Concept Combination in Weighted Logic

Guendalina Righetti[1], Claudio Masolo[2], Nicolas Troquard[1], Oliver Kutz[1] and Daniele Porello[3]

[1]KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, piazza Domenicani 3, 39100, Bolzano, Italy

[2]ISTC-CNR Laboratory for Applied Ontology, Via alla cascata 56/C, 38123, Trento, Italy

[3]University of Genoa, Via Balbi 2-4-6, 16126, Genova, Italy

Abstract

We present an algorithm for concept combination inspired and informed by the research in cognitive and experimental psychology. Dealing with concept combination requires, from a symbolic AI perspective, to cope with competitive needs: the need for compositionality and the need to account for typicality effects. Building on our previous work on weighted logic, the proposed algorithm can be seen as a step towards the management of both these needs. More precisely, following a proposal of Hampton [1], it combines two weighted Description Logic formulas, each defining a concept, using the following general strategy. First it selects all the features needed for the combination, based on the logical distinction between necessary and impossible features. Second, it determines the threshold and assigns new weights to the features of the combined concept trying to preserve the relevance and the necessity of the features. We illustrate how the algorithm works exploiting some paradigmatic examples discussed in the cognitive literature.

1. Introduction

Dealing with concept combination requires, from an AI point of view that blends logical and cognitive perspectives, to cope with competitive needs: the need for *compositionality* and the need to account for *typicality effects*. Compositionality would require (the representation of) a combined concept to be a function of (the representations of) the combining concepts. It is often advocated as one of the main explanations for the human ability to create and understand new meaningful concepts [2]. Typicality effects refer to a number of phenomena—mainly observed in cognitive psychology [3]—related to the categorization task: some instances of a concept are more representative, and then easier to be categorized, than others.

Logically, concepts are often reduced to sets of necessary and sufficient conditions precluding the possibility to deal with typicality effects. Vice versa, cognitive theories of concepts focus on typicality effects by sacrificing compositionality: this is the case of, e.g. Prototype Theory. According to the Prototype Theory [4, 1], concepts are represented by means of *prototypes*, i.e.,

CAOS 2021: 5th Workshop on Cognition And OntologieS, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11-18, 2021, Bolzano, Italy

guendalina.righetti@stud-inf.unibz.it (G. Righetti); masolo@loa.istc.cnr.it (C. Masolo); Nicolas.Troquard@unibz.it (N. Troquard); Oliver.Kutz@unibz.it (O. Kutz); daniele.porello@unige.it (D. Porello)

sets of features associated with weights representing their relevance for the concept.[1] Typicality can be evaluated by summing up the weights of the features (in the prototype) matched by a given individual: the most typical members, the *best exemplars*, are the individuals with the highest score. However, the Prototype Theory seems inadequate to capture compositionality, as paradigmatically illustrated by the Pet Fish example. A *gold fish* is a very typical example of Pet Fish but it is a quite atypical example of Fish, and a quite poor example of Pet. This is known as *conjunction effect*: when an individual is well described by a concept combination, it is usually more typical of the combined concept than of the two components [5]. In a different perspective, the prototypical instances of Pet are furry, the ones of Fish are grey, but the prototypical instances of Pet Fish are neither furry nor (likely) grey. So, it is argued, the typicality of the combined concept is not predictable from the one of the component concepts [6]. Consequently, prototypes are unable to account for concept combination and for the productivity of human concepts, i.e., concepts cannot be represented by prototypes [2].

Some approaches, e.g., [7] and [8], tried to overcome this impasse by 'importing' prototypes into a formal setting. Here we pursue this general idea by deploying the logical framework we proposed in [9] which extends Description Logic languages by means of a family of operators (denoted by the symbol $\nabla\!\!\!\nabla$, spoken 'tooth') allowing one to define concepts in terms of weighted features. Similarly to prototypes, each operator takes a list of concept descriptions and assigns a weight to each of them. Furthermore, once a threshold is determined, the operator returns a complex concept which applies to those instances that satisfy certain combinations of features, the ones that reach the chosen threshold by summing up the weight of the satisfied features.

Given the $\nabla\!\!\!\nabla$-definitions of two concepts (and a given knowledge base), we introduce an algorithm that returns the $\nabla\!\!\!\nabla$-definition of the combined concept. The general rules governing the algorithm are grounded on the ones analyzed in cognitive science studies. In particular, we refer to the work done by Hampton in [1]. We then illustrate how the algorithm works by analysing the Pet Fish example and several paradigmatic examples discussed in the literature.

2. Hampton's Model of Attribute Inheritance

Different models have been proposed in the context of prototype theory in order to deal with the kind of *conjunction effects* described above. A quite famous one is the *selective modification model* proposed by [10]. The authors propose an elaboration of the Prototype Theory, interpreting concepts in terms of schema structures. Distinguishing between dimensions (e.g., color) and features (e.g., red) of a concept, the proposed model is able to account for the conjunction effect in the case of adjective-noun combinations (e.g. red apple). Unfortunately, it has little to say in the context of noun-noun combinations.

A different kind of analysis is proposed in [1]. Still within the Prototype Theory, Hampton proposes an *attribute inheritance model*[2], which analyses the case of *conjunctive* noun-noun combinations (e.g., a Sport which is a Game). According to this proposal, the features of the combined concept are initially collected from the ones of the constituent concepts following the

[1]The Prototype Theory usually distinguishes *attributes* or *dimensions* (e.g., color) from *values* or *features* (e.g., crimson). In the following we do no consider this distinction and we focus on features (values).

[2]Following Hampton, in this section the terms 'attribute' and 'feature' are used interchangeably.

standard rule for conjunction. The weight of the features of the combined concept depends on the weight(s) this feature has in the prototypes of the combining concepts. In particular, when the feature appears in the prototypes of both the combining concepts, Hampton considers the average value, a sort of trade-off.

An original aspect of the proposal concerns the constraints posed on the inheritance of the features. Hampton introduces two main constraints: (i) the features that are necessary for either constituent are also necessary for the combination; (ii) the features that are impossible for either constituent are impossible also for the combination. The notions of necessity and impossibility are characterised in a logical way: an attribute is necessary when it holds for *all* instances of the concept while it is impossible when it is *necessarily false* for all the instances of the concept. Hampton analyzes the example of the Pet Fish: a Pet is necessarily owned, and for a Fish it is impossible to be cuddly. Then Pet Fish must inherit the first attribute but cannot inherit the second one. He also suggests that the idea of averaging the weights of the features shared by the combining concepts may not work in the case of impossible and necessary features. In case of necessity (impossibility) a maximum (minimum) rule is applied, i.e., the weight of the necessary (impossible) features of the combined concept is inherited from the combining concept where this feature is more (less) relevant.

The model proposed by Hampton is able to explain several phenomena concerning concept-combination as observed in the context of experimental psychology. Two of them are of particular interest in the analysis of Pet Fish, namely, *inheritance failure* and *attribute emergence*.

Inheritance failure occurs when a feature which is important for a constituent (i) becomes irrelevant for the conjunction or (ii) it is not inherited at all. The case (i) can be explained as an effect of the averaging procedure: when the constituents share a feature, its weight can be high in the prototype of one constituent but very low in the prototype of the other constituent. The case (ii) can be explained by exploiting the notion of impossibility discussed above: if a feature is impossible for one of the constituent concepts, it is not inherited at all.

Attribute emergence is the inverse of inheritance failure, namely the increased weight of a feature in the prototype of the conjunction w.r.t. its weights in the prototypes of the constituent concepts, or the emergence of a new feature of the conjunction. Hampton explains this phenomenon in terms of *extensional feedback*: there is a feedback from past experiences and background knowledge into the combined concept. This means that the prototype of the combined concept can be adapted taking into account the experienced exemplars and the available knowledge about the environment. In the case of Pet Fish, one can for instance introduce some necessary features like 'small' and 'lives in aquarium'.

3. Tooth Operators and Preliminary Hypotheses

In this section, we delineate the formal framework necessary to introduce the $\nabla\!\!\!\nabla$-definitions of concepts serving as input for the algorithm for concept combination presented in Sect. 4. Following the work done in [9, 11, 12], we extend standard DL languages [13] with a class of m-ary operators denoted by the symbol $\nabla\!\!\!\nabla$ (spoken 'tooth'). Each operator works as follows: (i) it takes a list of concepts, (ii) it associates a weight (i.e., a number) to each of them, and (iii) it returns a complex concept that applies to those instances that satisfy a certain combination of concepts, i.e., those instances for which, by summing up the weights of the satisfied concepts, a certain threshold is met. More precisely, we assume a vector of m weights $\vec{w} \in \mathbb{R}^m$ and a threshold value $t \in \mathbb{R}$. If $C_1, \ldots, C_m$ are concepts of $\mathcal{ALC}$, then $\nabla\!\!\!\!\nabla^t_{\vec{w}}(C_1, \ldots, C_m)$ is a concept of $\mathcal{ALC}_{\nabla\!\!\!\nabla}$. For $C'_i \in \mathcal{ALC}$, the set of $\mathcal{ALC}_{\nabla\!\!\!\nabla}$ concepts is described by the grammar:

$$C \quad ::= \quad A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C \mid \nabla\!\!\!\!\nabla^t_{\vec{w}}(C'_1, \ldots, C'_m)$$

To better visualise the weights an operator associates to the concepts, we often use the notation $\nabla\!\!\!\!\nabla^t((C_1, w_1), \ldots, (C_m, w_m))$ instead of $\nabla\!\!\!\!\nabla^t_{\vec{w}}(C_1, \ldots, C_m)$. A knowledge base KB is a finite set of concept inclusions of the form $C \sqsubseteq D$, where C and D are concept expressions. We write $C \equiv D$ to signify that $C \sqsubseteq D$ and $D \sqsubseteq C$.

Given finite, disjoint sets N_C and N_R of concept and role names, respectively, an interpretation I consists of a non-empty set Δ^I and a mapping $\cdot^I$ that maps every concept name C to a subset $C^I \subseteq \Delta^I$ and every role name $R \in N_R$ to a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. The semantics of the operator is obtained by extending the definition of the semantics of $\mathcal{ALC}$ as follows. Let $I = (\Delta^I, \cdot^I)$ be an interpretation of $\mathcal{ALC}$. The interpretation of a $\nabla\!\!\!\!\nabla$-concept $C = \nabla\!\!\!\!\nabla^t((C_1, w_1), \ldots, (C_m, w_m))$ is:

$$C^I = \{d \in \Delta^I \mid v^I_C(d) \geq t\} \tag{1}$$

where $v^I_C(d)$ is the *value* of $d \in \Delta^I$ under the concept C, defined as:

$$v^I_C(d) = \sum_{i \in \{1,\ldots,m\}} \{w_i \mid d \in C^I_i\} \tag{2}$$

For instance consider $C_E = \nabla\!\!\!\!\nabla^{1.8}((F_1, 1.2), (F_2, 1), (F_3, 0.4), (F_4, 0.1))$. If an individual $d \in F^I_1 \cap F^I_3$ but $d \notin F^I_2$, then $d \notin {C_E}^I$ because even when $d \in {F_4}^I$ we have that $v^I_{C_E}(d) = 1.2 + 0 + 0.4 + 0.1 = 1.7 < 1.8$.

The interpretation I is a *model* of the knowledge base KB if for every concept inclusion $C \sqsubseteq D$ in KB, it is the case that $C^I \subseteq D^I$. A concept inclusion $C \sqsubseteq D$ is *entailed* by the knowledge base KB (noted KB $\models C \sqsubseteq D$) when $C^I \subseteq D^I$ holds for every model I of KB. A concept C is *satisfiable* in KB when $C^I \neq \varnothing$ for some model I of KB. Adding tooth-expressions to the language of $\mathcal{ALC}$ is thus done without modifying the standard notion of interpretation in DL. As observed in [9], tooth-operators do not increase the expressive power of any language that contains the standard Boolean operators. It was shown in [14] that adding tooth-operators to such DL languages does not increase the complexity of the corresponding inference problem. Tooth-operators behave like *perceptrons* [15, 14]: a (non-nested) tooth expression is a linear classification model, which enables to *learn* weights and thresholds from real data (in particular, from set of assertions about individuals) exploiting standard linear classification algorithms. Multilayer perceptrons can then be represented via nested tooth expressions.

The design of the tooth operator is inspired by the *Prototype Theory*: the concepts in the $\nabla\!\!\!\!\nabla$-definition of C may be seen as the *features* of C and their weights may be intended to represent the relevance of such features (for C). This allows us to express typicality effects in the context of a logical representation: in our setting, the most typical instances, the *best exemplars*, of C are the individuals with the highest score, i.e., by exploiting the value v^I_C, individuals can be ordered in terms of typicality.

Given $C = \nabla\!\!\!\!\nabla^t((F_1, w_1), ..., (F_m, w_m))$, a knowledge base KB, and a set $\mathcal{P}$ of concepts, we introduce the following sets:

- $\mathbf{ft}(C) = \{F_1, \ldots, F_m\}$;
- $\mathbf{snc}(C) = \{F_i \in \mathbf{ft}(C) \mid \sum_{j \neq i} w_j < t\}$;
- $\mathbf{nc}(\mathrm{KB}, C, \mathcal{P}) = \{P \in \mathcal{P} \mid \mathrm{KB} \vDash C \sqsubseteq P\}$;
- $\mathbf{im}(\mathrm{KB}, C, \mathcal{P}) = \{P \in \mathcal{P} \mid \mathrm{KB} \vDash C \sqsubseteq \neg P\}$.

$\mathbf{ft}(C)$ is the set of the *features* of C while $\mathbf{snc}(C)$ is the set of the *strongly necessary features* of C: individuals lacking a feature in $\mathbf{snc}(C)$ cannot reach the threshold. Note that $\mathbf{snc}(C)$ is defined in a purely syntactic way, logical inference is not deployed here. By relying on $\mathbf{nc}$ and $\mathbf{im}$ (that are grounded on logical inference), the sets of *necessary* and *impossible* features of C w.r.t. KB can be defined as $\mathbf{nc}(\mathrm{KB}, C, \mathbf{ft}(C))$ and $\mathbf{im}(\mathrm{KB}, C, \mathbf{ft}(C))$, respectively. Note that $\mathbf{snc}(C) \subseteq \mathbf{nc}(\varnothing, C, \mathbf{ft}(C))$; indeed when $\vDash F_i \sqsubseteq F_j$ (with $i \neq j$) we have that $\mathbf{snc}(C) \subset \mathbf{nc}(\varnothing, C, \mathbf{ft}(C))$. In the previous example, $\mathbf{snc}(C_E) = \{F_1, F_2\}$ and, assuming that KB contains only $F_2 \sqsubseteq F_3$, $\mathbf{nc}(KB, C_E, \mathbf{ft}(C_E)) = \{F_1, F_2, F_3\}$.

In the following, we always consider a single knowledge base KB. To simplify the notation we then write $\mathbf{nc}(C, \mathcal{P})$ and $\mathbf{im}(C, \mathcal{P})$ rather than $\mathbf{nc}(\mathrm{KB}, C, \mathcal{P})$ and $\mathbf{im}(\mathrm{KB}, C, \mathcal{P})$, respectively. Furthermore, we assume that all the $\nabla\!\!\!\!\nabla$-concepts C are satisfiable in KB and that they are not redundant, i.e., $\mathbf{im}(C, \mathbf{ft}(C)) = \varnothing$ (none of the features included in the tooth contradicts any of the necessary features of C). Finally, given a set $\mathcal{P}$ of concepts, we write $\lceil \mathcal{P} \rceil$ to indicate the conjunction of all the concepts in $\mathcal{P}$.

4. An Algorithm to Combine $\nabla\!\!\!\!\nabla$-Concepts

We present an algorithm for concept combination inspired by the work of Hampton [1] discussed in section 2. Following [1], we mainly focus on the case of *conjunctive* concept combination, i.e., combinations that closely relate to a conjunction of the constituent concepts. The algorithm considers as input the $\nabla\!\!\!\!\nabla$-definitions of two concepts, one (H) playing the role of *head* and one (M) playing the role of *modifier*, and it outputs the $\nabla\!\!\!\!\nabla$-definition of the combined concept noted $M{\circ}H$. Without losing generality [9, 15], we assume that: (i) H and M have the same positive threshold and (ii) all the features have positive weights.

The head and modifier roles are based on a linguistic distinction on noun-noun compound [16]. Looking at noun-noun combinations in English, two parts can be distinguished, the Head and the Modifier, depending on the syntactic construction of the compound. Considering for instance a "Tool Weapon", the noun "weapon" would play the role of the Head, whereas "tool" would be the Modifier. As the names suggest, the Head provides the base category of the combined concept, whilst the Modifier alters the attributes of the Head. The result is that a "Tool Weapon" may be in principle quite different from a "Weapon Tool". However, the role of the Head concept may here be also compliant with the notion of a *dominant concept* (discussed below), as introduced in [1], in line with the work in [8].

The algorithm consists of three phases: phase 1 collects the features of $M{\circ}H$ by assuming that the head dominates the modifier; phase 2 selects the weights of the features of $M{\circ}H$; and phase 3 determines a range of possible thresholds for (the $\nabla\!\!\!\!\nabla$-definition of) $M{\circ}H$. As we will see, in phase 1 the logical nature of $\overline{\mathbb{W}}$-operators allows one to use the inference power of logic to determine incompatibilities between the features of H and M. Vice versa, phases 2 and 3 use only the information made available by the (intensional) $\overline{\mathbb{W}}$-definitions of concepts.

Phase 1: features. The set of the features of $M{\circ}H$ is built in two steps:

$$\begin{array}{ll}\text{(Step 1)} & \bar{\mathbf{ft}}(M{\circ}H) = \mathbf{nc}(H, \mathbf{ft}(H)) \cup (\mathbf{nc}(M, \mathbf{ft}(M)) \setminus \mathbf{im}(H, \mathbf{ft}(M))) \\ \text{(Step 2)} & \mathbf{ft}(M{\circ}H) = \mathbf{ft}(H) \setminus \mathbf{im}(H \sqcap \lceil \bar{\mathbf{ft}}(M{\circ}H) \rceil, \mathbf{ft}(H)) \cup \\ & \qquad\qquad\quad \mathbf{ft}(M) \setminus \mathbf{im}(M \sqcap \lceil \bar{\mathbf{ft}}(M{\circ}H) \rceil, \mathbf{ft}(M))\end{array}$$

Step 1 collects all the necessary features of H together with all the necessary features of M which are not impossible for H. This shows in which sense H dominates M, H is the base of $M{\circ}H$: in case of incompatibilities we discard necessary features of M, not of H. It follows that $\mathbf{ft}(M{\circ}H)$ and $\mathbf{ft}(H{\circ}M)$ can differ.

Step 2 builds on the previous step, examining all the non-necessary features of both H and M. Specifically, it aims at excluding all the features of H (resp. M), which are impossible for H (resp. M) itself, once all the necessary features of M (resp. H) in $\bar{\mathbf{ft}}(M{\circ}H)$ are added.

The selection of features based on the distinction between necessary and impossible features aims at mimicking the model proposed in [1] and discussed in section 2.

Phase 2: weights. Once built the set $\mathbf{ft}(M{\circ}H)$ containing all the features of $M{\circ}H$, weights are assigned to them in the following way:

(1) the weight of each feature in $\mathbf{ft}(H)$ (resp. $\mathbf{ft}(M)$) is divided by the maximal sum of the weights of consistent (in KB) subsets of $\mathbf{ft}(H)$ (resp. $\mathbf{ft}(M)$);
(2) for all the features in $\mathbf{s}\bar{\mathbf{ft}}(M{\circ}H) = \bar{\mathbf{ft}}(M{\circ}H) \cap (\mathbf{snc}(M) \cup \mathbf{snc}(H))$ we consider the weight calculated in (1) except for the ones in $\mathbf{s}\bar{\mathbf{ft}}(M{\circ}H) \cap \mathbf{ft}(H) \cap \mathbf{ft}(M)$ for which we consider the maximal weight (of the weights calculated in (1));
(3) for all the features in $\mathbf{ft}(M{\circ}H) \setminus \mathbf{s}\bar{\mathbf{ft}}(M{\circ}H)$ we consider the weight calculated in (1) except for the ones in $(\mathbf{ft}(M{\circ}H) \setminus \mathbf{s}\bar{\mathbf{ft}}(M{\circ}H)) \cap \mathbf{ft}(H) \cap \mathbf{ft}(M)$ for which we consider the average weight (of the weights calculated in (1)).

In Sect. 3 we observed that the weight of a feature can be seen as an indicator of the relevance of such feature for the concept. The idea in (1) is to normalize this indicator with respect to the value v_C^I of the possible best exemplars. The numbers of features of H and M may substantially differ preventing absolute weights to be accurate relevance-indicators. (2) and (3) attribute to the features of $M{\circ}H$ the weights calculated in (1) except when a feature belongs to $\mathbf{ft}(H) \cap \mathbf{ft}(M)$ and has different normalized weights (in H and M). In these cases, following Hampton [1], we consider the maximal weight for the features in $\mathbf{s}\bar{\mathbf{ft}}(M{\circ}H)$—as we will see in the discussion of the phase 3, $\mathbf{s}\bar{\mathbf{ft}}(M{\circ}H)$ corresponds to $\mathbf{snc}(M{\circ}H)$—and the average weight for the other features of $M{\circ}H$.

Phase 3: threshold. We fix the threshold for $M{\circ}H$ to assure that $\mathbf{s}\bar{\mathbf{ft}}(M{\circ}H) = \mathbf{snc}(M{\circ}H)$, i.e., the strongly necessary features of H together with the strongly necessary features of M (that are compatible with the necessary features of H) are also strongly necessary features of

$M \circ H$. To do that, the threshold must belong to the open interval $(w - w^{m+}, w - w^{M-})$ where w is the sum of the weights of the features in $\mathbf{ft}(M \circ H)$, w^{m+} is the minimal weight of the features in $\bar{\mathbf{sft}}(M \circ H)$, and w^{M-} is the maximal weight of the features in $\mathbf{ft}(M \circ H) \setminus \bar{\mathbf{sft}}(M \circ H)$. By increasing the threshold we exclude some combinations of non-necessary features. Furthermore, assume that $\lceil \bar{\mathbf{sft}}(M \circ H) \rceil$ implies some features in $\mathbf{ft}(M \circ H) \setminus \bar{\mathbf{sft}}(M \circ H)$. These implied features are necessary even though the threshold can be reached without counting their weights, i.e., they are not strongly necessary. It is also possible that some features in $\bar{\mathbf{ft}}(M \circ H) \setminus \bar{\mathbf{sft}}(M \circ H)$ are not necessary for $M \circ H$, i.e., the algorithm preserves the strong necessity but not the necessity. E.g., consider the case where $\mathbf{ft}(M) = \{F_1, F_2\}$, $F_1 \in \mathbf{snc}(M)$, $F_2 \notin \mathbf{snc}(M)$, $\mathrm{KB} \models F_1 \sqsubseteq F_2$, and F_1 (but not F_2) is incompatible with the necessary features of H. In this case $F_2 \in \bar{\mathbf{ft}}(M \circ H) \setminus \bar{\mathbf{sft}}(M \circ H)$ but F_2 is no more necessary for $M \circ H$ because F_1, which grounds the necessity of F_2, has been discarded.

5. Examples

We illustrate how the proposed algorithm works by means of several paradigmatic examples of noun-noun and adjective-noun combinations. Without presenting a direct empirical validation, we analyze how the algorithm accounts for the phenomena and rules identified by Hampton in [1] to build the prototypes of conjunctive combinations of concepts (in particular, inheritance failure, dominance effect, overextension, and emergence of features). Even though the $\nabla\!\!\!\!\nabla$-definitions and KBs we consider seem plausible, we cannot commit on their empirical foundation. Their main intent is to show how the effectiveness of the rules proposed by Hampton critically depends not only on the weights of the features and on the thresholds in the $\nabla\!\!\!\!\nabla$-definitions but also on the assumed background knowledge. The embedding of prototypes into a logical framework allows the algorithm to explicitly and formally take into account both these aspects.

5.1. Noun-Noun Combinations

5.1.1. Pet Fish

We start by considering the case of Pet Fish that has been advocated to show the inadequacy of the Prototype Theory to capture concept combinations. Consider the following $\nabla\!\!\!\!\nabla$-definitions and assume $Fish$ is the head and Pet the modifier:

$$Fish = \nabla\!\!\!\!\nabla^{10}((\forall livesIn.Water, 3), (\neg WarmBlooded, 3), (\forall breathesThrough.Gill, 3), (Grey, 0.9), (Scaly, 0.9), (\exists hasPart.Fin, 1))$$

$$Pet = \nabla\!\!\!\!\nabla^{10}((CaredFor, 3)(Pretty, 3), (\forall livesIn.House, 3), (Furry, 0.9), (WarmBlooded, 0.9), (\exists providesCompanionshipTo.Human, 1))$$

Furthermore, assume the following KB:

$$\forall livesIn.Water \sqcap \forall livesIn.House \sqsubseteq \forall livesIn.Aquarium \tag{3a}$$

$$Furry \sqsubseteq WarmBlooded \tag{3b}$$

$$Pretty \sqcap \forall breathesThrough.Gill \sqsubseteq \neg Grey \tag{3c}$$

Phase 1 collects the features of $Pet{\circ}Fish$. More precisely, (Step 1) defines the set of *necessary* features of $Pet{\circ}Fish$. It collects the necessary features of the Head concept $Fish$ ($\mathbf{nc}(Fish, \mathbf{ft}(Fish)) = \{\forall livesIn.Water, \neg WarmBlooded, \forall breathesThrough.Gill\}$) and the necessary features of the Modifier Pet which are not impossible for $Fish$. In this case there are no inconsistencies between the necessary features of the two concepts, therefore

$$\bar{\mathbf{ft}}(Pet{\circ}Fish) = \{\forall livesIn.Water, \neg WarmBlooded, \forall breathesThrough.Gill, CaredFor, \\ Pretty, \forall livesIn.House\}$$

(Step 2) examines the *non-necessary* features aiming at excluding all the features of $Fish$ (resp. Pet), which are impossible for the concept $Fish$ (resp. Pet) itself, once one adds all the necessary features of the concept Pet (resp. $Fish$) in $\bar{\mathbf{ft}}(Pet{\circ}Fish)$. In our example,

$$\mathbf{ft}(Pet{\circ}Fish) = \{\forall livesIn.Water, \neg WarmBlooded, \forall breathesThrough.Gill, CaredFor, \\ Pretty, \forall livesIn.House, Scaly, \exists hasPart.Fin, \\ \exists providesCompanionshipTo.Human\}$$

Indeed:

- $Furry \notin \mathbf{ft}(Pet{\circ}Fish)$ because of (3b) in the KB and $\neg WarmBlooded \in \mathbf{nc}(Fish, \mathbf{ft}(Fish))$ and then $\neg WarmBlooded \in \bar{\mathbf{ft}}(Pet{\circ}Fish)$;
- $WarmBlooded \notin \mathbf{ft}(Pet{\circ}Fish)$ again because $\neg WarmBlooded \in \mathbf{nc}(Fish, \mathbf{ft}(Fish))$;
- $Grey \notin \mathbf{ft}(Pet{\circ}Fish)$ because of (3c) in the KB and $Pretty, \forall breathesThrough.Gill \in \bar{\mathbf{ft}}(Pet{\circ}Fish)$.

Phase 2 assigns weights to the features in $\mathbf{ft}(Pet{\circ}Fish)$. First, note that $\bar{\mathbf{ft}}(Pet{\circ}Fish) = \mathbf{s}\bar{\mathbf{ft}}(Pet{\circ}Fish)$. Second, $\mathbf{ft}(Fish) \cap \mathbf{ft}(Pet) = \varnothing$, therefore there are no cases in (2) and (3) in phase 2 where we need to maximize or average the weights. For each feature of each component concept, the weight associated to that feature is divided by the score of the best exemplars of that concept (11.8 for $Fish$; $Fish$ has no incompatible features therefore the best exemplars have all the features in $\mathbf{ft}(Fish)$, similarly for Pet). We obtain:

$$Pet{\circ}Fish = \mathbb{W}^t((\forall livesIn.Water, 0.25), (\neg WarmBlooded, 0.25), \\ (\forall breathesThrough.Gill, 0.25), (CaredFor, 0.25), (Pretty, 0.25), \\ (\forall livesIn.House, 0.25), (Scaly, 0.07), (\exists hasPart.Fin, 0.08), \\ (\exists providesCompanionshipTo.Human, 0.08))$$

Following the phase 3, the threshold $t \in (1.48, 1.65)$. The lower bound is the sum of the weights of the features in $\mathbf{ft}(Pet{\circ}Fish)$ minus the smaller weight of the features in $\mathbf{s}\bar{\mathbf{ft}}(Pet{\circ}Fish)$, i.e., $1.73 - 0.25 = 1.48$. The upper bound is the sum of the weights of the features in $\mathbf{ft}(Pet{\circ}Fish)$ minus the bigger weight of the features in $\mathbf{ft}(Pet{\circ}Fish) \setminus \mathbf{s}\bar{\mathbf{ft}}(Pet{\circ}Fish)$, i.e., $1.73 - 0.08 = 1.65$. When $t \in (1.48, 1.65)$, we have $\mathbf{snc}(Pet{\circ}Fish) = \mathbf{s}\bar{\mathbf{ft}}(Pet{\circ}Fish)$ and none of the non-necessary features of $Fish$ and $Head$ becomes strongly necessary for $Pet{\circ}Fish$.

A note on the inheritance failure. One of the main points of the Pet Fish counter-example against the prototype theory is the *inheritance failure* of, e.g., the feature $Grey$: prototype representations are not compositional, it is argued, since there is no rule able to explain why, e.g., fishes, but not petfishes, are usually grey. In our example $Grey \in \mathbf{ft}(Fish)$ but $Grey \notin \mathbf{ft}(Pet \circ Fish)$. This is due to the fact that (i) $Grey$ is a non-necessary feature of $Fish$; (ii) $Pretty$ is necessary feature of Pet compatible with all the necessary features of $Fish$; and (iii) KB contains (3c). Even when dropping one of these assumptions, alternative strategies may be exploited to at least partially model inheritance failure in our framework,.

Assume, for instance, that $Pretty$ is a necessary feature of Pet, but KB does not contain (3c). In this case we would have that $Pretty, Grey \in \mathbf{ft}(Pet \circ Fish)$. The relevance of $Grey$ w.r.t. $Pet \circ Fish$ is lower than the one of $Grey$ w.r.t. $Fish$ but this holds in general when the number of features in $Pet \circ Fish$ is higher than the number of features in $Fish$. We may however find some mechanisms to enforce $Grey$ to have an increasingly marginal relevance w.r.t. $Pet \circ Fish$. Inspired by the idea of *extensional feedback* [17], we could decrease (or increase) the weight of a feature according to the number of instances, within KB, satisfying that feature in the context of the new concept. For instance, we may observe that, extensionally, when adding the feature $Pretty$ to the concept $Fish$, the number of $Grey$ fish (proportionally) decrease, and reduce the weight of the feature $Grey$ accordingly. This may be thought as an additional step within our algorithm that nicely integrates the prototype, knowledge, and exemplar views on concepts.

Conversely, if $Pretty$ is a non-necessary feature of Pet and KB contains (3c), still we have that $Pretty, Grey \in \mathbf{ft}(Pet \circ Fish)$. This would not cause any problem in terms of consistency because both $Pretty$ and $Grey$ are non-necessary features of $Pet \circ Fish$. However, one could modify the algorithm to discard some non-necessary features inconsistent with other non-necessary features in order to produce a $\nabla\!\!\!\!\nabla$-definition for the combined concept that maximizes the typicality of the best exemplars.

A note on emergent features. Assuming KB contains (3a), we have that $\mathrm{KB} \models Pet \circ Fish \sqsubseteq \forall livesIn.Aquarium$ but neither $\mathrm{KB} \models Fish \sqsubseteq \forall livesIn.Aquarium$ nor $\mathrm{KB} \models Pet \sqsubseteq \forall livesIn.Aquarium$ hold. $\forall livesIn.Aquarium$ can then be seen as an *emergent* property of $Pet \circ Fish$ which follows only by the conjunction of the necessary features of $Fish$ and Pet.

5.1.2. Sport Game

The Sport Game concept is among the combinations considered by Hampton in his experiments and it is an example of combinations becoming quite close to conjunctions. Assume that KB is empty and that $Sport$ and $Game$ are defined as follows:[3]

$$\begin{aligned}
Sport = \nabla\!\!\!\!\nabla^4((&PhysicalActivity, 3), (\exists doneFor.Fun, 0.6), (IsCompetitive, 0.7),\\
&(\exists doneFor.Health, 0.7), (\exists needs.AthleticAbilities, 0.8))\\
Game = \nabla\!\!\!\!\nabla^4((&\exists doneFor.Fun, 2), (IsExciting, 1), (IsCompetitive, 1),\\
&(\neg(\exists has.SeriousGoal), 1), (\exists requires.Concentration, 1))
\end{aligned}$$

In this case, $Game$ has no necessary features[4], i.e., $\mathbf{nc}(Game, \mathbf{ft}(Game)) = \varnothing$. Moreover, the two concepts do not include clashing information, namely $\mathbf{im}(Game, \mathbf{ft}(Sport)) = \varnothing$ and $\mathbf{im}(Sport, \mathbf{ft}(Game)) = \varnothing$. We would then obtain $\overline{\mathbf{ft}}(Sport \circ Game) = \{PhysicalActivity\}$.

As a result, at the end of phase 1, all the features of $Sport$ and $Game$ are collected, i.e.,

[3] A similar example was used in [11], and imitates the features collected in [1].

[4] Following the well-known argument proposed in [18].

$$\mathbf{ft}(Sport{\circ}Game) = \{PhysicalActivity, \exists doneFor.Fun, IsCompetitive, \\ \exists doneFor.Health, \exists needs.AthleticAbilities, IsExciting, \\ \neg(\exists has.SeriousGoal), \exists requires.Concentration\}$$

The weight assignment proceeds as usual but note that $IsCompetitive$ and $\exists doneFor.Fun$ are two non-necessary features of $Sport{\circ}Game$ which belong to $\mathbf{ft}(Sport) \cap \mathbf{ft}(Game)$. According to (3) of phase 2, for these features we need then to take the average of the weights they have in $Sport$ and $Game$. At the end of phase 2, we obtain

$$Sport{\circ}Game = \nabla\!\!\!\!\nabla^t((PhysicalActivity, 0.5), (\exists doneFor.Fun, 0.2), (IsCompetitive, 0.14), \\ (\exists doneFor.Health, 0.12), (\exists needs.AthleticAbilities, 0.13), \\ (IsExciting, 0.16), (\neg\exists has.SeriousGoal, 0.16), \\ (\exists requires.Concentration, 0.16))$$

Phase 3 establishes that $t \in (1.07, 1.37)$.

A note on dominance effect. In the example, $\mathbf{ft}(Sport{\circ}Game) = \mathbf{ft}(Game{\circ}Sport)$. Actually, the algorithm outputs the same $\nabla\!\!\!\!\nabla$-definition for $Sport{\circ}Game$ and $Game{\circ}Sport$. If

$$PhysicalActivity \sqsubseteq \exists has.SeriousGoal \tag{4}$$

is included in KB, the situation does not change, the only difference is that (Step 2) of phase 1 rules out $\neg\exists has.SeriousGoal$ from $\mathbf{ft}(Sport{\circ}Game) = \mathbf{ft}(Game{\circ}Sport)$.[5]

In these cases, the role played by $Sport$ and $Game$ is not relevant and the syntactic distinction between Head and Modifier given by the word order has no impact in the construction of the combined concept. However, in his experiments Hampton observed another phenomenon called *dominance effect*: when one of the component concepts has a greater number of important features, the resulting combination is more similar to that concept, no matter what the words order is in carrying out the combination (see [1], p.57). First, $Sport$ has an essential feature while $\mathbf{nc}(\text{Game}, \mathbf{ft}(Game)) = \varnothing$. Second, the best exemplars d of $Sport{\circ}Game$ (and $Game{\circ}Sport$)[6] would obtain an higher score w.r.t. $Sport$ than w.r.t. $Game$, i.e., $v^I_{Sport}(d^I) > v^I_{Game}(d^I)$. These two remarks suggest (i) that $Sport$ has more important features than $Game$; and (ii) that our algorithm is more sensible to the dominance effect than to the Head/Modifier distinction.

A note on overextension. As briefly discussed above, the choice of the threshold plays a central role in determining the extension of the combination. The flexibility of the threshold permits to deal with another phenomenon observed by Hampton [1], namely *overextension*. Hampton observed that, when classifying items under the combined concept, people usually do not follow a rule corresponding to the intersection: the extension of the combination is very often over-extended in order to include items that are very good examples of one of the two concepts, but that do not belong to the extension of the other. In our setting, we may account for overextension appropriately lowering the threshold below the maximal possible value.

For instance, assume that the individual (constant) boxing is characterized by the axioms

[5] The situation is different when considering $\neg\exists has.SeriousGoal$ as a necessary feature of $Game$.

[6] In the example there are no incompatibilities between the features of $Sport{\circ}Game$ ($Game{\circ}Sport$), thus the best exemplars have all the features.

$PhysicalActivity(\text{boxing}), \exists doneFor.Fun(\text{boxing}), IsCompetitive(\text{boxing}),$
$\exists needs.AthleticAbilities(\text{boxing}), \exists doneFor.Health(\text{boxing}).$

According to the previous definitions of $Sport$ and $Game$, boxing is an instance of $Sport$ but not of $Game$. However, it is easy to chose a value for the threshold for $Sport{\circ}Game$ in the $(1.07, 1.37)$ range (determined by the algorithm in the phase 3) to include boxing among the instances of $Sport{\circ}Game$. It is enough to set the threshold in the interval $(1.07, 1.09]$.

5.1.3. Fish Vehicle

The Fish Vehicle example is another case analysed by Hampton [19]. While Sport Game is close to a plain conjunction, Fish Vehicle is an *impossible combination* due to a number of clashes between the features of the component concepts. Sport Game and Fish Vehicle can be seen as two extremes in the the spectrum of concept combinations. Assume that:

$$Fish = \mathbb{W}^{10}((\forall livesIn.Water, 3), (Animal, 3), (\forall breathesThrough.Gill, 3), (Grey, 0.9), (Scaly, 0.9), (\exists hasPart.Fin, 1))$$

$$Vehicle = \mathbb{W}^{10}((Artefact, 3), (\exists hasCapacity.MoveFast, 3), (Controlled, 3), (Handy, 1), (\exists hasComponent.SteeringSystem, 1), (\exists hasComponent.Seat, 0.9))$$

and consider the following KB:

$$Animal \sqsubseteq \neg Artefact \tag{5a}$$

$$\forall breathesThrough.Gill \sqsubseteq \neg Controlled \tag{5b}$$

$$Artefact \sqsubseteq \forall livesIn.\bot \tag{5c}$$

$$Artefact \sqsubseteq \forall breathesThrough.\bot \tag{5d}$$

Contrarily to the previous example, the Head and Modifier roles have here a strong impact in the combination. When $Vehicle$ is the Head, all the necessary features of $Fish$ are discarded from $\overline{\mathbf{ft}}(Vehicle{\circ}Fish)$ because they belong to $\mathbf{im}(Vehicle, \mathbf{ft}(Fish))$, i.e., they are all impossible for $Vehicle$. The combined concept would then be essentially a $Vehicle$, with few marginal characteristic of the $Fish$ (e.g., being $Grey$ and $Scaly$). Conversely, assuming the $Fish$ as the Head would exclude many of the necessary features of $Vehicle$ (i.e., $Artefact$ and $Controlled$), leading to the opposite result.

To obtain a more *hybrid* combination, a different mechanism is needed: see [20] for a computational treatment of the same example in the context of formal ontologies, through the procedure of axiom weakening. Similar combinations are also analysed in the context of Computational Conceptual Blending (CCB) (see, e.g., [21, 22]). CCB, however, originates from a quite different conceptual framework [23] and it exploits different technical strategies (i.e., the identification of shared structures between the two input spaces through a cross-space mapping). A thorough comparison of the two approaches is beyond the scope of this paper.

5.2. Adjective-Noun Combinations

Our proposal to combine concepts strongly refers to the work of Hampton that mainly focuses on noun-noun combinations. However, the most common cases of concept combinations discussed in the literature are adjective-noun combinations, e.g., *red apple* or *pink elephant*. In the following, we analyze some of the examples of this kind of combinations discussed in [24].

First observe that the concepts representing adjectives like red, smooth, salty, etc., usually, do not have a $\mathbb{W}$-definition. Given the fact that our algorithm applies only to $\mathbb{W}$-defined concepts, we introduce here $\mathbb{W}$-definitions logically equivalent to these 'adjective-concepts': for any 'adjective-concept' C we add the concept $C_{\mathbb{W}}$ defined as $C_{\mathbb{W}} = \mathbb{W}^{w}((C, w))$. In this way we assure that $C_{\mathbb{W}} \equiv C$, i.e., $C_{\mathbb{W}}$ is logically indistinguishable from C.

Red Book. Assume that the $\mathbb{W}$-definition of $Book$ does not contain any color-feature. Red is a necessary feature of $Red_{\mathbb{W}}$ therefore, assuming Red is compatible with all the features of $Book$, the tooth of $Red_{\mathbb{W}} \circ Book$ contains the union of the features of $Book$ and $Red_{\mathbb{W}}$. We also have that $Red_{\mathbb{W}} \circ Book \equiv Book \circ Red_{\mathbb{W}}$, more specifically, they have the same $\mathbb{W}$-definition.

Red Apple and Red Brick. Assume $Yellow \sqcup Red$ (or, more generally, a color-feature F_{col} such that $Red \sqsubseteq F_{col}$) is a feature of $Apple$. Red is a strongly necessary feature of $Red_{\mathbb{W}}$ therefore, assuming it is compatible with all the features of $Apple$, it belongs to $\bar{\mathbf{sft}}(Red_{\mathbb{W}} \circ Apple)$. It follows that $Yellow \sqcup Red$ is a (non-strongly) necessary feature of $Red_{\mathbb{W}} \circ Apple$ (because $Red \sqsubseteq Yellow \sqcup Red$). By assuming that KB contains $Yellow \sqsubseteq \neg Red$, the instances of $Red_{\mathbb{W}} \circ Apple$ cannot be yellow because they would lack the necessary feature Red. Again $Apple \circ Red_{\mathbb{W}} \equiv Red_{\mathbb{W}} \circ Apple$.

The case of Red Brick is similar to the one of Red Apple but now we have that $F_{col} \sqsubseteq Red$ where F_{col} is the color feature of Brick, i.e., the color of bricks is a specialization of Red. If F_{col} is a necessary feature of $Brick$, then all the instances of $Red_{\mathbb{W}} \circ Brick$ have the color F_{col} and $Red_{\mathbb{W}} \circ Brick \equiv Brick$, i.e., $Red_{\mathbb{W}}$ has no impact on $Brick$. If F_{col} is a non-necessary feature of $Brick$, then Red is (strongly) necessary for $Red_{\mathbb{W}} \circ Brick$ but it is still possible to have red bricks that do not have the color F_{col}. In both cases $Brick \circ Red_{\mathbb{W}} \equiv Red_{\mathbb{W}} \circ Brick$.

Red Fish. According to the previous tooth definition, $Grey$ is a non-necessary feature of $Fish$. Assume now that KB contains the axiom $Red \sqsubseteq \neg Grey$. $Red_{\mathbb{W}} \circ Fish$ would then have the strongly necessary feature Red which overrules the original feature $Grey$. Also in this case $Fish \circ Red_{\mathbb{W}} \equiv Red_{\mathbb{W}} \circ Fish$.

Pink Elephant. This case is similar to the one of Red Fish: $Grey$ is a feature of $Elephant$ and KB contains $Pink \sqsubseteq \neg Grey$. However, $Grey$ is now a necessary feature of $Elephant$ incompatible with $Pink$. In $Elephant \circ Pink_{\mathbb{W}}$, $Grey$ would be overruled by $Pink$ while $Pink_{\mathbb{W}} \circ Elephant \equiv Elephant$, i.e., by being the Modifier, $Pink_{\mathbb{W}}$ has no impact on $Elephant$.

These examples show that some expected characteristics of the adjective-noun combinations are obtained in our framework. For instance, when $Red_{\mathbb{W}}$ plays the modifier role, Red is a (strongly) necessary features of $Red_{\mathbb{W}} \circ Book$, $Red_{\mathbb{W}} \circ Apple$, and $Red_{\mathbb{W}} \circ Fish$. The case of the Pink Elephant is less satisfactory because $Pink$ overrules $Grey$ only when $Pink_{\mathbb{W}}$ is the head concept (and not the modifier as expected). Probably, the weakest aspect concerns the fact that $Red_{\mathbb{W}} \circ Book \equiv Book \circ Red_{\mathbb{W}}$, $Red_{\mathbb{W}} \circ Apple \equiv Apple \circ Red_{\mathbb{W}}$, and $Red_{\mathbb{W}} \circ Fish \equiv Fish \circ Red_{\mathbb{W}}$. These equivalences show that our algorithm does not take into account the strong asymmetry in the behaviour of adjectives vs. nouns, i.e., it is more tuned to noun-noun combinations where this asymmetry is less pronounced.

6. Discussion and Related Work

We proposed an algorithm for concept combination able to deal, within a logical framework, with some of the phenomena observed in cognitive and experimental psychology. The algorithm consists of three phases. First, it selects all the features needed for the combination, based on the logical distinction between necessary and impossible features. Second, it assigns new weights to the features of the combined concept trying to preserve the relevance of features. Finally, it determines the threshold to assure that the consistent and strongly necessary features are preserved in the combination.

Different assignments of weights and threshold can, however, be considered. A first alternative consists in strengthening the relevance of the necessary features. In order to do that, we can maintain the original weights of the necessary feature, collect the original weights of the non-necessary features, and then normalise them so that their sum remains strictly lower than the weight of any of the necessary features. To preserve the necessary features, it suffices to set the threshold in the interval between the sum of the weights of the necessary features and the sum of the weights of all the features, minus the highest weight of the non-necessary features.

Considering again the example about $Game$ and $Sport$, we obtain (where $t \in (3, 5.1)$):

$$
\begin{aligned}
Game{\circ}Sport = \mathbb{W}^t(&(PhysicalActivity, 3), (\exists doneFor.Fun, 0.5), (IsCompetitive, 0.3),\\
&(\exists doneFor.Health, 0.3), (\exists needs.AthleticAbilities, 0.3),\\
&(IsExciting, 0.4), (\neg\exists has.SeriousGoal, 0.4),\\
&(\exists requires.Concentration, 0.4))
\end{aligned}
$$

When $t = 3$ (the minimal value) the necessary features become also *sufficient* for the classification under $Game{\circ}Sport$. Being inspired by the work in [1], our model accounts mostly for the kind of combinations analysed there, namely conjunctive, noun-noun combinations such as *a Sport which is a Game* or *a Tool which is a Weapon*. This does not exhaust the whole spectrum of combinations. According to [25], there exist at least two other types of noun-noun combinations. Let's consider a *robin snake* [25, p. 168]: it may be interpreted simply as a snake with a *red under-belly*, namely in terms of a *property* interpretation. In these cases, only a single (maybe very salient) property of the Modifier applies to the Head. Our algorithm can be modified to deal with this kind of combinations by exploiting the flexibility of the weights assignment and by strengthening the role of the Head. Still, our procedure would return essentially a 'mesh-up' of the concepts being combined. Besides, a *robin snake* may be interpreted also as a snake that *eats* robin, i.e., according to a *relation-linking* interpretation, where the eat-relationship holding between the snake and the robin is crucial. To account for this kind of interpretations, our algorithm would require additional contextual information, or possibly the reference to specific background ontologies.[7]

In a formal context, noun-noun combinations have been analysed in [7] and [8]. [7] proposes a model for concept combination based on conceptual spaces (as firstly introduced in [24]).

[7]The example of Brick Red where we assume that Brick Red (but not Red Brick) represents a color (the color of bricks) seems to require a relation-linking interpretation.

Exploiting the idea of a *hierarchy* of conceptual spaces, the approach proposed in [7] is able to merge different spaces (corresponding to different concepts) to account for some of the phenomena observed in [1]. As acknowledged by the authors themselves, however, many of the analyzed combinations strongly rely on the probability distribution's choice associated to membership function for the combining concepts, which can in some cases lead to quite unexpected results (e.g., a tree being classified as a Pet Fish with a probability of 0.2). Also, any appeal to a logical inference mechanism being lacking in their model, the analysis of the notion of impossibility and necessity is based exclusively on the role of negation: the necessity of a dimension (a feature, in our setting) L corresponds to the impossibility of $\neg L$. To account for this notion, a direct negation of the features involved is needed, and it is unclear how this idea may be exploited in more complex scenarios, e.g., the Pet Fish example analyzed in Sect. 5.1.

[8] introduces a nonmonotonic Description Logic of typicality. Two kinds of properties can be associated to a concept C: (i) *rigid* properties define C (e.g., $C \sqsubseteq D$); while (ii) *typical* properties with form $p :: \mathbf{T}(C) \sqsubseteq D$ represent the degree of belief of the typicality inclusion of C into D. Distinguishing between a head concept H and a modifier concept M the proposed algorithm outputs the set of typical (and rigid) properties of $M{\circ}H$. To determine the properties of $M{\circ}H$, the algorithm selects the most probable *scenario* (i.e., a selection among the union of the typical properties of M and H) satisfying three conditions: (i) it is consistent (including the rigid properties); (ii) it is non trivial, i.e., it does not include all the typical properties of H; (iii) in case of couples of typical properties with form $p :: \mathbf{T}(H) \sqsubseteq D$ and $p' :: \mathbf{T}(M) \sqsubseteq \neg D$, the second one is discarded. Applied to Pet and $Fish$ the algorithm shows that $Greyish$ is a typical property of $Fish$ but not of $Pet{\circ}Fish$. However, first notice that condition (ii) is established *a priori*, i.e., it is not the result of a general combination mechanism. Our algorithm guarantees this property only when some non-necessary features of H are inconsistent (in KB) with the necessary features of M (which are consistent with all the necessary features of H). However, our algorithm can be easily modified to always discard some feature of H, even though this sounds quite artificial to us. Second, in our framework, condition (iii) somewhat corresponds to the rule discarding the necessary features of M inconsistent with the necessary features of H. However incompatible non-necessary features are not dicarded by our algorithm. Third, in [8] the possibility to rule out $Greyish$ depends not only on the degree of belief about $\mathbf{T}(Fish) \sqsubseteq Greyish$ but also on the number of typical properties of $Fish$ with a lower degree of belief. In our approach $Greyish$ is discarded only when it is inconsistent with the conjunction of the necessary feature of Pet (which are consistent with all the necessary features of $Fish$).

The approach proposed here is plunged in the Prototype Theory paradigm, both in terms of general inspiration and in terms of strategies adopted for the combination of concepts. At the same time, appealing to an external KB, most of the examples proposed here are also in debt with the so called *Theory View* on concepts, namely that idea that concepts cannot stand in isolation, but should be represented as micro-theories, expressing our knowledge about the world. To deal with this, we mostly exploited the KB in terms of TBox axioms, expressing the background knowledge needed to carry out the combinations. By taking into account ABox statements, expressing particular knowledge over individuals, we may also take into account the *Exemplar View* on concepts—namely the idea that the categorization under a concept is based on the exemplars stored in memory. This may be particularly useful in the context of the weights assignment in the combination procedure: as mentioned above, we could *tune* the weight of a feature according to the number of instances, within our ABox, satisfying that feature. This may be considered a sophistication of the phase 2 of our algorithm, exploiting what has been done in [14]. This is, however, matter for future work.

References

[1] J. A. Hampton, Inheritance of attributes in natural concept conjunctions, Memory & Cognition 15 (1987) 55–71.

[2] J. Fodor, E. Lepore, The red herring and the pet fish: Why concepts still can't be prototypes, Cognition 58 (1996) 253–270.

[3] E. Rosch, Principles of categorization, in: E. Margolis, S. Laurence (Eds.), Concepts: Core Readings, volume 189, MIT press, 1999, pp. 189–206.

[4] E. Rosch, Cognitive representations of semantic categories, Journal of Experimental Psychology:General 104 (1975) 192–233.

[5] G. L. Murphy, The Big Book of Concepts, MIT press, 2002.

[6] D. N. Osherson, E. E. Smith, On the adequacy of prototype theory as a theory of concepts, Cognition 9 (1981) 35–58.

[7] M. Lewis, J. Lawry, Hierarchical conceptual spaces for concept combination, Artif. Intell. 237 (2016) 204–227. URL: https://doi.org/10.1016/j.artint.2016.04.008. doi:10.1016/j.artint.2016.04.008.

[8] A. Lieto, G. L. Pozzato, A description logic framework for commonsense conceptual combination integrating typicality, probabilities and cognitive heuristics, J. Exp. Theor. Artif. Intell. 32 (2020) 769–804. URL: https://doi.org/10.1080/0952813X.2019.1672799. doi:10.1080/0952813X.2019.1672799.

[9] D. Porello, O. Kutz, G. Righetti, N. Troquard, P. Galliani, C. Masolo, A toothful of concepts: Towards a theory of weighted concept combination, in: M. Simkus, G. E. Weddell (Eds.), Proceedings of the 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019, volume 2373 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2373/paper-24.pdf.

[10] E. E. Smith, D. N. Osherson, L. J. Rips, M. Keane, Combining prototypes: A selective modification model, Cognitive Science 12 (1988) 485–527.

[11] G. Righetti, D. Porello, O. Kutz, N. Troquard, C. Masolo, Pink panthers and toothless tigers: three problems in classification., in: Proceedings of the 7th International Workshop on Artificial Intelligence and Cognition (AIC 2019), 2019, pp. 39–53.

[12] G. Righetti, P. Galliani, O. Kutz, D. Porello, C. Masolo, N. Troquard, Weighted description logic for classification problems, in: D. Calvanese, L. Iocchi (Eds.), GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence, Bozen/Bolzano, Italy, 17-19 September 2019, volume 65 of *EPiC Series in Computing*, EasyChair, 2019, pp. 108–112.

[13] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017. doi:10.1017/9781139025355.

[14] P. Galliani, G. Righetti, O. Kutz, D. Porello, N. Troquard, Perceptron connectives in knowledge representation, in: C. M. Keet, M. Dumontier (Eds.), Knowledge Engineering and Knowledge Management - 22nd International Conference, EKAW 2020, Bolzano,

Italy, September 16-20, 2020, Proceedings, volume 12387 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 183–193. URL: https://doi.org/10.1007/978-3-030-61244-3_13. doi:10.1007/978-3-030-61244-3_13.
[15] P. Galliani, O. Kutz, D. Porello, G. Righetti, N. Troquard, On knowledge dependence in weighted description logic, in: D. Calvanese, L. Iocchi (Eds.), GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence, Bozen/Bolzano, Italy, 17-19 September 2019, volume 65 of *EPiC Series in Computing*, EasyChair, 2019, pp. 68–80.
[16] R. Jackendoff, English noun-noun compounds in conceptual semantics, The semantics of compounding (2016) 15–37.
[17] J. A. Hampton, Overextension of conjunctive concepts: Evidence for a unitary model of concept typicality and class inclusion., Journal of Experimental Psychology: Learning, Memory, and Cognition 14 (1988) 12–32.
[18] L. Wittgenstein, Tractatus Logico-Philosophicus, New York: Routledge, 2001 [1921].
[19] J. A. Hampton, Compositionality and concepts, in: J. A. Hampton, Y. Winter (Eds.), Compositionality and Concepts in Linguistics and Psychology, Springer International Publishing, Cham, 2017, pp. 95–121.
[20] G. Righetti, D. Porello, N. Troquard, O. Kutz, M. Hedblom, P. Galliani, Asymmetric hybrids: Dialogues for computational concept combination, in: Formal Ontology in Information Systems: Proceedings of the 12th International Conference (FOIS 2021), IOS Press, 2021.
[21] M. Eppe, E. Maclean, R. Confalonieri, O. Kutz, M. Schorlemmer, E. Plaza, K.-U. Kühnberger, A computational framework for conceptual blending, Artificial Intelligence 256 (2018) 105–129.
[22] R. Confalonieri, M. Eppe, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, Upward refinement operators for conceptual blending in the description logic $\mathcal{EL}^{++}$, Ann. Math. Artif. Intell. 82 (2018) 69–99. URL: https://doi.org/10.1007/s10472-016-9524-8. doi:10.1007/s10472-016-9524-8.
[23] G. Fauconnier, M. Turner, The Way We Think: Conceptual Blending and the Mind's Hidden Complexities, Basic Books, New York, 2003.
[24] P. Gärdenfors, Conceptual spaces - The geometry of thought, MIT Press, 2000.
[25] E. J. Wisniewski, When concepts combine, Psychonomic bulletin & review 4 (1997) 167–183.