Rational Concept Analysis

Lucas Carr

Department of Computer Science

University of Cape Town

crrluc003@myuct.ac.za

1 INTRODUCTION

Formal concept analysis (FCA) provides a framework, grounded in lattice theory, for mathematically reasoning about *formal concepts* and their hierarchies [5, 6, 13]. The matter of *concepts* has largely been of Philosophical concern: the notion of a concept as the dualism between *intension* and *extension* has foundations in Aristotle's *Organon* and, much later on, in the *Logic of Port-Royal* [2, 13]. In this view, the extension of a concept contains to those "things" that one might refer to as instances of the concept. Dually, intension describes the meaning, or *sense*, of a concept.

Formal concept analysis—which adopts this view of concepts—introduces a *formal context* as the structure of data. This is a triple consisting of a finite set of objects G, attributes M, and a binary relation, $I \subseteq G \times M$, which indicates that a particular object has a respective property [5, 6]. A *formal concept* is a pair, made up of the formal concept extension and intension, respectively. The set of all concepts, when ordered by the *sub/super-concept* relation, form the *concept lattice* used for analysis.

Another important topic in FCA is the discovery of *implications* that pertain to, or are *respected* by, a context [5, 13]. Implications are used to express correspondence between (sets of) attributes. The notion of an implication being *valid* in a context is analogous to that of *entailment* in classical logic [6]. As such, *validity* describes a monotonic notion of consequence.

The property of monotonicity in the logical underlying FCA means that it is ill-suited to reason about concepts with *exceptional members*, or *defeasible* implications.

Discussion and work on non-monotonic propositional, first-order, and description logic is a well established topic in artificial intelligence, [4, 7, 9, 11, 14]. However, there does not appear to be any effort to introduce this expressivity to the attribute logic of FCA.

The rest of this proposal is structured as follows: Section 2 is devoted to providing a brief introduction to both FCA (Section 2.1) and KLM-style non-monotonic reasoning (Section 2.2). Section 3 then presents arguments in favour of introducing non-monotonicity into FCA. In turn, Section 4 clarifies the previous section, explaining the current view of how this might be achieved.

2 BACKGROUND

The proceeding subsections provide an introduction to aspects of FCA, classical logic, and the KLM-framework for non-monotonic reasoning which are particularly relevant to the scope of this research.

2.1 Formal Concept Analysis

The starting point of FCA is the *formal context*. This is a triple consisting of a finite set of objects, a finite set of attributes, and a relation which describes when an object g 'has' an attribute m.

Definition 2.1. A *formal context* is a triple, $\mathbb{K} = (G, M, I)$, where G is a finite set of objects, M is a finite set of attributes, and $I \subseteq G \times M$ is an incidence relation such that for an object g and attribute m, $(g, m) \in I$ is interpreted as g *having* m, and $(g, m) \notin I$ as g having $\neg m$.

For contexts of reasonable size, a cross-table is frequently used as a visual aid. Each row represents an object, and each column an attribute. A '×' is used to indicate where an object has an attribute [5, 6]. See Table 1 for an example. For any subset of objects (attributes) we use a derivation operator to describe the attributes (objects) common to all members of that subset. These derivation operators form a anti-tone Galois connection between the power sets $\mathcal{P}(G)$ and $\mathcal{P}(M)$ [5].

Definition 2.2. In a context, $\mathbb{K} = (G, M, I)$, we define two *derivation operators*, both denoted by $(\cdot)'$, as follows:

$$A \subseteq G : \quad A' \coloneqq \{m \in M \mid \forall g \in A : (g, m) \in I\}$$
$$B \subseteq M : \quad B' \coloneqq \{g \in G \mid \forall m \in B : (g, m) \in I\}$$

In case of a singleton set, $\{g\}$, the braces are omitted. If g is an object, then g' describes the *object intent*; if g is an attribute, g' is the *attribute extent*. A derivation operator can be applied to the result of an earlier derivation operator, this is then referred to as a *double-prime operator*, and expressed as $(\cdot)''$. Any set derived from a double-prime operator is closed, and so each derivation operator describes a closure system on G and M, respectively [5, 6, 13]. As such, the double-prime operator $(\cdot)''$ satisfies:

$$\begin{array}{ll} \text{Monotonocity:} & \text{if } A_1 \subseteq A_2 \text{ then } A_1'' \subseteq A_2'' \\ \text{Idempotency:} & A'' = (A'')'' \\ \text{Extensivity:} & A \subseteq A'' \end{array}$$

Definition 2.3. A *formal concept* of a context, $\mathbb{K} = (G, M, I)$, is a pair, (A, B), with $A \subseteq G$ and $B \subseteq M$ and where $A' = B$ and $B' = A$. Then A is the *concept extent* and B is the *concept intent*. The set of all concepts of a context is given by $\mathfrak{B}(G, M, I)$.

There is obvious redundancy in this definition of concepts: if (A, B) is a formal concept, it could equivalently be given by (A, A') or (B', B). Moreover, for an arbitrary set A of objects (*resp.* attributes), A' defines a concept intent (*resp.* extent), and A'' defines a concept extent (*resp.* intent) [6].

Definition 2.4. Let (G, M, I) be a formal context, then for every object, $g \in G$, we define a *object concept* as

$$\gamma g \coloneqq (g'', \ g')$$

and for each attribute, $m \in M$, we define the *attribute concept* as

$$\mu m \coloneqq (m', \ m'')$$

The idea of sub and super-concepts gives rise to a natural partial order on $\mathfrak{B}(G, M, I)$. Specifically, if (A_1, B_1) and (A_2, B_2) are concepts in $\mathfrak{B}(G, M, I)$, then (A_1, B_1) is a *sub-concept* of (A_2, B_2) iff $A_1 \subseteq A_2$ (equivalently, $B_2 \subseteq B_1$). Respectively, (A_2, B_2) would be a *super-concept* of (A_1, B_1) and we say that $(A_1, B_1) \leq (A_2, B_2)$ [5]. Then, $\mathfrak{B}(G, M, I)$ and the relation $\leq$ form a complete lattice called the *concept lattice*. [link to concept lattice in appendix]

Previously, it was mentioned that contexts of a reasonable size have a tidy representation in the form of a cross-table. In case of larger contexts, the concept lattice can instead be inferred from *attribute implications* [5, 6].

Definition 2.5. Let M be an arbitrary attribute-set. An *implication* over M takes the form $B_1 \longrightarrow B_2$ where $B_1, B_2 \subseteq M$. Another set $C \subseteq M$, *respects* the implication iff $B_1 \nsubseteq C$ or $B_2 \subseteq C$. Then, $C \vDash B_1 \longrightarrow B_2$. *C respects a set* of $\mathcal{L}$ implications if it respects every implication in $\mathcal{L}$.

Respecting an implication can be generalised to the notion of *validity* w.r.t a formal context.

Definition 2.6. For a formal context $\mathbb{K} = (G, M, I)$, and an implication $B_1 \longrightarrow B_2$ over M, the implication is *valid* in the formal context ($\mathbb{K} \models B_1 \longrightarrow B_2$) iff for every object $g \in G$, it is the case that the object intent g' respects the implication. Then, the following are equivalent:

$$\mathbb{K} \models B_1 \longrightarrow B_2$$
$$B_1' \subseteq B_2'$$
$$B_2 \subseteq B_1''$$

If $\mathcal{L}$ is the set of all implications valid in a formal context, $\mathbb{K} = (G, M, I)$, then the concept intents can be given by $\{B \subseteq M \mid B \models \mathcal{L}\}$.

2.2 KLM-Style Non-monotonic Reasoning

In [14, 15], Shoham developed a semantic framework for introducing non-monotonicity, which he called *preferential reasoning*. The style of reasoning is based on the idea that an ordering can be imposed on the valuations of a knowledge base, expressing that some valuations are preferred to others. One might, for example, argue that a world in which pigs fly is worthy of less consideration than one where they do not.

In the classical setting, $\alpha \vDash \beta$ means that all models of α are also models of β. A corollary of this is that $\alpha \wedge \gamma \vDash \beta$, which implies monotonicity. Under preferential reasoning, $\alpha \vDash_{\sqsubset} \beta$ does not enforce the preferred models of $\alpha \wedge \gamma$ to be models of β, thus constituting a non-monotonic consequence [14].

Independently, Kraus, Lehmann, and Magidor introduced systems of non-monotonic reasoning as consequence relations, these have since been unified under the *KLM framework*. The central contribution of [9] is *system P*: a preferential consequence relation who's semantics are a variation of Shoham's preferential reasoning. A preferential consequence relation satisfies certain properties, called the *KLM postulates*.[1] Of particular interest is *cautious monotonicity*, which stipulates that new knowledge—which could already be inferred—cannot invalidate previous conclusions. Put plainly, only genuinely new information should cause retraction of existing information [8, 9].

Later, Lehmann and Magidor [11] introduced the stronger *rational consequence relation*, which satisfies all the properties of P and an additional property: *rational monotonicity*. Rational monotonicity says that new information which is consistent with existing knowledge should not lead to the retraction of prior inferences. *Ranked interpretations* provide a semantic characterisation of a rational consequence relation.

Definition 2.7. A *ranked interpretation* is a function $\mathsf{R} : \mathcal{U} \mapsto \mathcal{N} \cup \infty$ s.t. for every $n \in \mathcal{N}$ there exists a $u \in \mathcal{U}$ such that $\mathsf{R}(u) = n$. Then, there exists $u' \in \mathcal{U}$ s.t. $\mathsf{R}(u') = n'$ with $0 \leq n' < n$

Following the intuition of preferential reasoning, for two valuations, v, v', v is regarded as being representative of a more typical (i.e., preferable, less exceptional) state of affairs if $\mathsf{R}(v) < \mathsf{R}(v')$ [11]. A ranked interpretation, R, defines a rational consequence relation $\mid\!\sim_{\mathsf{R}}$. For $\alpha \mid\!\sim_{\mathsf{R}} \beta$, we define $[\![\alpha]\!]^{\mathsf{R}} \coloneqq \{u \in \mathcal{U} \mid u \Vdash \alpha\}$. Then, the conditional holds iff for all minimal valuations $v \in [\![\alpha]\!]^{\mathsf{R}}$, $v \Vdash \beta$.

What remains is to describe a process of non-monotonic entailment which is consistent with pattern of reasoning described by rational consequence relations.

3 MOTIVATION FOR RESEARCH AREA

In practice, FCA has been implemented as a framework for information retrieval [12], program analysis, ontology engineering. Until very recently [1, 3] there have been no attempts to lift the expressivity of FCA's attribute logic to a non-monotonic counterpart. To explicate why non-monotonicity would be desirable in the context of FCA, a toy example is provided.

When introducing younger children about the vertebrates it is often expressed that a defining feature of mammals is that they give birth to live young, and that reptiles lay eggs instead. Mammals and reptiles can be formalised as concepts $\mathtt{m}$ and $\mathtt{r}$:

$$\mathtt{m} \coloneqq \big(\{\mathtt{platypus,\ldots,horse}\},\ \{\mathtt{warm\text{-}blooded,\ldots,live\ young}\}\big)$$
$$\mathtt{r} \coloneqq \big(\{\mathtt{snake,\ldots,J\text{-}chameleon}\},\ \{\mathtt{cold\text{-}blooded,\ldots,lay\ eggs}\}\big)$$

There is, however, a problem with this construction. Namely, Jackson chameleons are reptiles that give birth to live young, and platypodes are mammals that lay eggs, and so they should not belong to these concepts. The classical FCA response to this issue might be that there should be two sub-concepts $\mathtt{m}_1$ and $\mathtt{m}_2$ of $\mathtt{m}$. Then, $\mathtt{m}_1$ may specify those mammals which give birth to live young, and $\mathtt{m}_2$ those that lay eggs.

This argument forces the abandon the kind of reasoning that appears so central to our intuition. We do not think of mammals as

[1]Further discussion of these postulates can be found in [9]

being divided between those that give birth to live young and those that do not; moreover, there are likely several properties which cause similar rifts in conception. A more natural way of reasoning about this matter would be to think of Jackson chameleons as *exceptional* mammals. *Exceptionality* obviously suggests a counter notion of *typicality*.

Another important concern is the use implications of a formal context to discover correspondence between attribute(s). As a reminder of Definition 2.6, classical FCA implications are only valid in a formal context when they are respected by every object. Consequently, a context modelling birds and their attributes might find that, although most objects have wings and fly, there are exceptions (e.g., penguins, ostriches) which prevent `wings` $\rightarrow$ `fly` from being valid; and so we remain unenlightened about the relationship between these attributes. Once again, the classical framework does not handle exceptions well.

A method to describe implications which only partially hold in a formal context is desirable. *Association rules* are an existing approach, and association rule mining in FCA has been discussed [6, 10]. Association rules, however, use *confidence* and *support* as a means of discovering relationships. These relationships then do not correspond to some describable pattern of reasoning—e.g., rational consequence relations—but are rather a flavour of 'majority rules'.

4 RESEARCH QUESTIONS & OBJECTIVES

We now develop a more thorough framing of the problems discussed in Section 3, detailing the current view on what the correct questions are, and how we undertake finding their solutions.

5 PROJECT PLAN

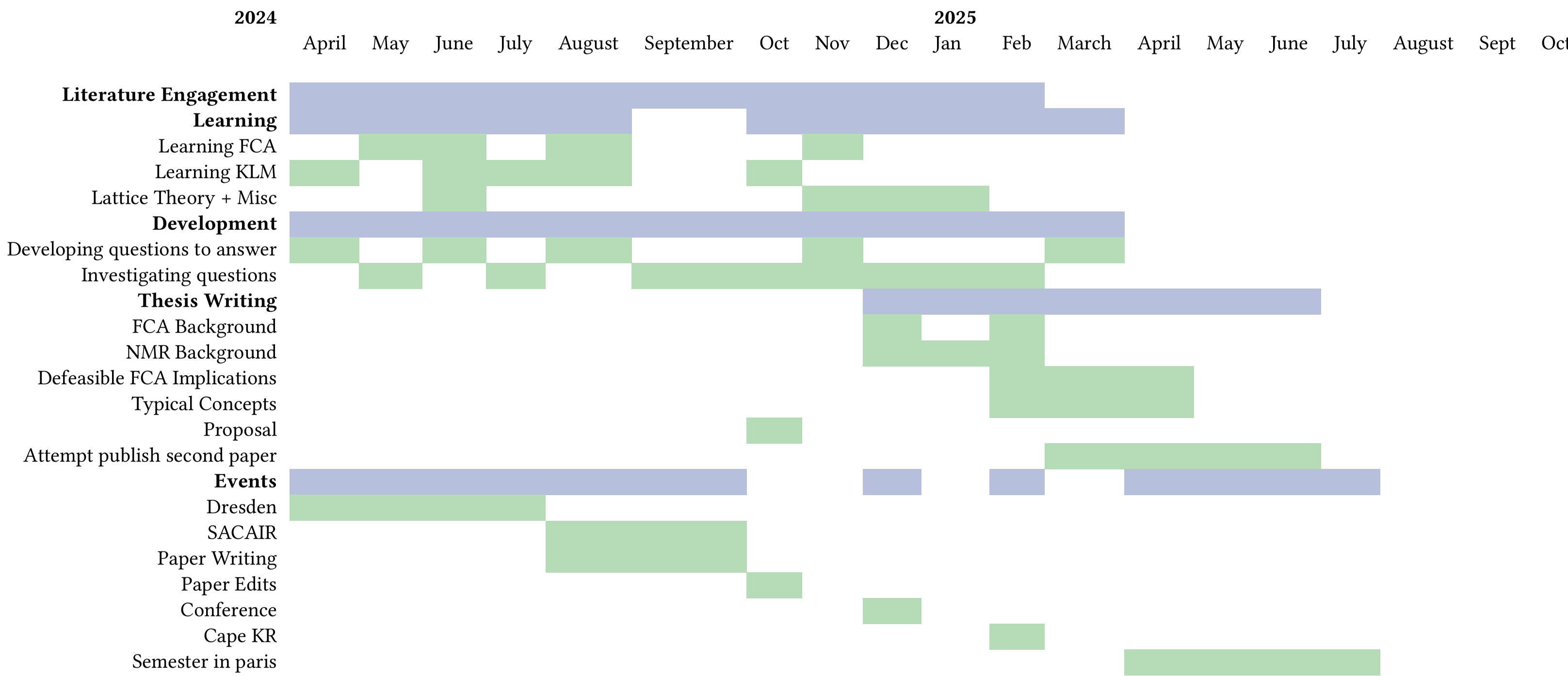
The period of October 2024 to January 2025 will be spent continuing the theoretical research with the aim of having results which collectively present a coherent, unified idea. More specifically, we aim to have looked into

(1) An algorithmic description of how to obtain a ranking over objects in a formal context
(2) An algorithmic definition for rational entailment in FCA
(3) A notion of *typical concepts* which includes a proof for lattice-like structure
(4) A notion of a *defeasible canonical basis*
(5) An algorithmic approach to determining typical concepts, analagous to *concept exploration*

5.1 Risks

5.2 Milestones, Deliverables, and Timeline

5.3 Gantt Chart



2024
2025
April
May
June
July
August
September
Oct
Nov
Dec
Jan
Feb
March
April
May
June
July
August
Sept
Oct
Literature Engagement
Learning
Learning FCA
Learning KLM
Lattice Theory + Misc
Development
Developing questions to answer
Investigating questions
Thesis Writing
FCA Background
NMR Background
Defeasible FCA Implications
Typical Concepts
Proposal
Attempt publish second paper
Events
Dresden
SACAIR
Paper Writing
Paper Edits
Conference
Cape KR
Semester in paris

REFERENCES

[1] Lucas Carr, Nicholas Leisegang, Thomas Meyer, and Sebastian Rudolph. 2024. Non-monotonic Extensions to Formal Concept Analysis via Object Preferences. (2024). arXiv:cs.LO/2410.04184 https://arxiv.org/abs/2410.04184

[2] Charles Castonguay. 2012. *Meaning and existence in mathematics*. Vol. 9. Springer Science & Business Media.

[3] Yiwen Ding, Krishna Manoorkar, Ni Wayan Switrayni, and Ruoding Wang. 2024. Defeasible Reasoning on Concepts. (2024). arXiv:cs.AI/2409.04887 https://arxiv.org/abs/2409.04887

[4] Kenneth G Ferguson. 2003. Monotonicity in practical reasoning. *Argumentation* 17 (2003), 335–346.

[5] B Ganter. 1999. Formal Concept Analysis: Mathematical Foundations. (1999).

[6] Bernhard Ganter, Sergei Obiedkov, Sebastian Rudolph, and Gerd Stumme. 2016. *Conceptual exploration*. Springer.

[7] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence* 226 (2015), 1–33.

[8] Adam Kaliski. 2020. An overview of KLM-style defeasible entailment. (2020).

[9] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44, 1-2 (1990), 167–207.

[10] Lotfi Lakhal and Gerd Stumme. 2005. Efficient mining of association rules based on formal concept analysis. In *Formal concept analysis: Foundations and Applications*. Springer, 180–195.

[11] Daniel Lehmann and Menachem Magidor. 1994. What does a conditional knowledge base entail? *Artificial Intelligence* 68, 2 (1994).

[12] Jonas Poelmans, Dmitry I Ignatov, Stijn Viaene, Guido Dedene, and Sergei O Kuznetsov. 2012. Text mining scientific papers: a survey on FCA-based information retrieval research. In *Advances in Data Mining. Applications and Theoretical Aspects: 12th Industrial Conference, ICDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 12*. Springer, 273–287.

[13] Sebastian Rudolph. 2007. *Relational exploration. Combining description logics and formal concept analysis for knowledge specification.*

[14] Yoav Shoham. 1987. Nonmonotonic Logics: Meaning and Utility.. In *IJCAI*, Vol. 10. Citeseer, 388–393.

[15] Yoav Shoham. 1987. *Reasoning about change: time and causation from the standpoint of artificial intelligence.* Yale University.

APPENDIX

	Aves	Mammal	Reptile	Aerial	Aquatic	Terrestrial	Migratory	Solitary	Carnivore	Eggs
Bat		×		×					×	
Crocodile			×		×	×			×	×
Crow	×			×						×
Hippo		×			×	×		×		
Ostrich	×					×		×		×
Penguin	×				×	×	×		×	×
Platypus		×			×	×		×	×	×
Snake			×			×		×	×	×
Swallow	×			×			×	×	×	×
Whale		×			×		×	×	×	

Table 1: A formal context describing animals (objects) and some of their features (attributes)

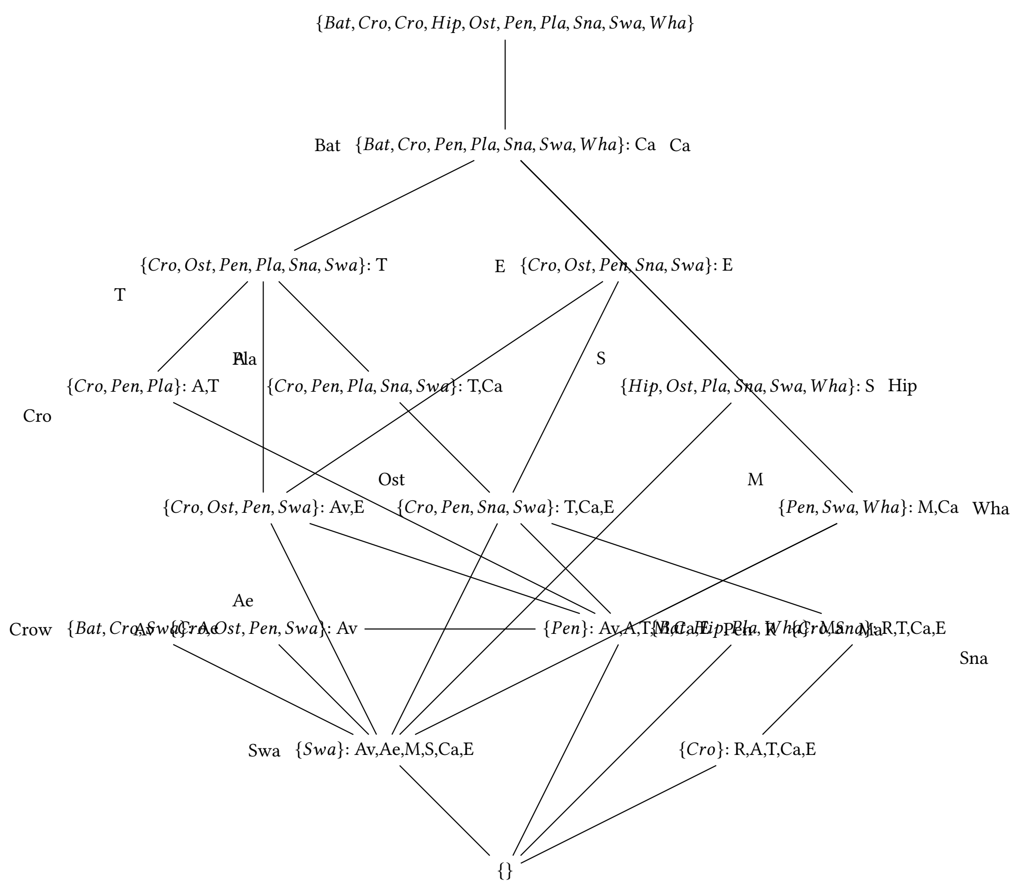


Figure 1: Concept Lattice for Animal Attributes

Algorithm 1 Computing the tolerance partition on objects G

Require: A formal context, $\mathbb{K} = (G, M, I)$;
Require: A set of defeasible implications, Δ;
Ensure: A tolerance partition $(R_0, \ldots, R_n)$ on G;
1: $P_0 \longleftarrow G$;
2: $i \longleftarrow 0$;
3: $\Delta \longleftarrow \text{Material}(\Delta)$;
4: **while** $P_{i-1} \neq P_i$ **do**
5: $\quad P_{i+1} \longleftarrow \{g \in P_i \mid \exists \delta \in \Delta g' \text{ s.t. } g' \not\models \delta\}$;
6: $\quad R_i \longleftarrow P_i \setminus P_{i+1}$;
7: $\quad \Delta_i \longleftarrow \{(\alpha \to \beta) \in \Delta \mid \exists g \in P_i \text{ s.t. } \alpha \subseteq g'\}$;
8: $\quad \Delta \longleftarrow \Delta \setminus \Delta_i$;
9: **end while**
10: **if** $P_{i-1} = \emptyset$ **then**
11: $\quad n \longleftarrow i - 1$;
12: **else**
13: $\quad n \longleftarrow i$;
14: **end if**
15: **return** $\big(R_0, \ldots, R_n\big)$;

To clarify some points on

- Line 3: this step turns Δ into a set of material implications;
- Line 5: this step puts into ranking $i + 1$ every object which **does not** respect Δ;
- Line 7: this sets Δ_i to be the set of formulae in Δ which have their antecedent satisfied by an object in P_i (implicitly all objects in P_i satisfy all formulae in Δ, but we only care about the ones which have their antecedent satisfied);
- We update Δ to be those formulae which are not yet "dealt with".

Algorithm 2 Checking entailment

Require: A formal context, $\mathbb{K} = (G, M, I)$;
Require: A set of defeasible implications, Δ;
Ensure: A tolerance partition $(R_0, \ldots, R_n)$ on G;
1: $P_0 \longleftarrow G$;
2: $i \longleftarrow 0$;
3: $\Delta \longleftarrow \text{Material}(\Delta)$;
4: **while** $P_{i-1} \neq P_i$ **do**
5: $\quad P_{i+1} \longleftarrow \{g \in P_i \mid \exists \delta \in \Delta g' \text{ s.t. } \nVdash \delta\}$;
6: $\quad P_i \longleftarrow P_i \setminus P_{i+1}$;
7: $\quad \Delta_i \longleftarrow \{(\alpha \to \beta) \in \Delta \mid \exists g \in P_i \text{ s.t. } \alpha \subseteq g'\}$;
8: $\quad \Delta \longleftarrow \Delta \setminus \Delta_i$;
9: **end while**
10: **if** $P_{i-1} = \emptyset$ **then**
11: $\quad n \longleftarrow i - 1$;
12: **else**
13: $\quad n \longleftarrow i$;
14: **end if**
15: **return** $\big(P_0, \ldots, P_n\big)$;