Next, we turn our attention to defeasible graphs G in which the strict $\rho df_\perp$ part may contain non-ground triples. We ask about the computational complexity of deciding defeasible entailment. Now, note the following:

1. Consider the ExceptionalC and ExceptionalP procedures. In both cases, in Step 3., the query triple is ground, and thus, by Corollary 5.3, both procedures run in polynomial time, and Proposition 3.40 applies.
2. From the previous point, also the ranking procedure Ranking runs in polynomial time and Proposition 3.41 applies.
3. From what was said above and Corollary 5.3, also the procedure StrictMinEntailment runs in polynomial time and Proposition 3.42 applies.

4. Finally, consider procedures DefMinEntailmentC and DefMinEntailmentP. Again, as the involved query triples are ground, both procedures run in polynomial time, and Propositions 3.43 and 3.44 apply.

Therefore, the analogue of Corollary 3.45 holds. That is,

Theorem 5.4. *Consider a defeasible graph $G = G^{str} \cup G^{def}$, where the $\rho df_\perp$ part of G^{str} may contain blank nodes. Consider a triple $[\![p, *, q]\!]$. Then $G \vDash_{\min} [\![p, *, q]\!]$ can be decided in polynomial time.*

Inheritance networks based closure   We eventually show how to extend our decision procedure for inheritance net closure described in Section 4 also to the non-ground case.

At first, as for Section 4, we will assume a possibly non-ground graph $G = G^{str} \cup G^{def}$ s.t. G^{str} is closed under StrictMinEntailment. How to do that has been described above. As next, we extend the notion of *path* given in Definition 4.1 naturally to the case in which also non-ground triples may occur in a path. So, for instance, by referring to the graph G in Example 5.1,

$$\pi = \{(a, \mathsf{sc}, x), (x, \mathsf{sc}, b)\}$$

is a path in G from a to b.

Finally, given a possibly non-ground graph G and a (ground) defeasible triple $\langle p, *, q \rangle$ with $* \in \{\mathsf{sc}, \mathsf{sp}\}$, the procedure for deciding whether $\langle p, *, q \rangle$ is in the closure of a graph G using the Defeasible Inheritance-based approach is exactly as the one described in Section 4. That is, we apply to G the procedure InheritanceCompletion(G), obtaining G_{in} and then define (see Definition 4.2), for any triple $[\![p, r, q]\!]$, that

$$G \vdash_{in} [\![p, r, q]\!] \text{ iff } G_{in} \vDash_{\min} [\![p, r, q]\!] \ .$$

From a computational complexity point of view, it is then easily verified that Lemma 4.7 holds for the non-ground case as well. Then, using Corollary 5.3, we obtain the analogue of Corollary 4.9:

Corollary 5.5. *Consider a possibly non-ground defeasible graph $G = G^{str} \cup G^{def}$. Then $G \vdash_{in} [\![p, r, q]\!]$ can be decided in polynomial time.*

In summary, even if blank nodes are allowed in $\rho df_\perp$, the computational complexity does not change w.r.t. the ground case, which concludes this section.

6. Related work

There have been some works in the past about extending RDFS with non-monotonic capabilities, which we briefly summarise below.

The series of works [1,2] essentially deals with *Extended* RDFS (ERDF), where an ERDF ontology consists of two parts: an ERDF graph and an ERDF program containing derivation rules. ERDF augments the expressivity of RDFS in the following way: an ERDF graph allows *negated* triples of the form $\neg(s, p, o)$ indicating informally that $\neg p(s, o)$[25] does hold, while in the body of derivation rules all the classical connectives $\neg, \supset, \wedge, \vee, \forall, \exists$, plus the weak negation (negation-as-failure) $\sim$ are allowed. Thanks to the latter, we can model non-monotonic reasoning via negation-as-failure semantics. For instance, one may express rules such as $\neg\mathsf{type}(x, EU Member) \leftarrow \sim \mathsf{type}(x, EU Member)$ (a non-EU Member state is one that can not be proven to be a EU member state), or $\neg\mathsf{type}(x, EU Member) \leftarrow \mathsf{type}(x, American Country)$.

ERDF allows to combine open-world and closed-world reasoning. In order to define entailment the authors have proposed a *stable model semantics* [1] and *#n-stable model semantics* [2]. In general, under stable model semantics, query answering is undecidable. Decidability can be obtained either by constraining the language or using the *#n-stable model semantics* [2]. Moreover, from a computational complexity point view, decision problems in ERDF are non-polynomial (see Table 1 in [2] for details). For instance, deciding model existence, and, thus, model existence is not guaranteed, ranges from NP to PSPACE, while query answering goes from co-NP to PSPACE, depending on the setting. In our case, model existence is guaranteed, the computational complexity is lower and no rule layer is introduced, remaining, thus, a triple language. Furthermore, while the authors did enrich RDFS language with two negation operators, we have introduced the disjointness properties $\perp_c$ and $\perp_p$ only, which are weaker than negation from a logical point of view.

On a similar line, *i.e.*, using rule languages on top of RDFS, and borrowing non-monotonic semantics developed within rules languages, are also all other approaches we are aware of, such as [3,6,32] and practical large scale solutions such as [31,42,45]. Let us also note that there are more general solutions in providing a rule layer on top of ontology languages such as RDFS and OWL, which can be found in *e.g.*, [19,20].

Strongly related to the present proposal is the introduction of non-monotonic reasoning in the framework of DL (a First-Order Logic that restricts to unary and binary predicates and specific syntactic constructs), that is aimed at modelling defeasible reasoning

---

[25] More precisely, for a predicate p there is a positive extension of p and a negative extension of p, which need not necessarily be disjoint. So, for instance, (s, p, o) enforces (s, o) to be in the positive extension of p, while $\neg(s, p, o)$ enforces (s, o) to be in the negative extension of p.

in the other main ontology formalism, the OWL family of languages. In particular, also in this area some of the most popular proposals are based on semantic solutions associated to rational closure, as mentioned in the introduction.

In [12] a form of rational closure for DLs is defined, that has later received a semantic characterisation in terms of ranked DL interpretations [11,16]. [12] introduces in the language a defeasible subsumption relation $\sqsubseteq$ between concepts, where $C \sqsubseteq D$ is read as 'Typically, an instance of the concept C is also an instance of the concept D'.

In [26] a similar approach is also considered, introducing in the language inclusion axioms of the form $\mathbf{T}C \sqsubseteq D$, whose meaning is the same as $C \sqsubseteq D$, and defining on the semantical side modular orderings on the domain, and it has been proved [16, Proposition 30] that the main semantic construction they propose is in fact equivalent to the one later proposed in [11,16]. Recently, the work [7] has also investigated how to extend the rational closure construction to DLs that do not satisfy the *disjoint union model property*.

As mentioned above, the entailment relation defined by rational closure suffers from the "drowning problem" (see Remark 3.3), that here we have addressed in Section 4 defining the inheritance-based approach. Also in DLs the same problem has been addressed in various ways, including the original formulation of the inheritance-based closure [14], and other extensions of rational closure [13,15,23].

Finally, another limit of rational closure in the framework of DLs is that it does not allow to make defeasible inferences across role connections: for example, if our ontology contains only $C \sqsubseteq D$ and $\top \sqsubseteq \exists R.C$, we cannot conclude $\top \sqsubseteq \exists R.D$. Such a limit has been addressed in *e.g.*, [41], the aim of which is to extend rational closure with defeasible inheritance across role expressions in the description logic $\mathcal{EL}_\perp$. An entailment relation that addresses the same issue, but without using a semantics based on rational closure, is the one proposed in [8]. In the RDFS framework this type of problem, *i.e.*, defeasible inheritance across existential role expressions is not present, due to the expressivity constraints of ρdf.

In summary, adopting a rule layer or, in general, a more expressive logic, however, may also come with an increase in computational cost.

7. Brief conclusions

Contribution. We have shown how one may integrate RC within RDFS and, thus, obtain a non-monotonic variant of the latter. To do so, we started from ρdf, which is the logic behind RDFS, and then extended it to $\rho df_\perp$, allowing to state that two entities are incompatible. Eventually, we have worked out defeasible $\rho df_\perp$ via a typical RC construction. Furthermore, we have addressed the "drowning problem", via a 'local', path-based application of RC.

The main and unique features of our approach are summarised as follows:

- defeasible $\rho df_\perp$ remains syntactically a triple language and is a simple vocabulary extension of ρdf by introducing some new predicate symbols, namely $\perp_c$ and $\perp_p$, with specific semantics allowing to state that two terms are incompatible;
- any RDFS reasoner/store may handle the new types of triples as ordinary ones if it does not want to take account of the extra semantics of the new predicate symbols;
- the defeasible entailment decision procedure is built on top of the $\rho df_\perp$ entailment decision procedure, which in turn is an extension of the one for ρdf via some additional inference rules, favouring a potential implementation;
- each defeasible graph has an unique minimal model;
- the "drowning problem" is addressed via a simple, 'local', path-based application of RC;
- defeasible entailment can be decided in polynomial time.

We have also analysed our proposal from a 'structural properties' point of view with respect to those properties that can be expressed in our language.[26]

Future work. Concerning future work, there are still some extensions that are worth to be addressed within our framework. In particular, (*i*) we would like to investigate how to extend defeasible triples also to other predicates of the ρdf vocabulary beyond sc and sp and possibly involving blank nodes in case it makes sense to do so; (*ii*) another point deals with the computational complexity of our framework. In particular, we would like to see whether an approach similar as described in [39] can be applied to our context as well, as explained in Remark 3.5, and we want to address the problem of conjunctive query answering; and (*iii*) last, but not least, we want to implement our proposal. In particular, to scale up in practice, we are interested in working out parallelisation of computation, a non-trivial problem for non-monotonic reasoning, that has already been investigated for some of the approaches based on the use of a top-layer of rules [46].

Declaration of competing interest

---

[26] We recall that some can not be expressed in $\rho df_\perp$ as Boolean connectives are missing in RDFS.

Acknowledgement

Appendix A. Proofs for Section 2

Lemma 2.2. *Let G and H be $\rho df_\perp$-graphs, let G be satisfiable, and let one of the following statements hold:*

- *$H \subseteq G$;*
- *there is an instantiation R/R' of one of the rules (2)-(7), such that $R \subseteq G$ and $H = G \cup R'$.*

Then, $G \vDash_{\rho df_\perp} H$.

Proof. G is satisfiable, hence let $\mathcal{I} = \langle \Delta_\mathsf{R}, \Delta_\mathsf{P}, \Delta_\mathsf{C}, \Delta_L, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$ be a model of G ($\mathcal{I} \Vdash_{\rho df_\perp} G$). That is, $\mathcal{I}$ satisfies all the conditions in Definition 2.1. We have to prove that $G \vDash_{\rho df_\perp} H$, that is, $\mathcal{I} \Vdash_{\rho df_\perp} H$.

We consider only the rules (5)-(7). The theorem has already been proved for groups of rules (1)-(4) in [39, Lemma 31].[27]

Rule (5a). Let $(c, \perp_\mathsf{c}, d) \in R$ for some $R \subseteq G$, $R' = R \cup \{(d, \perp_\mathsf{c}, c)\}$, obtained via the application of rule (5a), and $H = G \cup R'$. We have that for every model $\mathcal{I}$ of G, $\mathcal{I} \Vdash_{\rho df_\perp} R$, since $R \subseteq G$. Therefore, $\mathcal{I}$ satisfies $(c, \perp_\mathsf{c}, d)$, and, since it is a model of G and is symmetric on $\perp_\mathsf{c}$ (see Definition 2.1), we have that $(d^{\mathcal{I}}, c^{\mathcal{I}}) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^{\mathcal{I}}]\!]$. That is, $\mathcal{I}$ satisfies $(d, \perp_\mathsf{c}, c)$. Hence, from $\mathcal{I} \Vdash_{\rho df_\perp} R'$ and $\mathcal{I} \Vdash_{\rho df_\perp} G$, $\mathcal{I} \Vdash_{\rho df_\perp} H$ follows.

Rule (5b). Let $(c, \perp_\mathsf{c}, d)$ and (e, sc, c) be in R, for some $R \subseteq G$. Consider $R' = R \cup \{(e, \perp_\mathsf{c}, d)\}$, obtained via the application of rule (5b), and $H = G \cup R'$. We have that for every model $\mathcal{I}$ of G, $\mathcal{I} \Vdash_{\rho df_\perp} R$, since $R \subseteq G$. $\mathcal{I}$ satisfies $(c, \perp_\mathsf{c}, d)$ and (e, sc, c), and, since it is a model of G, by sc-transitivity of $\mathcal{I}$, $(e^{\mathcal{I}}, d^{\mathcal{I}}) \in \mathfrak{P}[\![\perp_\mathsf{p}{}^{\mathcal{I}}]\!]$ follows. That is, $\mathcal{I}$ satisfies $(e, \perp_\mathsf{c}, d)$. Hence, since $\mathcal{I} \Vdash_{\rho df_\perp} R'$ and $\mathcal{I} \Vdash_{\rho df_\perp} G$, we have that $\mathcal{I} \Vdash_{\rho df_\perp} H$.

Rule (5c). Let $(c, \perp_\mathsf{c}, c) \in R$ for some $R \subseteq G$, $R' = R \cup \{(c, \perp_\mathsf{c}, d)\}$, obtained via the application of rule (5c), and $H = G \cup R'$. We have that for every model $\mathcal{I}$ of G, $\mathcal{I} \Vdash_{\rho df_\perp} R$, since $R \subseteq G$. Therefore, $\mathcal{I}$ satisfies $(c, \perp_\mathsf{c}, c)$, and, since it is a model of G and is c-exhaustive on $\perp_\mathsf{c}$ (see Definition 2.1), we have that $(c^{\mathcal{I}}, d^{\mathcal{I}}) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^{\mathcal{I}}]\!]$. That is, $\mathcal{I}$ satisfies $(c, \perp_\mathsf{c}, d)$. Hence, from $\mathcal{I} \Vdash_{\rho df_\perp} R'$ and $\mathcal{I} \Vdash_{\rho df_\perp} G$, $\mathcal{I} \Vdash_{\rho df_\perp} H$ follows.

Rules (6a), (6b) **and** (6c). The argument is analogous to rules (5a), (5b) and (5c)

Rule (7a). Let (p, dom, c), (q, dom, d), and $(c, \perp_\mathsf{c}, d)$ be in R for some $R \subseteq G$, $R' = R \cup \{(p, \perp_\mathsf{p}, q)\}$, obtained via the application of rule (7a), and $H = G \cup R'$. We have that for every model $\mathcal{I}$ of G, $\mathcal{I} \Vdash_{\rho df_\perp} R$, since $R \subseteq G$. $\mathcal{I}$ satisfies (p, dom, c), (q, dom, d), and $(c, \perp_\mathsf{c}, d)$, that by condition 1 of Disjointness II implies $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{p}, q)$.

Rule (7b). As for rule (7a), just by referring to condition 2 of Disjointness II instead of condition 1 of Disjointness II. $\quad\square$

Lemma 2.3. *Given a $\rho df_\perp$-graph G, define an interpretation $\mathcal{I}_G$ as a tuple*

$$\mathcal{I}_G = \langle \Delta_\mathsf{R}, \Delta_\mathsf{P}, \Delta_\mathsf{C}, \Delta_\mathsf{L}, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}_G} \rangle$$

such that:

1. $\Delta_\mathsf{R} := \mathtt{uni}(G) \cup \rho df_\perp$;
2. $\Delta_\mathsf{P} := \{p \in \mathtt{uni}(G) \mid (s, p, o) \in \mathsf{Cl}(G)\} \cup \rho df_\perp \cup \{p \in \mathtt{uni}(G) \mid \text{either } (p, \mathsf{sp}, q), (q, \mathsf{sp}, p), (p, \mathsf{dom}, c), (p, \mathsf{range}, d), (p, \perp_\mathsf{p}, q) \text{ or } (q, \perp_\mathsf{p}, p) \in \mathsf{Cl}(G)\}$;
3. $\Delta_\mathsf{C} := \{c \in \mathtt{uni}(G) \mid (x, \mathsf{type}, c) \in \mathsf{Cl}(G)\} \cup \{c \in \mathtt{uni}(G) \mid \text{either } (c, \mathsf{sc}, d), (d, \mathsf{sc}, c), (p, \mathsf{dom}, c), (p, \mathsf{range}, c), (c, \perp_\mathsf{c}, d) \text{ or } (d, \perp_\mathsf{c}, c) \in \mathsf{Cl}(G)\}$;
4. $\Delta_\mathsf{L} := \mathtt{uni}(G) \cap \mathbf{L}$;
5. *$\mathfrak{P}[\![\cdot]\!]$ is an extension function $\mathfrak{P}[\![\cdot]\!] : \Delta_\mathsf{P} \to 2^{\Delta_\mathsf{R} \times \Delta_\mathsf{R}}$ s.t. $\mathfrak{P}[\![p]\!] := \{(s, o) \mid (s, p, o) \in \mathsf{Cl}(G)\}$;*
6. *$\mathfrak{C}[\![\cdot]\!]$ is an extension function $\mathfrak{C}[\![\cdot]\!] : \Delta_\mathsf{C} \to 2^{\Delta_\mathsf{R}}$ s.t. $\mathfrak{C}[\![c]\!] := \{x \in \mathtt{uni}(G) \mid (x, \mathsf{type}, c) \in \mathsf{Cl}(G)\}$;*
7. *$\cdot^{\mathcal{I}_G}$ is an identity function over Δ_R.*

Then, for every $\rho df_\perp$-graph G, $\mathcal{I}_G \Vdash_{\rho df_\perp} G$.

Proof. We need to prove that $\mathcal{I}_G$ satisfies the constraints in Definition 2.1. At first, note that for the conditions **Simple** to **Typing II**, the proof corresponds to the proof of Lemma 32 in [39].[28] So, let us verify the remaining conditions.

---

[27] In [39, Lemma 31] the authors consider a stronger system in which the predicates sc and sp are reflexive. We drop such properties, hence dropping the corresponding groups (6) and (7) of derivation rules in [39, Table 1]. Hence we need to follow the proof of Lemma 31 in [39] only until the point (4) included.

[28] With the minor difference that in [39, Lemma 32] the authors impose also reflexivity to the interpretations of the predicates sc and sp and consider the associated derivation rules, while here we do not.

Disjointness I:

1. If $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$, then $c,d \in \Delta_C$. This holds by construction of Δ_C.
2. $\mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$ is symmetric, sc-transitive and c-exhaustive over Δ_C.

   Symmetry: Let $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!] = \mathfrak{P}[\![\bot_c]\!]$. By construction of $\mathcal{I}_G$ we have that $(c, \bot_c, d) \in \mathsf{Cl}(G)$, and we also have that $c,d \in \Delta_C$. Due to the closure under rule (5a), $(d, \bot_c, c) \in \mathsf{Cl}(G)$, and by construction of $\mathfrak{P}[\![\cdot]\!]$, $(d,c) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$.

   sc-*transitivity*: Let $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!] = \mathfrak{P}[\![\bot_c]\!]$ and $(e,c) \in \mathfrak{P}[\![sc^{\mathcal{I}_G}]\!] = \mathfrak{P}[\![sc]\!]$. By construction of $\mathcal{I}_G$ we have that $(c, \bot_c, d), (e, sc, c) \in \mathsf{Cl}(G)$, and we also have that $c,d,e \in \Delta_C$. Due to the closure under rule (5b), $(e, \bot_c, d) \in \mathsf{Cl}(G)$, and by construction of $\mathfrak{P}[\![\cdot]\!]$, $(e,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$.

   c-*exhaustive*: Let $(c,c) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!] = \mathfrak{P}[\![\bot_c]\!]$. By construction of $\mathcal{I}_G$ we have that $(c, \bot_c, c) \in \mathsf{Cl}(G)$ and $c \in \Delta_C$. Due to the closure under rule (5c), $(c, \bot_c, d) \in \mathsf{Cl}(G)$, and by construction of $\mathfrak{P}[\![\cdot]\!]$, $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$ with $d \in \Delta_C$.
3. If $(p,q) \in \mathfrak{P}[\![\bot_p{}^{\mathcal{I}_G}]\!]$, then $p,q \in \Delta_P$. This holds by construction of Δ_P.
4. $\mathfrak{P}[\![\bot_p{}^{\mathcal{I}_G}]\!]$ is symmetric, sp-transitive and p-exhaustive over Δ_P. The proof is a rephrasing of the proof of point 2.

Disjointness II:

1. If $(p,c) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}_G}]\!]$, $(q,d) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}_G}]\!]$, and $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$, then $(p,q) \in \mathfrak{P}[\![\bot_p{}^{\mathcal{I}_G}]\!]$.

   Indeed, let $(p,c) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}_G}]\!]$, $(q,d) \in \mathfrak{P}[\![\mathsf{dom}^{\mathcal{I}_G}]\!]$, and $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$. By construction of $\mathfrak{P}[\![\cdot]\!]$, (p, dom, c), (q, dom, d), and $(c, \bot_c, d)$ are in $\mathsf{Cl}(G)$. Moreover, $\mathsf{Cl}(G)$ is closed under rule (7a) and, thus, $(p, \bot_c, q) \in \mathsf{Cl}(G)$, and, by construction of $\mathfrak{P}[\![\cdot]\!]$, $(p,q) \in \mathfrak{P}[\![\bot_p{}^{\mathcal{I}_G}]\!]$.
2. If $(p,c) \in \mathfrak{P}[\![\mathsf{range}^{\mathcal{I}_G}]\!]$, $(q,d) \in \mathfrak{P}[\![\mathsf{range}^{\mathcal{I}_G}]\!]$, and $(c,d) \in \mathfrak{P}[\![\bot_c{}^{\mathcal{I}_G}]\!]$, then $(p,q) \in \mathfrak{P}[\![\bot_p{}^{\mathcal{I}_G}]\!]$.

   The proof is analogous to the previous point: we just need to refer to range and rule (7b) instead of dom and rule (7a), which concludes. $\square$

Lemma 2.4. *Let G and H be $\rho df_\bot$-graphs. If $G \vDash_{\rho df_\bot} H$ then $H \subseteq \mathsf{Cl}(G)$.*

Proof. The proof mirrors the proof of Lemma 33 in [39]. In particular, consider the interpretation

$$\mathcal{I}_G = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^{\mathcal{I}_G} \rangle$$

as defined in Lemma 2.3. Therefore, as both $\mathcal{I}_G \Vdash_{\rho df_\bot} G$ and $G \vDash_{\rho df_\bot} H$ hold, we have $\mathcal{I}_G \Vdash_{\rho df_\bot} H$ by Definition 2.2. Therefore, for each $(s,p,o) \in H$, $p^{\mathcal{I}_G} \in \Delta_P$ and $(s^{\mathcal{I}_G}, o^{\mathcal{I}_G}) \in \mathfrak{P}[\![p^{\mathcal{I}_G}]\!]$. Moreover, by construction $p^{\mathcal{I}_G} = p$, and $\mathfrak{P}[\![p^{\mathcal{I}_G}]\!] = \mathfrak{P}[\![p]\!] = \{(s,o) \mid (s,p,o) \in \mathsf{Cl}(G)\}$. Finally, since $(s^{\mathcal{I}_G}, o^{\mathcal{I}_G}) \in \mathfrak{P}[\![p^{\mathcal{I}_G}]\!]$, we have that $(s^{\mathcal{I}_G}, p^{\mathcal{I}_G}, o^{\mathcal{I}_G})) \in \mathsf{Cl}(G)$, *i.e.*, $(s,p,o) \in \mathsf{Cl}(G)$, for each $(s,p,o) \in H$. Therefore, $H \subseteq \mathsf{Cl}(G)$, which concludes. $\square$

Appendix B. Proofs for Section 3

Proposition 3.1. *For every defeasible graph G, $|\min_\preceq(\mathfrak{R}_G)| = 1$.*

Proof. Assume to the contrary that $|\min_\preceq(\mathfrak{R}_G)| > 1$, *i.e.*, there are $\mathcal{R} = (\mathcal{M}, r)$, $\mathcal{R}' = (\mathcal{M}, r')$, with $\mathcal{R}, \mathcal{R}' \in \min_\preceq(\mathfrak{R}_G)$ and $\mathcal{R} \neq \mathcal{R}'$.

It cannot be the case that $\mathcal{R} \preceq \mathcal{R}'$ and $\mathcal{R}' \preceq \mathcal{R}$, since that would imply that $\mathcal{R} = \mathcal{R}'$. Hence it must be the case that $\mathcal{R}$ and $\mathcal{R}'$ are incomparable w.r.t. $\preceq$; that is, there are at least two interpretations $\mathcal{I}, \mathcal{J}$ in $\mathcal{M}$ s.t. $r(\mathcal{I}) < r'(\mathcal{I})$ and $r'(\mathcal{J}) < r(\mathcal{J})$.

Now, consider $\mathcal{R}^\star = (\mathcal{M}, r^\star)$, with $r^\star(\mathcal{I}) = \min\{r(\mathcal{I}), r'(\mathcal{I})\}$ for every $\mathcal{I} \in \mathcal{M}$. Clearly, $\mathcal{R}^\star \prec \mathcal{R}$ and $\mathcal{R}^\star \prec \mathcal{R}'$. As next, let us prove that $\mathcal{R}^\star \Vdash_{\rho df_\bot} G$. At first, note that $\mathcal{R}^\star \Vdash_{\rho df_\bot} G^{str}$ holds as the satisfaction of G^{str} depends only on $\mathcal{M}$ and not on the ranks. At second, assume to the contrary that $\mathcal{R}^\star \nVdash_{\rho df_\bot} G^{def}$, that is, there is a defeasible triple $\langle s,p,o \rangle \in G$ such that $\mathcal{R}^\star \nVdash_{\rho df_\bot} \langle s,p,o \rangle$ and let's assume that $\langle s,p,o \rangle$ is of the form $\langle c, sc, d \rangle$ (the proof for the case $\langle p, sp, q \rangle$ is similar). That means that there is $\bar{\mathcal{I}} \in c\_\min(c, \mathcal{R}^\star)$ s.t. $\bar{\mathcal{I}} \nVdash_{\rho df_\bot} (c, sc, d)$. But, by the definition of $r^\star$, either $\bar{\mathcal{I}} \in c\_\min(c, \mathcal{R})$ or $\bar{\mathcal{I}} \in c\_\min(c, \mathcal{R}')$ and, thus, either $\mathcal{R} \nVdash_{\rho df_\bot} \langle c, sc, d \rangle$ or $\mathcal{R}' \nVdash_{\rho df_\bot} \langle c, sc, d \rangle$ holds, against the hypothesis that both $\mathcal{R}$ and $\mathcal{R}'$ are ranked models of G. As a consequence, $\mathcal{R}^\star \Vdash_{\rho df_\bot} G^{def}$ has to hold and, thus, $\mathcal{R}^\star \Vdash_{\rho df_\bot} G$. That is, $\mathcal{R}^\star \in \mathfrak{R}_G$. But then, it can not be the case that $\mathcal{R}, \mathcal{R}' \in \min_\preceq(\mathfrak{R}_G)$ as there is $\mathcal{R}^\star \in \mathfrak{R}_G$ with $\mathcal{R}^\star \prec \mathcal{R}$ and $\mathcal{R}^\star \prec \mathcal{R}'$, which is against our hypothesis. Therefore, there can not be two distinct ranked models $\mathcal{R}, \mathcal{R}'$ in $\min_\preceq(\mathfrak{R}_G)$ and, as $\mathfrak{R}_G$ is not empty, $|\min_\preceq(\mathfrak{R}_G)| = 1$ has to hold. $\square$

Proposition 3.2. *A term t is c-exceptional (resp. p-exceptional) w.r.t. a defeasible graph G iff it is c-exceptional (resp. p-exceptional) w.r.t. $\mathcal{R}_{\min G}$.*

Proof. Let t be c-exceptional.

$\Rightarrow$) Immediate from the definition of exceptionality.
$\Leftarrow$) We proceed by contradiction: let t be c-exceptional w.r.t. $\mathcal{R}_{\min G}$, and assume that there is a model $\mathcal{R} = (\mathcal{M}, r_\mathcal{R}) \in \mathfrak{R}_G$ s.t. t is not c-exceptional w.r.t. $\mathcal{R}$. That is, there is an $\mathcal{I} \in \mathcal{M}$ s.t. $r_\mathcal{R}(\mathcal{I}) = 0$ and $\mathcal{I} \nVdash_{\rho df_\bot} (t, \bot_c, t)$. In such a case, by Definition 3.5 and Proposition 3.1, we would have that $\mathcal{I}$ has rank 0 also in $\mathcal{R}_{\min G}$. Consequently t cannot be c-exceptional w.r.t. $\mathcal{R}_{\min G}$, which contradicts our assumption.

The proof is analogous if t is p-exceptional. $\square$

Lemma 3.4. *Let T be a $\rho df_\perp$ proof tree from H to (p, sc, q). Then T contains only triples of the form (A, sc, B).*

Proof. The proof is on induction on the depth $d(T)$ of T, where by assumption, T has (p, sc, q) as root.

Case $d(T) = 0$. Hence, the tree's only node is (p, sc, q), which concludes.

Case $d(T) = 1$. In this case, there is only one possible tree, obtained by instantiating rule (3a):

$$\frac{(p, \mathsf{sc}, r), (r, \mathsf{sc}, q)}{(p, \mathsf{sc}, q)}$$

Remark. Note that we cannot instantiate rule (2b) in the form

$$\frac{(r, \mathsf{sp}, \mathsf{sc}), (p, r, q)}{(p, \mathsf{sc}, q)} \ ,$$

as $(r, \mathsf{sp}, \mathsf{sc})$ is not allowed to occur in our language (see Section 2).

Case $d(T) = n + 1$. Let us assume that the lemma holds for all proof trees of depth $m \leqslant n$, with $n \geqslant 1$. Let us show that it holds also for the case $d(T) = n + 1$ as well.

Note that the tree T of depth $n + 1$ with root (p, sc, q) can only be built by taking two trees T_1 and T_2 that have as roots triples of the form (A_i, sc, B_i) $(i = 1, 2)$ with $\max(d(T_1), d(T_2)) = n$, and applying to their roots rule (3a). Therefore, by construction of T and by induction on T_i, also the tree T of depth $n + 1$ contains only triples of the form (A, sc, B), which concludes. $\quad\square$

Lemma 3.5. *Let T be a $\rho df_\perp$ proof tree from H to $(p, \perp_\mathsf{c}, q)$. Then T contains only triples of the form (A, sc, B) or $(A, \perp_\mathsf{c}, B)$.*

Proof. The proof is on induction on the depth $d(T)$ of T, where by assumption, T has $(p, \perp_\mathsf{c}, q)$ as root.

Case $d(T) = 0$. Hence, the tree's only node is $(p, \perp_\mathsf{c}, q)$, which concludes.

Case $d(T) = 1$. In this case, there are only three possible trees, obtained by instantiating rules $(5a)$, $(5b)$ or $(5c)$: namely,

$$\frac{(q, \perp_\mathsf{c}, p)}{(p, \perp_\mathsf{c}, q)} \ , \frac{(s, \perp_\mathsf{c}, q), (p, \mathsf{sc}, s)}{(p, \perp_\mathsf{c}, q)} \ \text{ or } \ \frac{(p, \perp_\mathsf{c}, p)}{(p, \perp_\mathsf{c}, q)} \ .$$

In all three cases the lemma is satisfied, which concludes.

Case $d(T) = n + 1$. Let us assume that the lemma holds for all proof trees of depth $m \leqslant n$, with $n \geqslant 1$. Let us show that it holds also for the case $d(T) = n + 1$ as well.

Note that the tree T of depth $n + 1$ with root $(p, \perp_\mathsf{c}, q)$ can only be built in three ways:

- by applying rule (5a) to a tree T_1 of depth n having as root a triple of form $(A, \perp_\mathsf{c}, B)$;
- by applying the rule (5b) to two trees T_2 and T_3, with $\max(d(T_2), d(T_3)) = n$, having as root, respectively, a triple of form $(A, \perp_\mathsf{c}, B)$ and a triple of form (A, sc, B);
- by applying rule (5c) to a tree T_4 of depth n having as root a triple of form $(A, \perp_\mathsf{c}, A)$;

Now, by construction of T, by induction hypothesis on T_1, T_2, T_4 and by Lemma 3.4 applied to T_3, also the tree T of depth $n + 1$ contains only triples of the form (A, sc, B) and $(A, \perp_\mathsf{c}, B)$, which concludes. $\quad\square$

Lemma 3.6. *Let G be a defeasible $\rho df_\perp$-graph, $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model in $\mathfrak{R}_G$ and $\langle p, \mathsf{sc}, q \rangle \in G^{def}$. For every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$.*

Analogously, for every $\langle p, \mathsf{sp}, q \rangle \in G^{def}$ and every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{p}, p)$.

Proof. The proof is immediate from the fact that every $\mathcal{R}$ in $\mathfrak{R}_G$ is a model of G and Definition 3.4. In fact, if there is $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$ and $\mathcal{I}$ does satisfy neither (p, sc, q) nor $(p, \perp_\mathsf{c}, p)$, then $\mathcal{R} \nVdash_{\rho df_\perp} \langle p, \mathsf{sc}, q \rangle$ and, thus, $\mathcal{R}$ is not a model of G, against the hypothesis.

The proof for $\langle p, \mathsf{sp}, q \rangle \in G^{def}$ is similar. $\quad\square$

Lemma 3.7. *Let G be a defeasible $\rho df_\perp$-graph, $\mathcal{R} = (\mathcal{M}, r)$ be a ranked model in $\mathfrak{R}_G$ and let $G^s \vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ for some terms p, q. For every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$.*

Analogously, if $G^s \vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ for some terms p, q, then for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sp}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{p}, p)$.

Proof. We prove the first half, involving the subclass predicate.

So, assume $G^s \vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ and let $\mathcal{I}$ be a $\rho df_\perp$ interpretation in $\mathcal{M}$ that has rank 0. The proof is on induction on the depth $d(T)$ of a tree T, where T has (p, sc, q) as root.

Case $d(T) = 0$. Hence, the tree's only node is (p, sc, q). Therefore, either $(p, \mathsf{sc}, q) \in G^{str}$ or $\langle p, \mathsf{sc}, q \rangle \in G^{def}$. In the former case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$, as, being $\mathcal{R}$ a model of G, every $\mathcal{I} \in \mathcal{M}$ must satisfy G^{str}. In the latter case, by Lemma 3.6, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$, which concludes.

Case $d(T) = 1$. In this case, there is only one possible tree, obtained by instantiating rule (3a):

$$\frac{(p, \mathsf{sc}, r), (r, \mathsf{sc}, q)}{(p, \mathsf{sc}, q)}$$

Assume $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$. As sc is a transitive relation, we have two possibilities only:

Case $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$. Then, since $(p, \mathsf{sc}, r) \in G^s$, it must be the case that $\langle p, \mathsf{sc}, r \rangle \in G^{def}$ and, thus, by Lemma 3.6, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$ **but** $\mathcal{I} \not\Vdash_{\rho df_\perp} (r, \mathsf{sc}, q)$. Then, since $(r, \mathsf{sc}, q) \in G^s$, it must be the case that $\langle r, \mathsf{sc}, q \rangle \in G^{def}$. By Lemma 3.6, $\mathcal{I} \Vdash_{\rho df_\perp} (r, \perp_c, r)$. So we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$ and $\mathcal{I} \Vdash_{\rho df_\perp} (r, \perp_c, r)$. Given the derivation rule (EmptySC') and the fact that $\vdash_{\rho df_\perp}$ is sound, we can conclude that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.

Therefore, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, which concludes.

Case $d(T) = n + 1$. Let us assume that the lemma holds for all proof trees of depth $m \leqslant n$, with $n \geqslant 1$. Let us show that it holds also for the case $d(T) = n + 1$ as well, where T has (p, sc, q) as root. Now, since $d(T) > 1$, as for case $d(T) = 1$, the only possibility is that the tree terminates with an instantiation of rule (3a): that is,

$$\frac{(p, \mathsf{sc}, r), (r, \mathsf{sc}, q)}{(p, \mathsf{sc}, q)}$$

where (p, sc, r) and (r, sc, q) are, respectively, the roots of trees T_1 and T_2, the immediate subtrees of T, with $\max(d(T_1), d(T_2)) = n$.

By inductive hypothesis on T_i, either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, and $\mathcal{I} \Vdash_{\rho df_\perp} (r, \mathsf{sc}, q)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (r, \perp_c, r)$.

As a consequence, we have three cases:

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$. The lemma is satisfied immediately.

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$ **and** $\mathcal{I} \Vdash_{\rho df_\perp} (r, \perp_c, r)$. Then, given the derivation rule (EmptySC') and the fact that $\vdash_{\rho df_\perp}$ is sound, we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, which concludes.

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, r)$ **and** $\mathcal{I} \Vdash_{\rho df_\perp} (r, \mathsf{sc}, q)$. Then, by the transitivity of the interpretation of sc, we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, q)$, which concludes.

For the second half of the lemma, involving the predicate sp, the proof has exactly the same structure, with the only difference that it refers to the rule (2a) instead of the rule (3a). □

Lemma 3.8. *Let G be a defeasible graph, p and q terms, and let $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ be G's minimal model. If $G^s \vdash_{\rho df_\perp} (p, \perp_c, q)$ then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$ for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$.*

Proof. Let G be a defeasible graph s.t. $G^s \vdash_{\rho df_\perp} (p, \perp_c, q)$. Then by Proposition 3.3, there is a $\rho df_\perp$ proof tree T from H to $(p, \perp_c, q)$ for some $H \subseteq G^s$. We prove now the lemma by induction on the depth $d(T)$ of T.

Given $\mathcal{R}_{\min G} = (\mathcal{M}, r)$, we need to prove that $(p, \perp_c, q)$ is satisfied by every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$. Note that, as $\mathcal{R}_{\min G}$ is a model of G, for every $\mathcal{I} \in \mathcal{M}$, $\mathcal{I} \Vdash_{\rho df_\perp} G^{str}$ holds (see Definition 3.4).

Case $d(T) = 0$. The tree's only node is $(p, \perp_c, q)$, that is in G^s. Recall that G^s is the union of G^{str} and the strict translation of the defeasible triples in G^{def} (Equation (3)). Note that $(p, \perp_c, q)$ cannot be the strict form of a triple in G^{def}, as the defeasible triples must contain sc or sp as second element (Definition 3.1), hence $(p, \perp_c, q) \in G^{str}$. Being $\mathcal{R}_{\min G}$ a model of G, its strict part G^{str} must be satisfied by every $\mathcal{I} \in \mathcal{M}$ and, as a consequence, the triple $(p, \perp_c, q)$ must be satisfied by every $\mathcal{I} \in \mathcal{M}$ with rank 0, which concludes.

Case $d(T) = 1$. There are only three possible trees of depth 1 with a triple $(p, \perp_c, q)$ obtained by instantiating rules (5a), (5b) or (5c):

$$\frac{(q, \perp_c, p)}{(p, \perp_c, q)}, \quad \frac{(s, \perp_c, q), (p, \mathsf{sc}, s)}{(p, \perp_c, q)} \quad \text{or} \quad \frac{(p, \perp_c, p)}{(p, \perp_c, q)} \ .$$

In the first case, we can refer to the case $d(T) = 0$ and the fact that $\mathcal{I}$ must satisfy the symmetry of $\perp_c$. The third case is similar to the first one taking into account that $\mathcal{I}$ must be c-exhaustive on $\perp_c$. In the second case the tree consists of an instantiation of rule (5b), that has two premises, $(s, \perp_c, q)$ and (p, sc, s), that must both be in G^s. $(s, \perp_c, q)$ does not have a correspondent defeasible triple, so $(s, \perp_c, q) \in G^{str}$. Since G^{str} must be satisfied by every ρdf-interpretation in a ranked model of G, we have $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, q)$.

Concerning the premise (p, sc, s), we have two possible cases:

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathsf{sc}, s)$. In this case, from $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, q)$ and the soundness of $\vdash_{\rho df_\perp}$, we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$.

Case $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, \mathsf{sc}, s)$. In this case, $(p, \mathsf{sc}, s) \in G^s$ and $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, \mathsf{sc}, s)$ implies that $(p, \mathsf{sc}, s) \notin G^{str}$, so $\langle p, \mathsf{sc}, s \rangle \in G^{def}$ must hold. By Lemma 3.6, $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, \mathsf{sc}, s)$, implies that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$. Now, by rule (5c) and the soundness of $\vdash_{\rho df_\perp}$, we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$.

Case $d(T) = n + 1$. Let us assume that the proposition holds for all the proof trees of depth $m \leqslant n$, with $n \geqslant 1$. Let us show that it holds also for the case $d(T) = n + 1$ as well, where T has $(p, \perp_c, q)$ as root. Then, since $d(T) > 1$, as for the case $d(T) = 1$ the only three possibilities are that the tree terminates with an application rule (5a), (5b) or (5c): that is,

$$\frac{(q, \perp_c, p)}{(p, \perp_c, q)}, \quad \frac{(s, \perp_c, q), (p, \mathsf{sc}, s)}{(p, \perp_c, q)} \quad \text{or} \quad \frac{(p, \perp_c, p)}{(p, \perp_c, q)} \ .$$

The first case is straightforward, since the immediate subtree of T would be a tree of depth n that has $(q, \perp_c, p)$ as root: by inductive hypothesis $\mathcal{I} \Vdash_{\rho df_\perp} (q, \perp_c, p)$ and, since $\mathcal{I}$ must satisfy the symmetry of $\perp_c$, we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$. The third case can be proven similarly to the first case by relying on fact that $\mathcal{I}$ is c-exhaustive over Δ_C. In the second case $(s, \perp_c, q)$ and (p, sc, s) are, respectively, the roots of trees T_1 and T_2, the immediate subtrees of T, with $\max(d(T_1), d(T_2)) = n$. By inductive hypothesis on T_1, $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, q)$. Concerning T_2 and its root (p, sc, s) we have two possible cases:

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, sc, s)$. In this case, by rule (5b) and the soundness of $\vdash_{\rho df_\perp}$ we conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$.

Case $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, sc, s)$. In this case, by the second part of Lemma 3.7 we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$. Now, by rule (5c) and the soundness of $\vdash_{\rho df_\perp}$, we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, q)$.

This concludes the proof of then lemma. $\square$

Lemma 3.10. *Let G be a defeasible graph, p, q be any pair of terms, and let $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ be G's minimal model. If $G^s \vdash_{\rho df_\perp} (p, \perp_p, q)$ then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, q)$ for every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$.*

Proof. Let G be a defeasible graph s.t. $G^s \vdash_{\rho df_\perp} (p, \perp_p, q)$. Then by Proposition 3.3 there must be a $\rho df_\perp$ proof tree T deriving $(p, \perp_p, q)$ from some graph $H \subseteq G^s$. We prove now the lemma by induction on the depth $d(T)$ of T.

Given $\mathcal{R}_{\min G} = (\mathcal{M}, r)$, we need to prove that $(p, \perp_p, q)$ is satisfied by every $\mathcal{I} \in \mathcal{M}$ s.t. $r(\mathcal{I}) = 0$. Note that, as $\mathcal{R}_{\min G}$ is a model of G, for every $\mathcal{I} \in \mathcal{M}$, $\mathcal{I} \Vdash_{\rho df_\perp} G^{str}$ holds (see Definition 3.4).

Case $d(T) = 0$. The tree's only node is $(p, \perp_p, q)$, that is in G^s. Recall that G^s is the union of G^{str} and the strict translation of the defeasible triples in G^{def} (Equation (3)). Note that $(p, \perp_p, q)$ cannot be the strict form of a triple in G^{def}, since the defeasible triples must contain sc or sp as second element (Definition 3.1), hence $(p, \perp_p, q) \in G^{str}$. Being $\mathcal{R}_{\min G}$ a model of G, its strict part G^{str} must be satisfied by every $\mathcal{I} \in \mathcal{M}$ (see Definition 3.4), and as a consequence the triple $(p, \perp_p, q)$ must be satisfied by every $\mathcal{I} \in \mathcal{M}$ with rank 0, which concludes.

Case $d(T) = 1$. There are five possible trees of depth 1 with a triple $(p, \perp_p, q)$ as root, obtained by instantiating rule $(6a), (6b), (6c), (7a)$ or $(7b)$: *i.e.*,

(1) $\dfrac{(q, \perp_p, p)}{(p, \perp_p, q)}$

(2) $\dfrac{(s, \perp_p, q), (p, sp, s)}{(p, \perp_p, q)}$

(3) $\dfrac{(p, \perp_p, p)}{(p, \perp_p, q)}$

(4) $\dfrac{(p, dom, r), (q, dom, s), (r, \perp_c, s)}{(p, \perp_p, q)}$

(5) $\dfrac{(p, range, r), (q, range, s), (r, \perp_c, s)}{(p, \perp_p, q)}$.

Case (1). We can refer to the case $d(T) = 0$ and the fact that $\mathcal{I}$ must satisfy the symmetry of $\perp_p$.

Case (3). This case is similar to Case (1) by referring to the fact that $\mathcal{I}$ is c-exhaustive.

Case (2). The tree consists of an instantiation of rule (6b), that has two premises, $(s, \perp_p, q)$ and (p, sp, s), that must both be in G^s. $(s, \perp_p, q)$ does not have a correspondent defeasible triple, so $(s, \perp_p, q) \in G^s$ implies $(s, \perp_p, q) \in G^{str}$. Since G^{str} must be satisfied by every ρdf-interpretation in a ranked model of G, we have $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_p, q)$.

Concerning the premise (p, sp, s), we have two possible cases:

Case $\mathcal{I} \Vdash_{\rho df_\perp} (p, sp, s)$. In this case, from $\mathcal{I} \Vdash_{\rho df_\perp} (p, sp, s)$, $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_p, q)$, and the soundness of $\vdash_{\rho df_\perp}$, we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, q)$.

Case $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, sp, s)$. In this case, the situation in which $(p, sp, s) \in G^s$ and $\mathcal{I} \not\Vdash_{\rho df_\perp} (p, sp, s)$ is possible only if $(p, sp, s) \in G^{def}$. By Lemma 3.6, we have $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, p)$. Now, by rule (6c) and the soundness of $\vdash_{\rho df_\perp}$ we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, q)$.

Case (4). $(p, dom, r), (q, dom, s), (r, \perp_c, s)$ are in G^s, and since they cannot have a defeasible version, they must be in G^{str} too. All the triples in G^{str} must be satisfied by every $\mathcal{I} \in \mathcal{M}$. This, together with the soundness of $\vdash_{\rho df_\perp}$, guarantees that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, q)$.

Case (5). Exactly as Case (4), just consider the triples of form $(A, range, B)$ instead of the triples (A, dom, B).

Case $d(T) = n + 1$. Let us assume that the proposition holds for all the proof trees of depth $m \leqslant n$, with $n \geqslant 1$. Let us show that it holds also for the case $d(T) = n + 1$ as well, where T has $(p, \perp_p, q)$ as root. Then, since $d(T) > 1$, as for the case with $d(T) = 1$ there are five possibilities, with the tree terminating with an application of the rule $(6a), (6b), (6c), (7a)$ or $(7b)$: namely,

(1) $\dfrac{(q, \perp_p, p)}{(p, \perp_p, q)}$

(2) $\dfrac{(s, \perp_p, q), (p, sp, s)}{(p, \perp_p, q)}$

(3) $\dfrac{(p, \perp_p, p)}{(p, \perp_p, q)}$

(4) $\dfrac{(p, dom, r), (q, dom, s), (r, \perp_c, s)}{(p, \perp_p, q)}$

(5) $\dfrac{(p,\mathsf{range},r),(q,\mathsf{range},s),(r,\perp_\mathsf{c},s)}{(p,\perp_\mathsf{p},q)}$.

Case (1). Straightforward, as the immediate subtree of T is a tree of depth n that has $(q,\perp_\mathsf{p},p)$ as root: by inductive hypothesis $\mathcal{I}\Vdash_{\rho df_\perp}(q,\perp_\mathsf{p},p)$ and, since $\mathcal{I}$ must satisfy the symmetry of $\perp_\mathsf{p}$, we have $\mathcal{I}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{p},q)$.

Case (3). The proof is as for Case (1) by referring to the fact that $\mathcal{I}$ is c-exhaustive.

Case (2). $(s,\perp_\mathsf{p},q)$ and (p,sp,s) are, respectively, the roots of T_1 and T_2, the immediate subtrees of T, with $\max(d(T_1),d(T_2))=n$. By inductive hypothesis on T_1, $\mathcal{I}\Vdash_{\rho df_\perp}(s,\perp_\mathsf{p},q)$. Concerning T_2 and its root (p,sp,s), we have two possible cases:

  Case $\mathcal{I}\Vdash_{\rho df_\perp}(p,\mathsf{sp},s)$. In this case, from $\mathcal{I}\Vdash_{\rho df_\perp}(s,\perp_\mathsf{p},q)$ and by the soundness of $\vdash_{\rho df_\perp}$ we can conclude $\mathcal{I}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{p},q)$.

  Case $\mathcal{I}\nVdash_{\rho df_\perp}(p,\mathsf{sp},s)$. In this case, by Lemma 3.7 we have $\mathcal{I}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{p},p)$. Now, by rule (6c) and the soundness of $\vdash_{\rho df_\perp}$ we can conclude $\mathcal{I}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{p},q)$.

Case (4). Please note that for every pair of terms p,q, $G^s\vdash_{\rho df_\perp}(p,\mathsf{dom},q)$ holds only if $(p,\mathsf{dom},q)\in G^s$: no triple of form (A,dom,B) can be derived, since there are no rules in $\rho df_\perp$ that have triples of form (A,dom,B) as conclusions. The only possibility could be the rule (2b), by substituting E with dom, but in that case in the premises we would have a triple with dom in the third position, that is not acceptable in our language (see Section 2.1).

Since the triples (A,dom,B) do not have a defeasible version, $(p,\mathsf{dom},q)\in G^s$ implies that $(p,\mathsf{dom},q)\in G^{str}$, and consequently $\mathcal{I}\Vdash_{\rho df_\perp}(p,\mathsf{dom},q)$. It follows that in case the tree T terminates with an application of rule (7a), it will have three immediate subtrees: two trees will both have depth 0 and will consist, respectively, only of the nodes (p,dom,r) and (q,dom,s); the third subtree, called T', will have $(r,\perp_\mathsf{c},s)$ as root.

We know from Lemma 3.8 applied to T' that $\mathcal{I}\Vdash_{\rho df_\perp}(r,\perp_\mathsf{c},s)$. So we have that $\mathcal{I}\Vdash_{\rho df_\perp}(p,\mathsf{dom},r)$, $\mathcal{I}\Vdash_{\rho df_\perp}(q,\mathsf{dom},s)$, $\mathcal{I}\Vdash_{\rho df_\perp}(r,\perp_\mathsf{c},s)$, and, by the soundness of $\vdash_{\rho df_\perp}$, we can conclude $\mathcal{I}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{p},q)$.

Case (5). The proof for this case is analogous to Case (4): it is sufficient to substitute dom with range.

This concludes the proof of the lemma. □

Proposition 3.12. *A term t is c-exceptional (resp., p-exceptional) w.r.t. a defeasible graph G iff $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$).*

Proof. $\Rightarrow$.) Let us show that if a term t is c-exceptional (resp., p-exceptional) w.r.t. a defeasible graph G, then $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$).

Let t be a term in G, and let t be c-exceptional (resp., p-exceptional) w.r.t. G. Then t is c-exceptional (resp., p-exceptional) w.r.t. $\mathcal{R}_{\min G}=(\mathcal{M},r)$, that is, for every $\mathcal{I}\in\mathcal{M}$ s.t. $r(\mathcal{I})=0$, we have that $\mathcal{I}\Vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $\mathcal{I}\Vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$). Now, we need to prove that $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$).

Let $\mathcal{I}_{G^s}$ be the canonical model of G^s. By construction and Lemma 2.3, $\mathcal{I}_{G^s}\in\mathcal{M}$. It is now sufficient to prove that $r(\mathcal{I}_{G^s})=0$. Let us proceed by contradiction by assuming that $r(\mathcal{I}_{G^s})>0$, and let $\mathcal{R}'=(\mathcal{M},r')$ be the ranked interpretation obtained from $\mathcal{R}_{\min G}$, with for every $\mathcal{I}\in\mathcal{M}$,

$$r'(\mathcal{I})=\begin{cases}0 & \text{if }\mathcal{I}=\mathcal{I}_{G^s}\\ r(\mathcal{I}) & \text{otherwise}.\end{cases}$$

We can easily check that $\mathcal{R}'$ is a model of $G=G^{str}\cup G^{def}$:

- $\mathcal{R}'$ is still a model of G^{str}, since the satisfaction of $\rho df_\perp$-triples is not affected by the ranking function.
- for every $\langle p,\mathsf{sc},q\rangle\in G^{def}$, $\mathcal{I}_{G^s}\Vdash_{\rho df_\perp}(p,\mathsf{sc},q)$ since $\mathcal{I}_{G^s}$ is a model of G^s. Now we have two cases:

  $\mathcal{I}_{G^s}\Vdash_{\rho df_\perp}(p,\perp_\mathsf{c},p)$. In such a case, by Definition 3.4, $\mathcal{I}_{G^s}\notin \mathsf{c\_min}(p,\mathcal{R}')$ and $\mathcal{I}_{G^s}$ is irrelevant to decide the satisfaction of $\langle p,\mathsf{sc},q\rangle$ in $\mathcal{R}'$. Consequently $\mathsf{c\_min}(p,\mathcal{R}')=\mathsf{c\_min}(p,\mathcal{R}_{\min G})$, and $\mathcal{R}_{\min G}\Vdash_{\rho df_\perp}\langle p,\mathsf{sc},q\rangle$ implies $\mathcal{R}'\Vdash_{\rho df_\perp}\langle p,\mathsf{sc},q\rangle$.

  $\mathcal{I}_{G^s}\nVdash_{\rho df_\perp}(p,\perp_\mathsf{c},p)$. Then, since $r'(\mathcal{I}_{G^s})=0$ and $r'(\mathcal{I})=r(\mathcal{I})$ for all the other elements $\mathcal{I}$ of $\mathcal{M}$, $\mathcal{R}'\Vdash_{\rho df_\perp}\langle p,\mathsf{sc},q\rangle$. Otherwise in $\mathcal{R}'$ there should be an $\mathcal{I}'$ with rank 0 s.t. $\mathcal{I}'\nVdash_{\rho df_\perp}(p,\mathsf{sc},q)$ and $\mathcal{I}'\nVdash_{\rho df_\perp}(p,\perp_\mathsf{c},p)$. But then also and $\mathcal{R}_{\min G}$ should contain such $\mathcal{I}'$ at rank 0, but that cannot be the case, as otherwise $\mathcal{R}_{\min G}$ would not satisfy $\langle p,\mathsf{sc},q\rangle$, against the fact that $\mathcal{R}_{\min G}$ is a model of G.

  Hence $\mathcal{R}'\Vdash_{\rho df_\perp}\langle p,\mathsf{sc},q\rangle$.

- We proceed analogously to prove that for every $\langle p,\mathsf{sp},q\rangle\in G^{def}$, $\mathcal{R}'\Vdash_{\rho df_\perp}\langle p,\mathsf{sp},q\rangle$.

Therefore, $\mathcal{R}'$ is a model of G, but by Definition 3.5 we also have $\mathcal{R}'\prec\mathcal{R}_{\min G}$, against the definition of $\mathcal{R}_{\min G}$. Consequently it must be the case that $r(\mathcal{I}_{G^s})=0$, that implies that, for every t, $\mathcal{I}_{G^s}\Vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $\mathcal{I}_{G^s}\Vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$).

Since $\mathcal{I}_{G^s}$ is the canonical model for G^s, $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$).

$\Leftarrow$.) Let us show that given a defeasible graph G, if $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{c},t)$ (resp., $G^s\vdash_{\rho df_\perp}(t,\perp_\mathsf{p},t)$), then a term t is c-exceptional (resp., p-exceptional) w.r.t. G.

This is an immediate consequence of Corollaries 3.9 and 3.11.

This completes the proof. □

Lemma 3.14. *Let $G=G^{str}\cup G^{def}$ be a defeasible graph and $\mathcal{R}_{\min G}=(\mathcal{M},r)$ its minimal ranked model. Then $\mathcal{R}^i_{\min G}=(\mathcal{M}^i,r^i)$ is a model of the subgraph $G^i=G^{str}\cup G^{def}_i$.*

Proof. $\mathcal{R}_{\min G}=(\mathcal{M},r)$ is a model of G. Hence all the interpretations in $\mathcal{M}$ are models of G^{str} and, since $\mathcal{M}^i\subseteq\mathcal{M}$, also $\mathcal{R}^i_{\min G}$ satisfies G^{str}. Concerning the defeasible triples in G^{def}_i, we proceed by contradiction, assuming $\mathcal{R}^i_{\min G}$ is not a model of G^i. So,

let $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$ and $\mathcal{R}_{\min G}^i \nVdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$, that is, there is a $\rho df_\perp$-interpretation $\mathcal{I}$ s.t. $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G}^i)$ and $\mathcal{I} \nVdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$. Since $h_G^c(p) \geqslant i$, $\mathrm{c\_min}(p, \mathcal{R}_{\min G}^i) = \mathrm{c\_min}(p, \mathcal{R}_{\min G})$, and consequently $\mathcal{R}_{\min G} \nVdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$, against the assumption that $\mathcal{R}_{\min G}$ is the minimal model of G. The case $\langle p, \mathrm{sp}, q \rangle \in G_i^{def}$ is proved similarly, which concludes. $\quad\square$

Lemma 3.15. *Given a defeasible graph G, $\mathcal{R}_i^*$ is the minimal model of the subgraph G^i.*

Proof. At first, we prove that $\mathcal{R}_i^*$ is a model of G^i. So, let $\mathcal{R}_{\min G}$ be the minimal model of the graph G. From the definitions of $\mathcal{R}_{\min G}$, $\mathcal{R}_{\min G}^i$ and $\mathcal{R}_i^*$ it is clear that for every $\mathcal{I} \in \mathscr{M}$, $r_i^*(\mathcal{I}) < \infty$ iff $r(\mathcal{I}) < \infty$. Hence $\mathcal{R}_i^* \Vdash_{\rho df_\perp} G^{str}$. Now, let $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$. From the construction of G_i^{def}, $\mathcal{R}_{\min G}^i$ and R_i^*, we have that for $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$ it holds that $h_G^c(p) \geqslant i$. Therefore, since for every $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$ $r(\mathcal{I}) < i$, it must be the case that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ and, thus, for all the ρdf-interpretations $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.

As a consequence, for every $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$, $\mathrm{c\_min}(p, \mathcal{R}_{\min G}) = \mathrm{c\_min}(p, \mathcal{R}_{\min G}^i)$, and since $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$ for every $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$, $\mathcal{R}_i^* \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$ has to hold too.

Analogously, for every $\langle p, \mathrm{sp}, q \rangle \in G_i^{def}$, we have that

- for all the ρdf-interpretations $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_p, p)$;
- $\mathrm{p\_min}(p, \mathcal{R}_{\min G}) = \mathrm{p\_min}(p, \mathcal{R}_{\min G}^i)$; and
- for every $\langle p, \mathrm{sp}, q \rangle \in G_i^{def}$, $\mathcal{R}_i^* \Vdash_{\rho df_\perp} \langle p, \mathrm{sp}, q \rangle$.

Therefore, $\mathcal{R}_i^*$ is a model of G^i.

As next, we have to prove that $\mathcal{R}_i^*$ is in fact the *minimal* model of G^i. To do so, we proceed by contradiction, by assuming that this is not the case. Then there is a model $\mathcal{R}' = (\mathscr{M}, r')$ of G^i s.t. for every $\mathcal{I} \in \mathscr{M}$, $r'(\mathcal{I}) \leqslant r^*(\mathcal{I})$, and there is an $\mathcal{I}' \in \mathscr{M}$ s.t. $r'(\mathcal{I}') < r^*(\mathcal{I}')$. Note that, since $r^*(\mathcal{I}) = 0$ for every $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$, $\mathcal{I}' \in \mathscr{M}^i$ necessarily. We have to prove that such an $\mathcal{R}'$ cannot exist.

Given $\mathcal{R}_{\min G} = (\mathscr{M}, r)$ and $\mathcal{R}' = (\mathscr{M}, r')$, we build a ranked interpretation $\mathcal{R}^+ = (\mathscr{M}, r^+)$ defining r^+ in the following way:

$$r^+(\mathcal{I}) = \begin{cases} r'(\mathcal{I}) + i & \text{if } \mathcal{I} \in \mathscr{M}^i \\ r(\mathcal{I}) & \text{otherwise.} \end{cases}$$

That is,

- $r^+(\mathcal{I}) = r(\mathcal{I})$ for every $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$;
- from the definitions of r^i, r' and r^+ we can conclude that $r^+(\mathcal{I}) \leqslant r(\mathcal{I})$ for every $\mathcal{I} \in \mathscr{M}^i$, and there is an $\mathcal{I}' \in \mathscr{M}^i$ s.t. $r^+(\mathcal{I}') < r(\mathcal{I}')$.

As a consequence, $\mathcal{R}^+ \prec \mathcal{R}_{\min G}$. Also, $\mathcal{R}^+$ is a model of G, because

- all the $\rho df_\perp$-interpretations $\mathcal{I} \in \mathscr{M}$ are models of G^{str}, hence $\mathcal{R}^+ \Vdash_{\rho df_\perp} G^{str}$;
- for all the defeasible triples $\langle p, \mathrm{sc}, q \rangle \in G^{def} \setminus G_i^{def}$, $\mathrm{c\_min}(p, \mathcal{R}^+) = \mathrm{c\_min}(p, \mathcal{R}_{\min G})$, and, since $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$, we have $\mathcal{R}^+ \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$;
- analogously, for all the defeasible triples $\langle p, \mathrm{sp}, q \rangle \in G^{def} \setminus G_i^{def}$, $\mathrm{p\_min}(p, \mathcal{R}^+) = \mathrm{p\_min}(p, \mathcal{R}_{\min G})$, and, since $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, \mathrm{sp}, q \rangle$, we have $\mathcal{R}^+ \Vdash_{\rho df_\perp} \langle p, \mathrm{sp}, q \rangle$;
- for all the defeasible triples $\langle p, \mathrm{sc}, q \rangle \in G_i^{def}$, $\mathrm{c\_min}(p, \mathcal{R}^+) = \mathrm{c\_min}(p, \mathcal{R}')$, and, since $\mathcal{R}' \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$, we have $\mathcal{R}^+ \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$;
- analogously, for all the defeasible triples $\langle p, \mathrm{sp}, q \rangle \in G_i^{def}$, $\mathrm{p\_min}(p, \mathcal{R}^+) = \mathrm{p\_min}(p, \mathcal{R}')$, and, since $\mathcal{R}' \Vdash_{\rho df_\perp} \langle p, \mathrm{sp}, q \rangle$, we have $\mathcal{R}^+ \Vdash_{\rho df_\perp} \langle p, \mathrm{sp}, q \rangle$.

Therefore, $\mathcal{R}^+$ is a model of G, which is impossible, as $\mathcal{R}_{\min G}$ is the minimal model of G and, thus, $\mathcal{R}_i^*$ is the minimal model of the subgraph G^i. $\quad\square$

Lemma 3.16. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and let $\mathcal{R}_{\min G}$ be its minimal model, with $h(\mathcal{R}_{\min G}) = n$. For every $i \leqslant n$ and term p s.t. $h_G^c(p) \geqslant i$ (resp., $h_G^p(p) \geqslant i$) p is c-exceptional (resp., p-exceptional) w.r.t. $\mathcal{R}_{\min G}^i = (\mathscr{M}^i, r^i)$ iff it is c-exceptional (resp., p-exceptional) w.r.t. $\mathcal{R}_i^* = (\mathscr{M}, r_i^*)$.*

Proof. Let $i \leqslant n$, with $\mathcal{R}_i^* = (\mathscr{M}, r^*)$ and $\mathcal{R}_{\min G}^i = (\mathscr{M}^i, r^i)$ be the models of G^i built as described above.

For every $\mathcal{I} \in \mathscr{M}_i$, $r^i(\mathcal{I}) = r^*(\mathcal{I})$. If $\mathcal{I} \in \mathscr{M} \setminus \mathscr{M}^i$ we have seen above that $r^*(\mathcal{I}) = 0$ and $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ for every term p s.t. $h_G^c(p) \geqslant i$. Given these facts, the following statements are equivalent:

- p is not c-exceptional w.r.t. $\mathcal{R}_{\min G}^i$;
- there is an $\mathcal{I} \in \mathscr{M}^i$ s.t. $r^i(\mathcal{I}) = 0$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_c, p)$;
- there is an $\mathcal{I} \in \mathscr{M}^i$ s.t. $r_i^*(\mathcal{I}) = 0$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_c, p)$;
- p is not c-exceptional w.r.t. $\mathcal{R}_i^*$.

Analogously, we can prove that if we consider a term p s.t. $h_G^{\mathsf{p}}(p) \geqslant i$, the following statements are equivalent:

- p is not p-exceptional w.r.t. $\mathcal{R}_{\min G}^i$;
- there is an $\mathcal{I} \in \mathcal{M}^i$ s.t. $r^i(\mathcal{I}) = 0$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_\mathsf{p}, p)$;
- there is an $\mathcal{I} \in \mathcal{M}^i$ s.t. $r_i^*(\mathcal{I}) = 0$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_\mathsf{p}, p)$;
- p is not p-exceptional w.r.t. $\mathcal{R}_i^*$,

which concludes the proof. $\quad\square$

Lemma 3.17. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, and $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ its minimal model, with $h(\mathcal{R}_{\min G}) = n$, and $\langle p, \mathsf{sc}, q\rangle \in G^{def}$. Then,*

- *for every $i < n$, $h_G^{\mathsf{c}}(p) \geqslant i + 1$ iff $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G^i)$;*
- *for $i = n$, $h_G^{\mathsf{c}}(p) = \infty$ iff $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G^i)$.*

Analogously, let $\langle p, \mathsf{sp}, q\rangle \in G^{def}$. Then,

- *for every $i < n$, $h_G^{\mathsf{p}}(p) \geqslant i + 1$ iff $\langle p, \mathsf{sp}, q\rangle \in \mathtt{ExceptionalP}(G^i)$;*
- *for $i = n$, $h_G^{\mathsf{p}}(p) = \infty$ iff $\langle p, \mathsf{sp}, q\rangle \in \mathtt{ExceptionalP}(G^i)$.*

Proof. Let $\langle p, \mathsf{sc}, q\rangle \in G^{def}$.

For $i < n$, the following statements are equivalent:

- $h_G^{\mathsf{c}}(p) \geqslant i + 1$;
- for all $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^{i+1}$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$;
- for all $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^i$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$ and p is c-exceptional w.r.t. $\mathcal{R}_{\min G}^i$;
- for all $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^i$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$ and p is c-exceptional w.r.t. R_i^* (by Lemma 3.16);
- p is c-exceptional w.r.t. G^i (by Lemma 3.15);
- $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G^i)$ (by Corollary 3.13).

For $i = n$, the following statements are equivalent:

- $h_G^{\mathsf{c}}(p) = \infty$;
- for all the $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^\infty$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$;
- for all the $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^n$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$ and p is c-exceptional w.r.t. $\mathcal{R}_{\min G}^n$;
- for all the $\mathcal{I} \in \mathcal{M} \setminus \mathcal{M}^n$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_\mathsf{c}, p)$ and p is c-exceptional w.r.t. R_i^* (by Lemma 3.16);
- p is c-exceptional w.r.t. G^n (by Lemma 3.15);
- $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G^n)$ (by Corollary 3.13).

This proves the first half of the proposition.

For triples $\langle p, \mathsf{sp}, q\rangle \in G^{def}$ the proof is analogous, which concludes. $\quad\square$

Lemma 3.18. *Let G be a defeasible graph, with n being the height of its minimal model. Then, for every $i \leqslant n$, $G^i = G_i^{\mathsf{D}}$.*

Proof. We prove it by induction on the value of i. If $i = 0$, then $G_0^{def} = G^{def} = \mathsf{D}_0$, that is, $G^0 = G_0^{\mathsf{D}}$. Now, assume that $G^i = G_i^{\mathsf{D}}$ holds for all $i < n$, which implies also $G_i^{def} = \mathsf{D}_i$. By Lemma 3.17, for every $\langle p, \mathsf{sc}, q\rangle \in G_i^{def}$, $\langle p, \mathsf{sc}, q\rangle \in G_{i+1}^{def}$ iff $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G^i)$ iff $\langle p, \mathsf{sc}, q\rangle \in \mathtt{ExceptionalC}(G_i^{\mathsf{D}})$, that in turn is equivalent to $\langle p, \mathsf{sc}, q\rangle \in \mathsf{D}_{i+1}$.

Analogously, by Lemma 3.17, we have that for every $\langle p, \mathsf{sp}, q\rangle \in G_i^{def}$, $\langle p, \mathsf{sp}, q\rangle \in G_{i+1}^{def}$ iff $\langle p, \mathsf{sp}, q\rangle \in \mathtt{ExceptionalP}(G^i)$ iff $\langle p, \mathsf{sp}, q\rangle \in \mathtt{ExceptionalP}(G_i^{\mathsf{D}})$, that in turn is equivalent to $\langle p, \mathsf{sp}, q\rangle \in \mathsf{D}_{i+1}$, which concludes. $\quad\square$

Lemma 3.21. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, $\mathrm{r}(G) = \{\mathsf{D}_0, \dots, \mathsf{D}_n, \mathsf{D}_\infty\}$ its ranking, and $\mathcal{R}_{\min G} = (\mathcal{M}, r)$ its minimal model, with $\mathcal{I} \in \mathcal{M}$. Then $\mathcal{I} \in \mathcal{M}_\mathbb{N}$ iff $\mathcal{I} \Vdash_{\rho df_\perp} G^{str} \cup \{(p, \perp_\mathsf{c}, p) \mid \langle p, \mathsf{sc}, q\rangle \in \mathsf{D}_\infty\} \cup \{(p, \perp_\mathsf{p}, p) \mid \langle p, \mathsf{sp}, q\rangle \in \mathsf{D}_\infty\}$.*

Proof. $\Rightarrow$) From the construction of $G = G^{str} \cup G^{def}$ and Proposition 3.19, it is obvious that if $\mathcal{I} \in \mathcal{M}_\mathbb{N}$ then $\mathcal{I} \Vdash_{\rho df_\perp} G^{str} \cup \{(p, \perp_\mathsf{c}, p) \mid \langle p, \mathsf{sc}, q\rangle \in \mathsf{D}_\infty\} \cup \{(p, \perp_\mathsf{p}, p) \mid \langle p, \mathsf{sp}, q\rangle \in \mathsf{D}_\infty\}$.

$\Leftarrow$) We proceed by contradiction. Assume there is a $\rho df_\perp$-interpretation $\mathcal{I}' \in \mathcal{M}$ s.t. $r(\mathcal{I}') = \infty$ and $\mathcal{I}' \Vdash_{\rho df_\perp} G^{str} \cup \{(p, \perp_\mathsf{c}, p) \mid \langle p, \mathsf{sc}, q\rangle \in \mathsf{D}_\infty\} \cup \{(p, \perp_\mathsf{p}, p) \mid \langle p, \mathsf{sp}, q\rangle \in \mathsf{D}_\infty\}$.

Assume $h(\mathcal{R}_{\min G}) = n$ (see Definition 3.7), and let $\mathcal{R}' = (\mathcal{M}, r')$ be a ranked interpretation where r' is defined in the following way:

$$r'(\mathcal{I}) = \begin{cases} r(\mathcal{I}) & \text{if } \mathcal{I} \in \mathscr{M}_{\mathbb{N}} \\ n+1 & \text{if } \mathcal{I} = \mathcal{I}' \\ \infty & \text{otherwise.} \end{cases}$$

Informally, $\mathcal{R}'$ has been obtained from $\mathcal{R}_{\min G}$ simply by moving $\mathcal{I}'$ from rank ∞ to the top of $\mathscr{M}_{\mathbb{N}}$. Clearly $\mathcal{R}' \prec \mathcal{R}_{\min G}$, since for every $\mathcal{I} \in \mathscr{M}$, $r'(\mathcal{I}) \leqslant r(\mathcal{I})$, and $r'(\mathcal{I}') < r(\mathcal{I}')$.

It is easy to check that $\mathcal{R}'$ is a model of G: every model with a finite rank satisfies G^{str}; for every $\langle p, \mathrm{sc}, q \rangle \in D_i$, for $i < \infty$, $\mathrm{c\_min}(p, \mathcal{R}') = \mathrm{c\_min}(p, \mathcal{R}_{\min G})$, and consequently $\mathcal{R}' \Vdash_{\rho df_\perp} \langle p, \mathrm{sc}, q \rangle$. Analogously for every $\langle p, \mathrm{sp}, q \rangle \in D_i$, for $i < \infty$. For every $\langle p, \mathrm{sc}, q \rangle \in D_\infty$, every model with a finite rank satisfies $(p, \perp_c, p)$; analogously for every $\langle p, \mathrm{sc}, q \rangle \in D_\infty$. Hence $\mathcal{R}'$ is a model of G and $\mathcal{R}' \prec \mathcal{R}_{\min G}$, against the assumption that $\mathcal{R}_{\min G}$ is the minimal model of G.

As a consequently for every $\mathcal{I}$, $\mathcal{I} \Vdash_{\rho df_\perp} G^{str} \cup \{(p, \perp_c, p) \mid \langle p, \mathrm{sc}, q \rangle \in D_\infty\} \cup \{(p, \perp_p, p) \mid \langle p, \mathrm{sp}, q \rangle \in D_\infty\}$ implies $\mathcal{I} \in \mathscr{M}_{\mathbb{N}}$. $\quad\square$

Lemma 3.23. *Let T be a proof tree from a graph H to a triple (p, sc, q), and let $\mathcal{I}$ be a model of H. If $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, s)$ for some triple $(s, \mathrm{sc}, o) \in H$, then $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.*

Proof. The proof proceeds by induction on the depth of the proof tree T.

Case $d(T) = 0$. In this case, $H = \{(p, \mathrm{sc}, q)\}$. Consequently $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ obviously implies $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.

Case $d(T) = 1$. The only way of deriving a triple of form (p, sc, q) is using the rule (3a), as shown in the proof of Lemma 3.4. Hence the proof tree consists of an instantiation of rule (3a) with (p, sc, q) as root.

$$(3a) \quad \frac{(p, \mathrm{sc}, t), (t, \mathrm{sc}, q)}{(p, \mathrm{sc}, q)}$$

Now, assume that $\mathcal{I} \Vdash_{\rho df_\perp} (A, \perp_c, A)$ holds for the antecedent of at least one of the two premises, *i.e.*, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ or $\mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_c, t)$ holds. We have to prove that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$.

Case (1). $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$. The result is immediate.

Case (2). $\mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_c, t)$. Then together with $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathrm{sc}, t)$ we derive $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, by the soundness of rule (5b).

Case $d(T) = n + 1$. Assume that the proposition holds for all the proof trees with depth $m \leqslant n$, with $n > 1$. Let us show that it holds also for trees of depth $n + 1$. So, let T be a proof tree from H to (p, sc, q) with depth $n + 1$. The last step in T must correspond to an instantiation of the rule (3a) with (p, sc, q) as root:

$$(3a) \quad \frac{(p, \mathrm{sc}, t), (t, \mathrm{sc}, q)}{(p, \mathrm{sc}, q)}$$

Hence T has two immediate subtrees: T', having (p, sc, t) as root, and T'', having (t, sc, q) as root; each of them has a depth of at most n. Since we assume that the proposition holds for trees of depth at most n, if $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, s)$ for some $(s, \mathrm{sc}, o) \in H$, then by induction hypothesis either $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ (if (s, sc, o) appears as a leaf in T') or $\mathcal{I} \Vdash_{\rho df_\perp} (t, \perp_c, t)$ (if (s, sc, o) appears as a leaf in T''). Likewise case $d(T) = 1$, in both the cases we can conclude $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, which concludes the proof. $\quad\square$

Lemma 3.24. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph, and let $\mathrm{r}(G) = \{D_0, \ldots, D_n, D_\infty\}$ be its ranking. For any pair of terms p, q s.t. $h_G^c(p) \leqslant n$,*

$$G \models_{\min} \langle p, \mathrm{sc}, q \rangle \text{ iff } G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q),$$

where D^p is defined as in the $\mathtt{DefMinEntailmentC}$ procedure.

Proof. Consider $G = G^{str} \cup G^{def}$ and let $\mathcal{R}_{\min G} = (\mathscr{M}, r)$ be its minimal model. Let $h_G^c(p) = k$, with $k \leqslant n$. From Corollary 3.20 we can conclude that in the $\mathtt{DefMinEntailmentC}$ procedure, $G' \nvdash_{\rho df_\perp} (p, \perp_c, p)$ for $j = k$, and $D^p = \{\langle r, \mathrm{sc}, s \rangle \mid \langle r, \mathrm{sc}, s \rangle \in D_k \setminus D_{k+1}\}$.

We have to prove that $G \models_{\min} \langle p, \mathrm{sc}, q \rangle$ iff $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$.

$\Leftarrow$.) Let us show that $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$ implies $G \models_{\min} \langle p, \mathrm{sc}, q \rangle$. Assume $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$. In order to prove $G \models_{\min} \langle p, \mathrm{sc}, q \rangle$ we need to show that for every $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G})$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$ holds.

We proceed by contradiction. So, assume that $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G})$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \mathrm{sc}, q)$. $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G})$ implies $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_c, p)$. Also, since $h_G^c(p) = k$, $r(\mathcal{I}) = k$ follows. Since $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$, $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_c, p)$ and $\mathcal{I} \Vdash_{\rho df_\perp} G^{str}$ (Lemma 3.4), then there is at least a triple in $(D^p)^s$ that is not satisfied by $\mathcal{I}$. By Lemma 3.4 such a triple must be of the kind (s, sc, t). In particular, $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$ implies that there must be a graph $H \subseteq G^{str} \cup (D^p)^s$ s.t. there is a proof tree T proving (p, sc, q) from H, and there is some triple $(s, \mathrm{sc}, t) \in H \cap (D^p)^s$ s.t. $\mathcal{I} \nVdash_{\rho df_\perp} (s, \mathrm{sc}, t)$. As $\langle s, \mathrm{sc}, t \rangle \in D_k \setminus D_{k+1}$, by Proposition 3.19 we have $h_G^c(s) = k$. $h_G^c(s) = k$ and $r(\mathcal{I}) = k$ imply that if $\mathcal{I} \nVdash_{\rho df_\perp} (s, \mathrm{sc}, t)$, then $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, s)$ ($\mathcal{I} \notin \mathrm{c\_min}(s, \mathcal{R}_{\min G})$), otherwise $\mathcal{R}_{\min G}$ would not be a model of G. However, by Lemma 3.23, $\mathcal{I} \Vdash_{\rho df_\perp} (s, \perp_c, s)$ implies $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$, against the hypothesis that $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G})$. Therefore, we must conclude that $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$.

$\Rightarrow$.) Let us show that $G \models_{\min} \langle p, \mathrm{sc}, q \rangle$ implies $G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$. Let $h_G^c(p) = k$, with $k \leqslant n$. $G \models_{\min} \langle p, \mathrm{sc}, q \rangle$ means that, given the minimal model $\mathcal{R}_{\min G} = (\mathscr{M}, r)$, for $\mathcal{I} \in \mathscr{M}$ s.t. $r(\mathcal{I}) = k$ and $\mathcal{I} \nVdash_{\rho df_\perp} (p, \perp_c, p)$, $\mathcal{I} \in \mathrm{c\_min}(p, \mathcal{R}_{\min G})$ and $\mathcal{I} \Vdash_{\rho df_\perp} (p, \mathrm{sc}, q)$.

Now, consider the graph $G^* := G^{str} \cup (D_k \setminus D_{k+1})^s \cup \{(r \perp_c r) \mid \langle r, sc, t \rangle \in D_{k+1}\} \cup \{(r \perp_p r) \mid \langle r, sp, t \rangle \in D_{k+1}\}$. That is, G^* contains all the strict triples in G, G^{str}, all the defeasible triples that are satisfied at height k, that is, $D_k \setminus D_{k+1}$, and for all the triples that are exceptional in k, the set $\{(r \perp_c r) \mid \langle r, sc, t \rangle \in D_{k+1}\} \cup \{(r \perp_p r) \mid \langle r, sp, t \rangle \in D_{k+1}\}$.

Let $\mathcal{I}_{G^*}$ be the characteristic $\rho df_\perp$ model of G^*, built as defined in Lemma 2.3. We know that $\mathcal{I}_{G^*}$ satisfies exactly $Cl(G^*)$. Also, $\mathcal{I}_{G^*} \in \mathcal{M}$, as it can be checked by the definition in Section 3.2 of the ranked interpretations in $\mathfrak{R}_G$.

Since $\mathcal{I}_{G^*}$ satisfies $\mathcal{I} \Vdash_{\rho df_\perp} G^{str} \cup \{(r, \perp_c, r) \mid \langle r, sc, t \rangle \in D_\infty\} \cup \{(r, \perp_p, r) \mid \langle r, sp, t \rangle \in D_\infty\}$, by Lemma 3.21 it must be in $\mathcal{M}_\mathbb{N}$. Moreover, it must hold that $r(\mathcal{I}_{G^*}) = k$, since:

- for every triple $\langle s, sc, t \rangle \in D_k \setminus D_{k+1}$ (resp., $\langle s, sp, t \rangle \in D_k \setminus D_{k+1}$) $\mathcal{I}_{G^*}$ does not satisfy $(s, \perp_c, s)$ (resp., $(s, \perp_p, s)$), while $h_G^c(s) = k$ (resp., $h_G^p(s) = k$). Hence it cannot be $r(\mathcal{I}_{G^*}) < k$.
- $\mathcal{I}_{G^*}$ is compatible with $r(\mathcal{I}_{G^*}) = k$, since it satisfies all the triples in $D_k \setminus D_{k+1}$ and for every $\langle s, sc, t \rangle \in D_{k+1}$, $\mathcal{I}_{G^*} \Vdash_{\rho df_\perp} (s, \perp_c, s)$ (resp., for every $\langle s, sp, t \rangle \in D_{k+1}$, $\mathcal{I}_{G^*} \Vdash_{\rho df_\perp} (s, \perp_p, s)$).
- $\mathcal{R}_{\min G}$ assigns the minimal rank to every $\rho df_\perp$ interpretation, and since $\mathcal{I}_{G^*}$ is compatible with $r(\mathcal{I}_{G^*}) = k$ and not with $r(\mathcal{I}_{G^*}) < k$, we can conclude $r(\mathcal{I}_{G^*}) = k$.

$h_G^c(p) = k$ implies that $G_k^D \not\models_{\rho df_\perp} (p, \perp_c, p)$ (Corollary 3.20). Since $G_k^D \models_{\rho df_\perp} G^*$, $G^* \not\models_{\rho df_\perp} (p, \perp_c, p)$.
In summary, we have the following situation:

- $r(\mathcal{I}_{G^*}) = k$;
- $h_G^c(p) = k$;
- since $G^* \not\models_{\rho df_\perp} (p, \perp_c, p)$ and $\mathcal{I}_{G^*}$ is the characteristic model of G^*, $\mathcal{I}_{G^*} \not\Vdash_{\rho df_\perp} (p, \perp_c, p)$.

Hence $\mathcal{I}_{G^*} \in c\_min(p)$. This, together with $G \models_{\min} \langle p, sc, q \rangle$, implies $\mathcal{I}_{G^*} \Vdash_{\rho df_\perp} (p, sc, q)$. Being $\mathcal{I}_{G^*}$ the characteristic model of G^*, $\mathcal{I}_{G^*}$ satisfies a triple (s, sc, t) iff $(s, sc, t) \in Cl(G^*)$. Hence $\mathcal{I}_{G^*} \Vdash_{\rho df_\perp} (p, sc, q)$ implies $G^* \vdash_{\rho df_\perp} (p, sc, q)$.

By Lemma 3.4, we know that to derive a triple of form (A, sc, B) from a graph it is sufficient to consider only the triples in the same graph with the same form, and in G^* all such triples are in $G^{str} \cup (D_k \setminus D_{k+1})^s$, that is, $G^{str} \cup (D^p)^s$. Hence $G^* \vdash_{\rho df_\perp} (p, sc, q)$ implies

$$G^{str} \cup (D^p)^s \vdash_{\rho df_\perp} (p, sc, q) \,,$$

which concludes. $\quad\square$

Theorem 3.25. *Let $G = G^{str} \cup G^{def}$ be a defeasible graph and let $\langle p, sc, q \rangle$ be a defeasible triple. Then,*

$$G \models_{\min} \langle p, sc, q \rangle \text{ iff } \texttt{DefMinEntailmentC}(G, r(G), \langle p, sc, q \rangle) \,.$$

Proof. Let $\mathcal{R}_{\min G}$ be the minimal model of G, with $h(\mathcal{R}_{\min G}) = n$. Given $\langle p, sc, q \rangle$ we have two possible cases:

Case $h_G^c(p) \leqslant n$. The result is guaranteed by Lemma 3.24.

Case $h_G^c(p) = \infty$. By Definition 3.4, $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, sc, q \rangle$. At the same time, if $h_G^c(p) = \infty$, then by definition of the procedure, $\texttt{DefMinEntailmentC}(G, r(G), \langle p, sc, q \rangle)$ must be the case. $\quad\square$

Proposition 3.31. $\models_{\min}$ *satisfies (LLE_c), (RW_c), (LLE_p), and (RW_p).*

Proof. We prove (LLE_c) and (RW_c). The proofs for (LLE_p) and (RW_p) are analogous.

(LLE_c). Let $G \models_{\min}(p, sc, q)$, $G \models_{\min}(q, sc, p)$, and $G \models_{\min} \langle p, sc, r \rangle$. $G \models_{\min}(p, sc, q)$ and $G \models_{\min}(q, sc, p)$ imply that every $\rho df_\perp$-interpretation in $\mathcal{R}_{\min G}$ satisfies (p, sc, q) and (q, sc, p). Since the semantics is sound w.r.t. the derivation rules, by rule $(5b)$ we have that for every $\rho df_\perp$-interpretation $\mathcal{I}$ in $\mathcal{R}_{\min G}$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, \perp_c, p)$ iff $\mathcal{I} \Vdash_{\rho df_\perp} (q, \perp_c, q)$. Consequently $c\_min(p, \mathcal{R}_{\min G}) = c\_min(q, \mathcal{R}_{\min G})$.
Let $\mathcal{I}$ be in $c\_min(p, \mathcal{R}_{\min G})$. Since $G \models_{\min} \langle p, sc, r \rangle$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, sc, r)$. We also have $\mathcal{I} \Vdash_{\rho df_\perp} (q, sc, p)$, and, being $\mathcal{I}$ sound w.r.t. rule $(3a)$, we conclude $\mathcal{I} \Vdash_{\rho df_\perp} (q, sc, r)$. $c\_min(p, \mathcal{R}_{\min G}) = c\_min(q, \mathcal{R}_{\min G})$ implies $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle q, sc, r \rangle$, that is, $G \models_{\min} \langle q, sc, r \rangle$.
(RW_c). $G \models_{\min}(q, sc, r)$ implies that every $\rho df_\perp$-interpretation in $\mathcal{R}_{\min G}$ satisfies (q, sc, r). Let $\mathcal{I}$ be in $c\_min(p, \mathcal{R}_{\min G})$. Since $G \models_{\min} \langle p, sc, q \rangle$, $\mathcal{I} \Vdash_{\rho df_\perp} (p, sc, q)$. Hence $\mathcal{I} \Vdash_{\rho df_\perp} (p, sc, r)$, that implies $\mathcal{R}_{\min G} \Vdash_{\rho df_\perp} \langle p, sc, r \rangle$, that is, $G \models_{\min} \langle p, sc, r \rangle$. $\quad\square$

Appendix C. Proofs for Section 4

Proposition 4.5. Cl_{in} *does not satisfy* Idempotence. *That is, there is a graph G such that*

$$Cl_{in}(Cl_{in}(G)) \neq Cl_{in}(G) \,.$$

Proof. To prove Proposition 4.5 it is sufficient to consider the following example, in which we apply the operator Cl_{in} to a graph G, and then we show that applying the inheritance completion to $Cl_{in}(G)$ we obtain new defeasible triples, so $(Cl_{in}(G))_{in} \neq Cl_{in}(G)$, which implies the proposition.
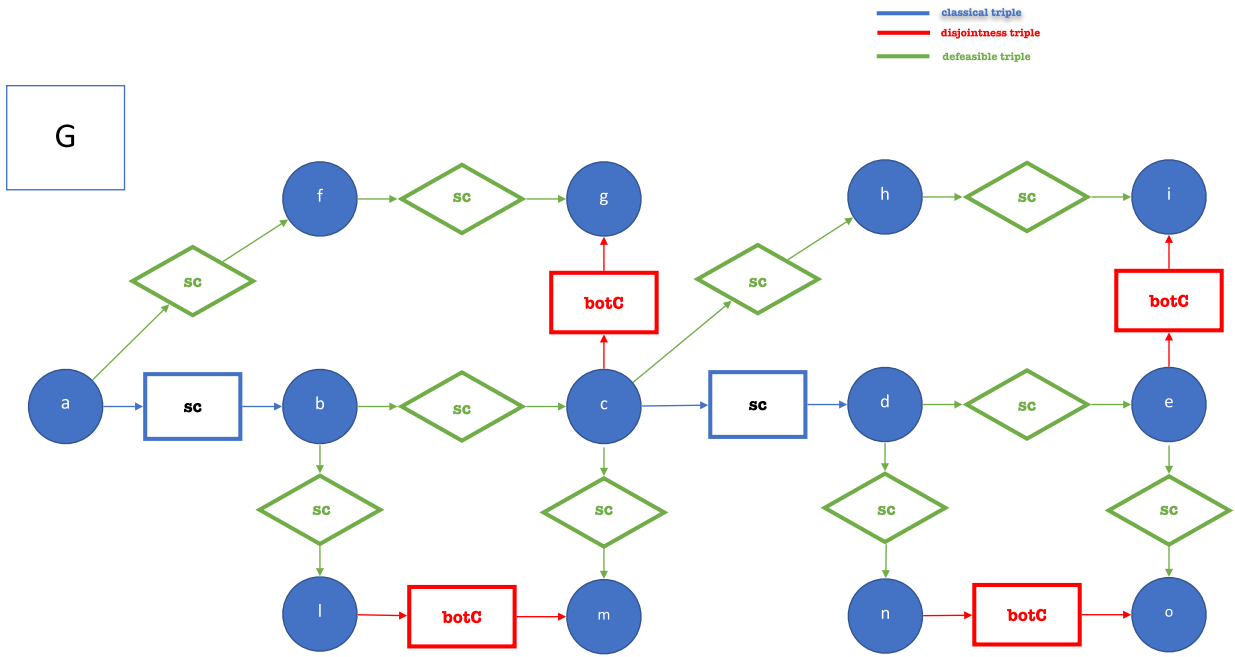
Fig. C.2. The graph G.

So, consider the following graph $G = G^{str} \cup G^{def}$ with

$$G^{str} = \{(a, \mathsf{sc}, b), (c, \mathsf{sc}, d), (c, \perp_c, g), (l, \perp_c, m), (e, \perp_c, i), (n, \perp_c, o)\}$$

$$G^{def} = \{\langle a, \mathsf{sc}, f\rangle, \langle f, \mathsf{sc}, g\rangle, \langle b, \mathsf{sc}, c\rangle, \langle b, \mathsf{sc}, l\rangle, \langle c, \mathsf{sc}, m\rangle, \langle c, \mathsf{sc}, h\rangle, \langle h, \mathsf{sc}, i\rangle, \langle d, \mathsf{sc}, e\rangle,$$

$$\langle d, \mathsf{sc}, n\rangle, \langle e, \mathsf{sc}, o\rangle \} .$$

The graph is represented in Fig. C.2. In order for G^{str} to be closed under `StrictMinEntailment` it is sufficient to add the symmetric version of the triples $(A, \perp_c, B)$, obtained through the application of rule (5a).

At first, we proceed to determine $\mathrm{Cl}_{in}(G)$. As a first step we determine G_{in} by applying the procedure `InheritanceCompletion(G)`, and obtaining the following completion:

$$G_{in} = G \cup \{\langle a, \mathsf{sc}, l\rangle, \langle b, \mathsf{sc}, h\rangle, \langle b, \mathsf{sc}, d\rangle, \langle b, \mathsf{sc}, n\rangle, \langle c, \mathsf{sc}, n\rangle\} .$$

The new triples in G_{in} are represented in orange in Fig. C.3. Please note that $\langle a, \mathsf{sc}, e\rangle$ is not in G_{in}: $\Delta_{a,e} = \{\langle a, \mathsf{sc}, f\rangle, \langle f, \mathsf{sc}, g\rangle, \langle b, \mathsf{sc}, c\rangle,$ $\langle c, \mathsf{sc}, h\rangle, \langle h, \mathsf{sc}, i\rangle, \langle d, \mathsf{sc}, e\rangle\}$, and $G^{str} \cup \Delta_{a,e} \not\vdash_{\min} \langle a, \mathsf{sc}, e\rangle$.

Determining $\mathrm{Cl}_{in}(G)$ corresponds to determine $\mathrm{Cl}_{\min}(G_{in})$. The latter, represented in Fig. C.4, is

$$\mathrm{Cl}_{in}(G) = G_{in} \cup \{\langle a, \mathsf{sc}, c\rangle, \langle c, \mathsf{sc}, e\rangle\} .$$

$\langle a, \mathsf{sc}, e\rangle$ is still not derivable, since $G_{in} \not\vdash_{\min} \langle a, \mathsf{sc}, e\rangle$.

Now, if Cl_{in} satisfies Idempotence, we would have $\mathrm{Cl}_{in}(\mathrm{Cl}_{in}(G)) = \mathrm{Cl}_{in}(G)$, that implies that the inheritance completion of $\mathrm{Cl}_{in}(G)$, namely $(\mathrm{Cl}_{in}(G))_{in}$, does not add any new triples to $(\mathrm{Cl}_{in}(G))$, i.e., $(\mathrm{Cl}_{in}(G))_{in} = \mathrm{Cl}_{in}(G)$. However, once we determine $(\mathrm{Cl}_{in}(G))_{in}$, we obtain a new defeasible triple, as indicated in Fig. C.5:

$$(\mathrm{Cl}_{in}(G))_{in} = \mathrm{Cl}_{in}(G) \cup \{\langle a, \mathsf{sc}, e\rangle\} ,$$

which concludes. □

Lemma 4.7. *Consider a defeasible graph $G = G^{str} \cup G^{def}$ and two nodes, p and q, in G. $\Delta_{p,q}$ can be determined in $O(|G|^2)$ time.*

Proof. In order to determine the set $\Delta_{p,q}$, given any two nodes p, q in a graph G, we adapt the analogous method presented for ducts in [14, Section 3.1.6] to our case.

Specifically, given that the construction of paths is independent w.r.t. the kind of triples that are involved, we can analyse the problem using simple directed graphs. Given a graph G, it is sufficient to define the correspondent directed graph $\mathcal{G} = \langle V, E\rangle$ in the following way:
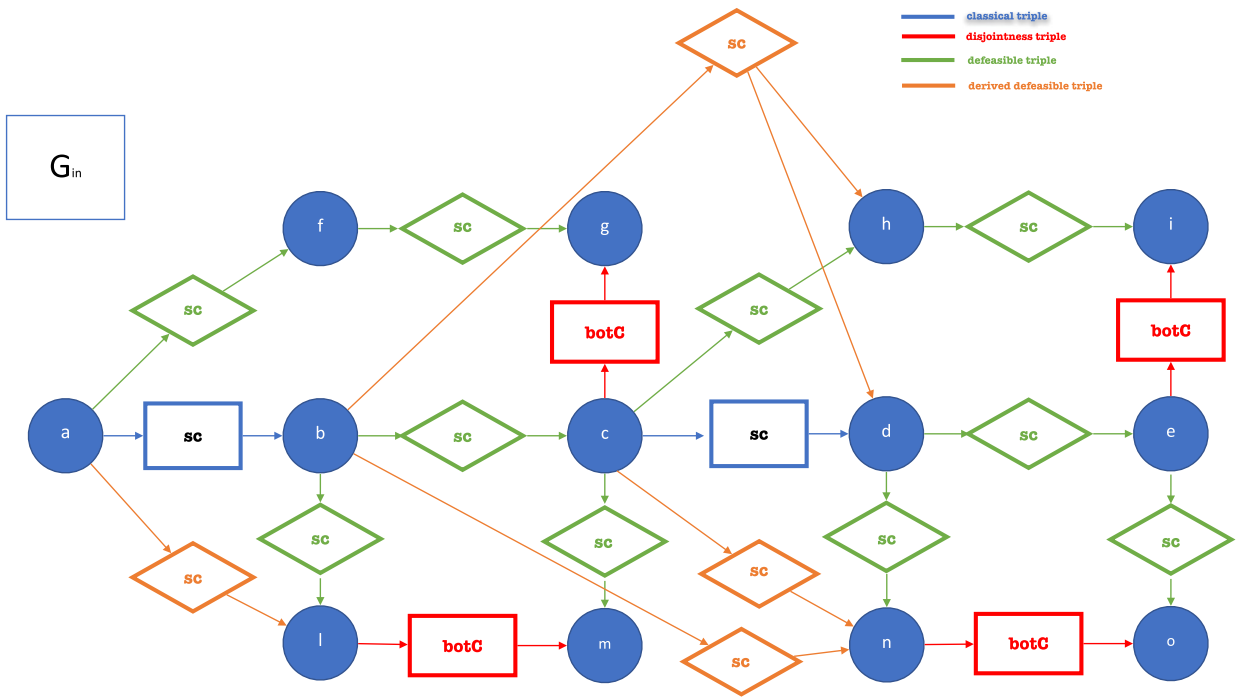
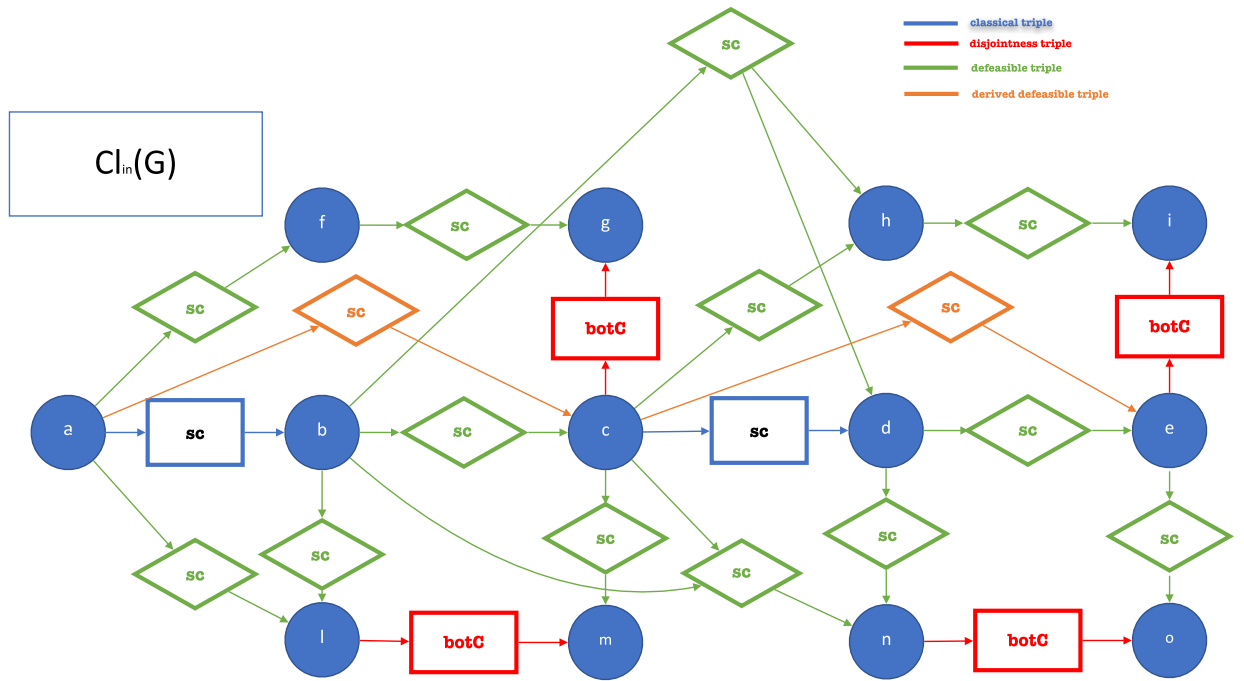**Fig. C.3.** The inheritance completion of the graph G.

Fig. C.4. The inheritance based closure of the graph G.

- V is the set of terms that appear in the first or third position in the triples in G;
- E is a set of directed links $\langle s, t \rangle$, with $s, t \in V$, s.t. $\langle s, t \rangle \in E$ iff $[\![s, r, t]\!] \in G$, for any r.

Once we have defined $\mathcal{G}$, let us recall a well-known result in graph theory saying that in a directed graph, given two nodes p and q, determining if there is a path from p to q can be determined in time $\mathcal{O}(|V| + |E|)$, e.g., using BFS (Breadth First Search) [17]. Now, the following argument shows that indeed $\Delta_{p,q}$ can be determined in polynomial time.
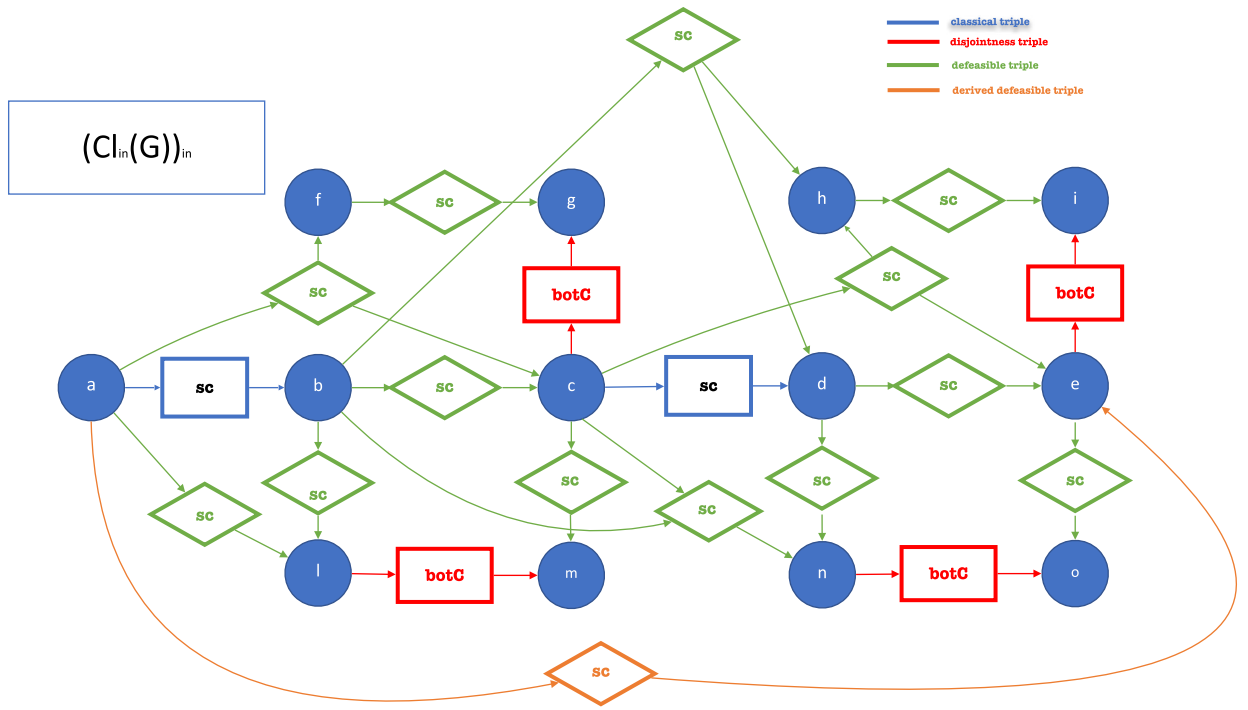
Fig. C.5. An extra inheritance completion of the graph $Cl_{in}(G)$.

At first, we check if there is a path between p and q. If not, then $\Delta_{p,q} = \emptyset$. Otherwise, we call the procedure **Delta(p)** below:

Delta(p): for each outgoing edge $\langle p, x \rangle$ of p, such that both $\langle p, x \rangle$ and x are not marked, do: if there is a path between x and q then mark both $\langle p, x \rangle$ and x, and recursively, call **Delta(x)**.

Once finished, $\Delta_{p,q}$ can immediately be built from the marked edges.

Note that each edge is marked once and each node is marked (*i.e.*, explored) once and, thus, the algorithm is bounded polynomially by the size of the graph $\mathcal{G}$: the procedure **Delta(p)** needs to be called at most $|E|$ times, hence the entire construction of $\Delta_{p,q}$ requires $\mathcal{O}(|E|(|V| + |E|))$. Since $|V| \leqslant 2|E|$ and $|E| \leqslant |G|$, $|E|(|V| + |E|) \leqslant |G|(3|G|) = 3|G|^2$. Hence the construction of $\Delta_{p,q}$ runs in time $\mathcal{O}(|G|^2)$. $\square$

Proposition 4.8. *Consider a defeasible graph* $G = G^{str} \cup G^{def}$. *Assume that* G^{str} *is closed under* StrictMinEntailment. *Then the procedure* InheritanceCompletion(G) *runs in time* $O(|G^{def}||G|^4)$.

Proof. First, we analyse the cost of computing the lines 3-8 in InheritanceCompletion(G). That is, given a graph G and a pair $\langle p, q \rangle$ of nodes in G, what is the cost of calculating whether $\langle p, \mathtt{sc}, q \rangle$ or $\langle p, \mathtt{sp}, q \rangle$ need to be added in G_{in}. We proceed first by calculating $\Delta_{p,q}$ ($O(|G|^2)$, see Lemma 4.7), followed by DefMinEntailmentC ($O(|G^{def}||G|^2)$, see Proposition 3.43) and DefMinEntailmentC ($O(|G^{def}||G|^2)$, see Proposition 3.44). Hence, combining these 3 procedures, these lines can be computed in time $O(|G^{def}||G|^2)$.

We have to repeat such lines for every pair $p, q \in G$. Therefore, InheritanceCompletion(G) runs in time $O(|G^{def}||G|^4)$. $\square$

Corollary 4.9. *Consider a defeasible graph* $G = G^{str} \cup G^{def}$. *Then* $G \vdash_{in} [\![p, o, q]\!]$ *can be decided in polynomial time.*

Proof. The claim follows from the facts that: (*i*) by Proposition 3.39, the closure of G^{str} under procedure StrictMinEntailment can be computed in polynomial time; (*ii*) InheritanceCompletion(G) needs to be run only once for any graph G and, thus, requires polynomial time (see Proposition 4.8)); and (*iii*) once we have obtained G_{in} from InheritanceCompletion(G), following Definition 4.2, deciding whether $G \vdash_{in} [\![p, o, q]\!]$ corresponds to deciding whether $G_{in} \vDash_{\min} [\![p, o, q]\!]$, which is tractable (see Corollary 3.45). Combining the three facts above, $G \vdash_{in} [\![p, o, q]\!]$ can be decided in polynomial time, which concludes. $\square$

References

[1] A. Analyti, G. Antoniou, C.V. Damásio, G. Wagner, Extended RDF as a semantic foundation of rule markup languages, J. Artif. Intell. Res. 32 (2008) 37–94.

[2] A. Analyti, C.V. Damásio, G. Antoniou, Extended RDF: computability and complexity issues, Ann. Math. Artif. Intell. 75 (3–4) (2015) 267–334.

[3] G. Antoniou, Nonmonotonic rule systems on top of ontology layers, in: Proceedings of the 1st International Semantic Web Conference (ISWC-02), in: Lecture Notes in Computer Science, vol. 2663, Springer Verlag, 2002, pp. 394–398.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, 2 edition, Cambridge University Press, 2007.

[5] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, H. Prade, Inconsistency management and prioritized syntax-based entailment, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93), vol. 1, Morgan Kaufmann Publishers Inc., 1993, pp. 640–645.

[6] A. Billig, A triple-oriented approach for integrating higher-order rules and external contexts, in: Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems (RR-08), in: Lecture Notes in Computer Science, vol. 5341, Springer, 2008, pp. 214–221.

[7] P. Bonatti, Rational closure for all description logics, Artif. Intell. 274 (2019) 197–223.

[8] P. Bonatti, M. Faella, I. Petrova, L. Sauro, A new semantics for overriding in description logics, Artif. Intell. 222 (2015) 1–48.

[9] R. Booth, J.B. Paris, A note on the rational closure of knowledge bases with both positive and negative knowledge, J. Log. Lang. Inf. 7 (2) (1998) 165–190.

[10] K. Britz, G. Casini, T. Meyer, K. Moodley, U. Sattler, I. Varzinczak, Rational defeasible reasoning for description logics, Technical report, University of Cape Town, South Africa, 2018, https://tinyurl.com/yc55y7ts.

[11] K. Britz, G. Casini, T. Meyer, K. Moodley, U. Sattler, I. Varzinczak, Principles of KLM-style defeasible description logics, ACM Trans. Comput. Log. 22 (1) (2021) 1–46.

[12] G. Casini, U. Straccia, Rational closure for defeasible description logics, in: Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA-10), in: Lecture Notes in Artificial Intelligence, vol. 6341, Springer Verlag, 2010, pp. 77–90.

[13] G. Casini, U. Straccia, Lexicographic closure for defeasible description logics, in: Proceedings of the 8th Australasian Ontology Workshop (AOW-12), in: CEUR Workshop Proceedings, vol. 969, 2012, pp. 28–39, CEUR.org.

[14] G. Casini, U. Straccia, Defeasible inheritance-based description logics, J. Artif. Intell. Res. 48 (2013) 415–473.

[15] G. Casini, T. Meyer, K. Moodley, R. Nortjé, Relevant closure: a new form of defeasible reasoning for description logics, in: Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA-14), in: Lecture Notes in Computer Science, vol. 8761, Springer, 2014, pp. 92–106.

[16] G. Casini, U. Straccia, T. Meyer, A polynomial time subsumption algorithm for nominal safe $\mathcal{ELO}_\perp$ under rational closure, Inf. Sci. 501 (2019) 588–620.

[17] T.H. Cormen, C. Stein, R.L. Rivest, C.E. Leiserson, Introduction to Algorithms, 2nd edition, McGraw-Hill Higher Education, 2001.

[18] T. Eiter, T. Lukasiewicz, Default reasoning from conditional knowledge bases: complexity and tractable cases, Artif. Intell. 124 (2) (2000) 169–241.

[19] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, H. Tompits, Combining answer set programming with description logics for the semantic web, Artif. Intell. 172 (12–13) (2008) 1495–1539.

[20] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, Well-founded semantics for description logic programs in the semantic web, ACM Trans. Comput. Log. 12 (2) (2011) 11.

[21] M. Freund, Preferential reasoning in the perspective of Poole default logic, Artif. Intell. 98 (1998) 209–235.

[22] L. Giordano, D.T. Dupré, ASP for minimal entailment in a rational extension of $\mathcal{SROEL}$, Theory Pract. Log. Program. 16 (5–6) (2016) 738–754.

[23] L. Giordano, V. Gliozzi, A reconstruction of multipreference closure, Artif. Intell. 290 (2021) 103398.

[24] L. Giordano, V. Gliozzi, N. Olivetti, G.L. Pozzato, A non-monotonic description logic for reasoning about typicality, Artif. Intell. 195 (2013) 165–202.

[25] L. Giordano, V. Gliozzi, N. Olivetti, G.L. Pozzato, Rational closure in $\mathcal{SHIQ}$, in: Proceedings of the 27th International Workshop on Description Logics (DL-14), in: CEUR Workshop Proceedings, vol. 1193, 2014, pp. 543–555, CEUR.org.

[26] L. Giordano, V. Gliozzi, N. Olivetti, G.L. Pozzato, Semantic characterization of rational closure: from propositional logic to description logics, Artif. Intell. 226 (2015) 1–33.

[27] C. Gutiérrez, C.A. Hurtado, A.O. Mendelzon, J. Pérez, Foundations of semantic web databases, J. Comput. Syst. Sci. 77 (3) (2011) 520–541.

[28] L.C. Hill, J.B. Paris, When maximizing entropy gives the rational closure, J. Log. Comput. 13 (1) (2003) 51–68.

[29] A. Hogan, M. Arenas, A. Mallea, A. Polleres, Everything you always wanted to know about blank nodes, J. Web Semant. 27–28 (2014) 42–69.

[30] J.F. Horty, Some direct theories of nonmonotonic inheritance, in: Handbook of Logic in Artificial Intelligence and Logic Programming: Nonmonotonic Reasoning and Uncertain Reasoning, vol. 3, Oxford University Press, 1994, pp. 111–187.

[31] G. Ianni, T. Krennwallner, A. Martello, A. Polleres, A rule system for querying persistent RDFS data, in: Proceedings of the 6th European Semantic Web Conference (ESWC-09), in: Lecture Notes in Artificial Intelligence, vol. 5554, 2009, pp. 857–862.

[32] E. Kontopoulos, N. Bassiliades, G. Antoniou, Deploying defeasible logic rule bases for the semantic web, Data Knowl. Eng. 66 (1) (2008) 116–146.

[33] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, Artif. Intell. 44 (1–2) (1990) 167–207.

[34] D. Lehmann, Another perspective on default reasoning, Ann. Math. Artif. Intell. 15 (1) (1995) 61–82.

[35] D. Lehmann, M. Magidor, What does a conditional knowledge base entail?, Artif. Intell. 55 (1) (1992) 1–60.

[36] T. Lukasiewicz, Expressive probabilistic description logics, Artif. Intell. 172 (6–7) (2008) 852–883.

[37] D. Makinson, General patterns in nonmonotonic reasoning, in: Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3, Oxford University Press, 1994, pp. 35–110.

[38] B. Motik, R. Rosati, Reconciling description logics and rules, J. ACM 57 (5) (2010) 1–62.

[39] S. Muñoz, J. Pérez, C. Gutierrez, Simple and efficient minimal RDFS, Web Semant. Sci. Serv. Agents World Wide Web 7 (3) (2009) 220–234.

[40] J. Pearl, System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning, in: Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning About Knowledge (TARK-90), Morgan Kaufmann Publishers Inc., 1990, pp. 121–135.

[41] M. Pensel, A.Y. Turhan, Reasoning in the defeasible description logic $\mathcal{EL}_\perp$ - computing standard inferences under rational and relevant semantics, Int. J. Approx. Reason. 103 (2018) 28–70.

[42] T. Pham, A. Mileo, M.I. Ali, Towards scalable non-monotonic stream reasoning via input dependency analysis, in: Proceedings of the 33rd IEEE International Conference on Data Engineering (ICDE-17), IEEE Computer Society, 2017, pp. 1553–1558.

[43] G.L. Pozzato, Typicalities and probabilities of exceptions in nonmonotonic description logics, Int. J. Approx. Reason. 107 (2019) 81–100.

[44] G. Qi, A. Hunter, Measuring incoherence in description logic-based ontologies, in: The Semantic Web, in: Lecture Notes in Computer Science, vol. 4825, Springer, 2007, pp. 381–394.

[45] I. Tachmazidis, G. Antoniou, G. Flouris, S. Kotoulas, Scalable nonmonotonic reasoning over RDF data using MapReduce, in: Proceedings of the Joint Workshop on Scalable and High-Performance Semantic Web Systems, in: CEUR Workshop Proceedings, vol. 943, 2012, pp. 75–90, https://ceur-ws.org/.

[46] I. Tachmazidis, G. Antoniou, G. Flouris, S. Kotoulas, L. McCluskey, Large-scale parallel stratified defeasible reasoning, in: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI-12), in: Frontiers in Artificial Intelligence and Applications, vol. 242, IOS Press, 2012, pp. 738–743.

[47] H.J. ter Horst, Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the owl vocabulary, J. Web Semant. 3 (2–3) (2005) 79–115.

[48] D.S. Touretzky, The Mathematics of Inheritance Systems, Pitman, 1986.