TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT FÜR INFORMATIK
INSTITUT FÜR THEORETISCHE INFORMATIK
LEHRSTUHL FÜR AUTOMATENTHEORIE

Simi-Framework for Concept-Similarity-Measures: property analysis and expansion to more powerful Description Logics

Diplomarbeit
zur Erlangung des akademischen Grades
Diplom-Informatiker

Autor:
Paul Lätsch
MNr. 3514105

Betreuerin:
PD. Dr.-Ing. habil.
Anni-Yasmin Turhan

07.12.2015

Erklärung

Hiermit erkläre ich, dass ich die am heutigen Tag eingereichte Diplomarbeit mit dem Titel

Simi-Framework for Concept-Similarity-Measures: property analysis and expansion to more powerful Description Logics

unter Betreuung von Dr.-Ing. Anni-Yasmin Turhan selbständig erarbeitet, verfasst und alle Zitate kenntlich gemacht habe. Andere als die von mir angegebenen Hilfsmittel wurden von mir nicht benutzt.

Dresden, den 07.12.2015

Contents

1 Introduction

In our modern times, the growing amount of data requires efficient procedures to handle it. Amongst other areas, the semantic web and science have a demand for such tools. This includes queries to explicit data and the deriving of new knowledge. Therefore high-level representation and reasoning formalisms have to be invented and investigated. One well-investigated family of such formalisms are *Description Logics* (DLs). They allow conceptual knowledge representation and there already exist several reasoning procedures and relations to other logics, like first-order logic and modal logics. One example of the usage of DLs is the Web ontology language: OWL DL.

Typically these DL just work with *crisp knowledge*, meaning that an individual is an instance of an certain concept, or not. Because these restrictions are some times too strong for practical use, as an alternative relaxed queries are getting more attention. Such not just evaluate whether an individual belongs to a concept or not, but whether it is considered 'good enough' to be responded to match the query.

One way to achieve this, are *Concept Similarity Measures* (CSMs). Given two concept descriptions, a CSM calculates an value between *0* and *1*. The idea is, that the higher this similarity value, the more similar are the both concept descriptions. Further more should the scaling of this value match human intuition. So *0* should stand for total dissimilarity and *1* stand for total similarity. Also the semantics should be essential for the calculation of the similarity, which means that just syntactical differences should be irrelevant. Besides this, there is an interest to find CSMs that have metric properties, in particular the triangle inequality from metric spaces. Of course for CSM it has to be adapted to the interval $[0, 1]$, but it would ensure that the similarity values from two concept descriptions to a third one could just differ within a certain range, that is related to the similarity value of the first two ones. Unfortunately this property is for now not well accomplished.

In [4] Karsten Lehmann provides with the *simi* framework similarity measures for $\mathcal{EL}$, a commonly used DL. He introduced formal properties (also covering the ones previously mentioned) and proved whether or not *simi* fulfils those. In this paper, we want to expand *simi* by new constructors and analyse possible relations between the formal properties from [4]. Because the plain definitions of these formal properties do not provide usable relations between then, we want to try the method of *Formal Concept Analysis* (FCA). This is a method, that derives implications between these properties from a data set of objects. This implications just hold within the context of the data set, but their generality could be proved or they could give a hint of possible relations between the properties.

In this paper we restrict our works to deterministic and symmetric CSMs. Chapter 2 introduces the basic needs for DLs, *simi*, the formal properties and other basic notations. In Chapter 3 we expand *simi* and analyse under which circumstances this CSM provide the formal properties for DLs with certain constructors. In Chapter 4 we introduce some modifications for the rules of the CSM to change their behaviour respectively to the fulfilling of the formal properties. With the knowledge of Chapter 3 and Chapter 4 we are able to produce a big data set of CSM and their properties. This allows us in Chapter 5 to use the FCA tool *Con-Exp* to infer implications between the formal properties. Here we will also prove the generality of some of them and outline the limits of our research for the other implications.

2 Preliminaries

In this chapter we introduce the basics for DLs and triangular norms and conorms. This definitions are taken from [1]. For RBoxes we stay with the definition from [4]. Also we recap the concept similarity measure *simi* and the Jaccard Index from [4]. At last we will shortly sum up our preconditions for this paper.

2.1 Description Logics

Description logics essentially are composed of a set of concept names N_C, a *set of role names* N_R, the constants $\top$ and $\bot$ and constructors to combine those to complex concept descriptions. To refer to a DL in general, we want to use $\mathcal{DL}$.
We can use a DL $\mathcal{DL}$ to represent knowledge in the form of a knowledge base, composed of a TBox, a RBox and an ABox. The TBox contains terminological knowledge in the form of axioms about the relations between concept descriptions of $\mathcal{DL}$. The same way the RBox contains relations between the role names in N_R. At last the ABox contains assertional knowledge in the form of axioms that link between concept descriptions and individuals.
In the following we introduce concept descriptions, TBoxes, RBoxes, ABoxes and rules to receive an unique normal form.

2.1.1 Concept Descriptions

Since we will take a look at different DLs, with variations of constructor sets, we formulate the definition for concept descriptions in a way, that suites them all.

Definition 1 (concept descriptions)**.** *For a specific description logic $\mathcal{DL}$, with a set of concept names N_C, a set of role names N_R, the constants $\top$, $\bot$ and a variety of constructors from $\{\neg, \sqcap, \sqcup, \exists, \forall\}$, we define that:*

- *the constants $\top$, $\bot$ and all $A \in N_C$ are $\mathcal{DL}$ concept descriptions*
- *C is a $\mathcal{DL}$ concept description and $\neg$ is a constructor of $\mathcal{DL}$, then $\neg C$ is a $\mathcal{DL}$ concept description*
- *C, D are $\mathcal{DL}$ concept descriptions and $\sqcap$ is a constructor of $\mathcal{DL}$, then $C \sqcap D$ is a $\mathcal{DL}$ concept description*
- *C, D are $\mathcal{DL}$ concept descriptions and $\sqcup$ is a constructor of $\mathcal{DL}$, then $C \sqcup D$ is a $\mathcal{DL}$ concept description*
- *C is a $\mathcal{DL}$ concept description, $r \in N_R$ and $\exists$ is a constructor of $\mathcal{DL}$, then $\exists r.C$ is a $\mathcal{DL}$ concept description*

- *C is a $\mathcal{DL}$ concept description, $r \in N_R$ and $\forall$ is a constructor of $\mathcal{DL}$, then $\forall r.C$ is a $\mathcal{DL}$ concept description*

We define $\mathcal{C}(\mathcal{DL})$ as the set of all $\mathcal{DL}$ concept description. Further we define $const(\mathcal{DL})$ as the Set of constructors of $\mathcal{DL}$.

The semantics of DLs are defined by using interpretations $\mathcal{I}$. Such interpretations are composed of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$, which maps the concept descriptions to subsets of $\Delta^{\mathcal{I}}$.

Definition 2 (interpretation)**.** *An* interpretation *is a pair $\mathcal{I}= (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where the domain $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a function mapping every $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and every $r \in N_R$ to a binary relation $r^I \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This function is extended to the typical $\mathcal{DL}$ constructors as follows:*

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ *and* $\bot^{\mathcal{I}} = \varnothing$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}}/C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = (C^{\mathcal{I}}) \cap (D^{\mathcal{I}})$
- $(C \sqcup D)^{\mathcal{I}} = (C^{\mathcal{I}}) \cup (D^{\mathcal{I}})$
- $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
- $(\forall r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \forall y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \longrightarrow y \in C^{\mathcal{I}}\}$.

where $C, D \in \mathcal{C}(\mathcal{DL})$ and $r \in N_R$.

This definition covers rules for all constructors we use later on. Since a rule can just not be used, if the corresponding constructor is not in $const(\mathcal{DL})$, it suites for all $\mathcal{DL}$s we will cover in this paper.

2.1.2 TBoxes and RBoxes

The TBox presents terminological knowledge in the form of concept axioms. The most general form of concept axioms are general concept inclusion. They are formulas that indicate the subsumption of one concept description by an other. The most common axioms are concept definitions, which indicate the definition of one concept name by a concept description.

Definition 3 (GCI and concept definition)**.** *A* general concept inclusion axiom (GCI) *is of the form $C \sqsubseteq D$, where C, D are concept descriptions. If the axiom is of the form $A \equiv C$ where $A \in N_C$ and C is a concept description, then it is called a* concept definition. *All $B \in N_C$ that occur in C are called directly used by A and A is called defined.*

Definition 4 (TBox)**.** *A* TBox $\mathcal{T}$ *is a finite set of concept axioms. A* general TBox $\mathcal{T}$ *is a finite set of GCI's.*

Since we later want to use TBoxes, that do not contain GCI's, we introduce normalized TBoxes. All other TBoxes we will refer to as general TBoxes.

Definition 5 (normalized TBox). *A TBox $\mathcal{T}$ containing just concept definitions is called* normalized. *Concept names that just never occur on the left hand sides of a concept definitions are called* primitive concepts. *$\mathcal{T}$ is called* acyclic, *if none of the defined concepts uses itself (directly or under the closure of transitivity), otherwise it is called* cyclic.
Axioms of the form $A \sqsubseteq D$, where $A \in N_C$ and D is a concept description, can be transformed to concepts definitions by introducing a new concept name A':

$$A \sqsubseteq D \longrightarrow A \equiv A' \sqcap D$$

Note that for the normalisation the new introduced concept names not occur in N_C before. Also note, that the normalisation of those subsumption axioms requires the conjunction $\sqcap$ to be valid constructor in the used DL.
The semantics for these axioms are also handled by interpretations. These are models of a TBox $\mathcal{T}$, if they interpret the axioms of $\mathcal{T}$ in a consistent way.

Definition 6 (TBox model). *An interpretation $\mathcal{I}$ is a* model *of a TBox $\mathcal{T}$ ($\mathcal{I} \models \mathcal{T}$) if it satisfies all of $\mathcal{T}$'s axioms, meaning:*

- *for all $C \sqsubseteq D$ holds: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$*
- *for all $C \equiv D$ holds: $C^{\mathcal{I}} = D^{\mathcal{I}}$.*

Two TBoxes $\mathcal{T}$ and $\mathcal{T}^*$ are equivalent *($\mathcal{T} \equiv \mathcal{T}^*$), if they have the same models.*

As a standard form of TBoxes for this paper, we want them to be normalized and extended. If we have a DL $\mathcal{DL}$ with $\sqcap \notin const(\mathcal{DL})$, then on order to be normalized, the TBox should contain of just concept definitions in the first way, because a normalisation of subsumption axioms would lead to terms not in $\mathcal{C}(\mathcal{DL})$. By extending a TBox $\mathcal{T}$, we want to accomplish that the left hand side of all concept definition in $\mathcal{T}$ is only composed of primitive concept names. Therefore $\mathcal{T}$ has to be unfoldable.

Definition 7 (unfoldable TBox). *A TBox $\mathcal{T}$ is called* unfoldable, *if it is acyclic, consists just of concept definitions and ever defined concept occurs at most ones to the left hand side of an axiom.*
Every unfoldable TBox $\mathcal{T}$ can be extended to a TBox $\mathcal{T}^*$, where on the right hand side of the axioms occur only primitive concepts, so that $\mathcal{T} \equiv \mathcal{T}^*$.

For the rest of this paper, we want all TBoxes $\mathcal{T}$ to be unfoldable, normalized and extended. Further more we assume for all concept description, that they are extended with the concept definitions from $\mathcal{T}$, so that they are only composed of primitive concept names, role names and constructors. So we do not have to regard TBoxes for the rest of the paper.

Similar to TBoxes, RBoxes contain axioms about relations between role names. We call them role inclusions. Since we do not assume more complex ways to build such role inclusions, the structure of RBoxes is more simple.

Definition 8 (RBox). *A* role inclusion *is an axiom of the form $r \sqsubseteq s$, where $r, s \in N_R$. A* RBox $\mathcal{R}$ *is a finite set of role inclusion axioms*

Again the semantics are handled by interpretations, which act as models.

Definition 9 (RBox model). *An interpretation $\mathcal{I}$ is a* model *of a RBox $\mathcal{R}$ if it satisfies all of $\mathcal{R}$'s axioms, meaning that for all $r \sqsubseteq s$ holds $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.*

For the rest of the paper we want to say, that

$$r \sqsubseteq_{\mathcal{R}} s \iff \forall \mathcal{I} \models \mathcal{R} : r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$$

2.1.3 ABoxes

For completeness, we also introduce ABoxes. They contain assertional knowledge in the from of concept and role assertion axiom. Therefore we need a finite set N_I of individuals. Those individuals can be asserted to concept descriptions, or directed pairs of them to role names.

Definition 10 (ABox). *Let N_I be finite set of individual, $x, y \in N_I$ C be a concept description and $r \in N_R$. We call $C(x)$ a* concept assertion axiom *and $r(x, y)$ a* roll assertion axiom. *An* ABox $\mathcal{A}$*is a finite set of concept and role assertions.*

Also here the semantics are handled by interpretations, which act as models.

Definition 11 (ABox model). *An interpretation $\mathcal{I}$ in a* model *of an ABox $\mathcal{A}$if it satisfies all its axioms, meaning that for all $C(x)$ holds $x^{\mathcal{I}} \in C^{\mathcal{I}}$ and for all $r(x, y)$ holds $(x^{\mathcal{I}}, y^{\mathcal{I}}) \in r^{\mathcal{I}}$.*

For this paper, ABoxes will be of no relevance, since we will not work with knowledge bases directly. More detail to ABoxes can be read in [1].

2.1.4 Normal Form

At some point, the existence of an unique normal form for our DLs will be relevant. The concrete normal form is nonrelevant, so we just present some transformation rules, that will achieve a unique normal form. If a rule is not applicable because the corresponding constructors does not exist in the DL, we can just ignore it.
Before we formulate these transformation rules, we want to introduce a property for normal forms, that will prevent trivial violations of one the later following formal properties.

Definition 12 (name unique). *Let $\mathcal{DL}$ be a specific description logic, $\mathcal{R}$ be a RBox and $\mathcal{T}$ be a TBox. We define the operator $[\cdot]$ that maps every $r \in N_R$ to a set of role names*

$$[r] = \{s \in N_R | s \equiv_{\mathcal{R}} r\}$$

and every $A \in N_C$ to a set of concept names

$$[A] = \{B \in N_C | B \equiv_{\mathcal{T}} A\}$$

We want to call a (normal) form for $\mathcal{DL}$ name unique*, if for all primitive concept $A \in N_{C_{pr}}$ names hold that there is uniquely one $A' \in [A]$ and every $B \in [A]$ is represented by A'. And also for all role names $r \in N_R$ hold that there is uniquely one $r' \in [r]$ and every $s \in [r]$ is represented by r'.*

This property assures that all elements of a set of equivalent role names or equivalent concept names are represented by the same one uniquely chosen element of the set. By this property we want to prevent, that there exist pairs of role names or primitive concept names, that are equivalent to each other, but have different similarities to other role names or primitive concept names. Technically those still exist, but by using a name unique (normal) form, we choose one of the participants of each of these sets and substitute the others, so they do not occur in concept descriptions.
Since we will cover many DLs, we do not introduce an extra normal form for each of them, but a set of transformation rules, that will achieve an unique normal form. Let $\mathcal{T}$ be a unfoldable, normalized and extended TBox, $\mathcal{R}$ be a RBox and A, C, D be concept descriptions already extended with the concept definitions from $\mathcal{T}$, then an unique normal form will be achieved by applying the following rules (if possible):

1. $A \sqcap \top \longrightarrow A$
2. $A \sqcup \top \longrightarrow \top$
3. $A \sqcup \bot \longrightarrow A$
4. $A \sqcap \bot \longrightarrow \bot$
5. $A \sqcap A \longrightarrow A$
6. $A \sqcup A \longrightarrow A$
7. $\neg(C \sqcup D) \longrightarrow \neg C \sqcap \neg D$
8. $\neg(C \sqcap D) \longrightarrow \neg C \sqcup \neg D$
9. $A \sqcup (C \sqcup D) \longrightarrow A \sqcup C \sqcup D$
10. $A \sqcap (C \sqcap D) \longrightarrow A \sqcap C \sqcap D$
11. $\neg \forall r.C \longrightarrow \exists r.\neg C$
12. $\neg \exists r.C \longrightarrow \forall r.\neg C$
13. $\exists r.C \sqcup \exists r.D \longrightarrow \exists r.(C \sqcup D)$
14. $\forall r.C \sqcap \forall r.D \longrightarrow \forall r.(C \sqcap D)$
15. $\exists r.C \sqcap \exists s.D \longrightarrow \exists r.C$, if $r \sqsubseteq_{\mathcal{R}} s$ and $C \sqsubseteq D$
16. $\forall r.C \sqcup \forall s.D \longrightarrow \forall r.C$, if $s \sqsubseteq_{\mathcal{R}} r$ and $D \sqsubseteq C$

17. $\forall r.C \sqcap \exists r.C \longrightarrow \forall r.C \sqcap \exists r.(C \sqcap D)$
18. $A \sqcap (C \sqcup D) \longrightarrow (A \sqcap C) \sqcup (A \sqcap D)$

until no more rule can be applied.
This rules are partly taken from [4] and just completed by rules for the additional constructors and the new constant $\bot$. We also can exchange the last rule by:

$$A \sqcup (C \sqcap D) \longrightarrow (A \sqcup C) \sqcap (A \sqcup D)$$

The normal form would still be unique, just the conjunction would be the more outer constructor, instead of the disjunction. Note that this set of rules will just be sufficient for the DLs we cover in this paper. For example they all will need the negation to be just in front of primitive concept names.

2.2 Triangular Norms and Conorms

Triangular norms (t-norms) are a generalisation of conjunctions in propositional logics. As in [4] we will use then to make an asymmetric CSM symmetric. Their the absorbing element *0*, neutral element *1* and monotonicity property will be essential to obtain some of the later defined formal properties. We take the definitions form [4].

Definition 13 (t-norm)**.** $\otimes : [0,1]^2 \longrightarrow [0,1]$ *is a* triangular norm *if for all* $x, y, z, w \in [0,1]$ *holds:*

- $x \otimes y = y \otimes x$ *(Commutativity)*
- $x \leq z \wedge y \leq w \Longrightarrow x \otimes y \leq z \otimes w$ *(Monotonicity)*
- $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ *(Associativity)*
- $x \otimes 1 = x$ *(Identity)*

Dual to t-norm are triangular conorms (t-conorm), which are generalisations of disjunctions. Here *0* is the neutral and *1* the absorbing element.

Definition 14 (t-conorm)**.** $\oplus : [0,1]^2 \longrightarrow [0,1]$ *is a* triangular conorm *if for all* $x, y, z, w \in [0,1]$ *holds:*

- $x \oplus y = y \oplus x$ *(Commutativity)*
- $x \leq z \wedge y \leq w \Longrightarrow x \oplus y \leq z \oplus w$ *(Monotonicity)*
- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ *(Associativity)*
- $x \oplus 0 = x$ *(Identity)*

Definition 15 (duality of t-norm and t-conorm)**.** *For every t-norm* $\otimes$ *and its dual t-conorm* $\oplus$ *it holds that:*

- $x \oplus y = 1 - ((1-x) \otimes (1-y))$
- $x \otimes y = 1 - ((1-x) \oplus (1-y))$

The duality will not be important for this paper, but can be used to make a proof of the absorbing elements.

Claim 1 (absorbing elements)**.** *Let $\otimes$ be a t-norm and $\oplus$ be a t-conorm. Then holds:*

- $0 \otimes x = 0$
- $1 \oplus x = 1$

The proof of this claim is rather simple and can be found in [4]. In particular, we will just speak about three t-norms, which are the Product t-norm $\otimes_{prod}$, the Gödel t-norm $\otimes_{min}$ (or minimum t-norm) and the Hamanach product $\otimes_{H_0}$.

$$\begin{aligned} x \otimes_{prod} y &= xy \\ x \otimes_{min} y &= min(x, y) \\ x \otimes_{H_0} y &= \begin{cases} 0 & x = y = 0 \\ \dfrac{xy}{x + y - xy} & otherwise \end{cases} \end{aligned}$$

$x, y \in [0, 1]$

In terms of analyses we favour the Gödel t-norm, because of its predictable result. Returning one of its argument, allows stronger conclusions about the properties of its results. Product t-norm an Hamanach product will allow us some proofs later on in the paper.

2.3 The Simi-Framework

Since we are going to expand *simi* form [4], we will give a brief overview of its components and *simi* it self. We also already do a little rephrasing by expanding some definitions, so they can later also be used for new constructors like disjunction and value-quantification. At last we will take a look at the in [4] defined formal properties and which are fulfilled by *simi*.

2.3.1 Simi

In [4] Karsten Lehmann uses an operator $\widehat{(\cdot)}$ to gather the participants of the conjunction $\sqcap$. We define different operator, that will also work on disjunctions and negations.

Definition 16 (first level $\bigcirc$-participant set). *Let C be a not trivial $\mathcal{ALC}$-concept description. Then we define $\widehat{C}_{\bigcirc}$ as the* first level participant set, *which contains all participants of the most outer operator $\bigcirc \in \{\sqcap, \sqcup, \neg\}$ of C, so that:*

$$C = \bigcirc_{C' \in \widehat{C}_{\bigcirc}} C'$$

In all other cases, we define $\widehat{C}_{\bigcirc}$ to be empty.

Also we do not cover negations in general in this paper, we already included them in this definition, so future work can benefit form it. Note that for $\widehat{(\cdot)}_{\sqcap}$ and $\widehat{(\cdot)}_{\sqcup}$ the set will hold exactly one element, when the term is not actually a conjunction or disjunction, name the term itself. In the rules of the CSMs this must be handled by the condition, that the set must at least hold two elements, before the suitable conjunction or disjunction rule is applied.
We also expand the weighting function from [4], so we not have to just redefine it later.

Definition 17 (weighting function for $\mathcal{DL}$). *Let $\mathcal{DL}$ be a specific description logic and f be an arbitrary mapping $f : N_C \cup N_R \longrightarrow [0, 1]$. Then we can define the* weighting function $g_{\mathcal{DL}} : \mathcal{DL} \longrightarrow \mathbb{R}_{>0}$ *as:*

$$g_{\mathcal{DL}}(C) := \begin{cases} f(A') & \text{if } C \text{ is a Literal and } A' \text{ is its concept name} \\ g_{\mathcal{DL}}(C') & \text{if } C \text{ is of the form } \neg C' \\ f(r) g_{\mathcal{DL}}(C') & \text{if } C \text{ is of the form } \exists r.C' \text{ or } \forall r.C' \\ \prod_{C' \in \widehat{C}_{\bigcirc}} g_{\mathcal{DL}}(C') & \text{if } C \text{ is a conjunction or disjunction} \end{cases}$$

Because some circumstances may not need any concepts to be weighted, a default weighting function is defined, which weights every concept with *1*.

Definition 18 (default weighting function for $\mathcal{DL}$). $g_{\mathcal{DL},def} : \mathcal{DL} \longrightarrow \mathbb{R}_{>0}$ *with* $\forall C \in \mathcal{DL} : g_{\mathcal{DL},def}(C) := 1$, *is the* default weighting function *for $\mathcal{DL}$.*

For most of this paper we will assume the use of $g_{\mathcal{DL},def}$ to eliminate a variance factor and because $g_{\mathcal{DL}}$ has no impact to most of the formal properties. If we not use $g_{\mathcal{DL},def}$, we will refer to the weighting function explicitly.
To measure the similarity between primitive concepts or role names, Karsten Lehmann introduces *primitive measures* (*pm*). The values a *pm* maps to, have to be pre-designed. This can be done by assumptions or experiences or statistical analyses of pre-measured data. For the case that no pre-design is made or as a starting point for designing such a *pm*, Karsten Lehmann introduced in his work a default primitive measure pm_{def}. Since we will replace *pm* by an other measure and pm_{def} is not needed to understand *simi*, we do not define pm_{def} explicitly.

Definition 19 (primitive measure). *A function $pm : N_{C_{pr}}^2 \cup N_R^2 \longrightarrow [0, 1]$ is a* primitive measure, *if for all $A, B \in N_{C_{pr}}$ and $r, s, t \in N_R$ holds:*

- $pm(A, B) = 1 \iff A = B$,

- $pm(r,s) = 1 \Longleftrightarrow s \sqsubseteq r$,
- $s \sqsubseteq_{\mathcal{R}} r \Longrightarrow pm(s,r) > 0$ *and*
- $t \sqsubseteq_{\mathcal{R}} s \Longrightarrow pm(r,s) \leq pm(r,t)$

Note that a *pm* do not have to be symmetric. The last component is a fuzzy connector, which allows us to make *simi* commutative and so symmetric.

Definition 20 (fuzzy connector)**.** *We define that* $\otimes : [0,1]^2 \longrightarrow [0,1]$ *is a* fuzzy connector *if for all* $x, y \in [0,1]$ *holds:*

- $x \otimes y = y \otimes x$ *(Commutativity)*
- $x \otimes y = 1 \Longleftrightarrow x = y = 1$ *(Equivalence closed)*
- $x \leq y \Longrightarrow 1 \otimes x \leq 1 \otimes y$ *(Weak monotonicity)*
- $x \otimes y = 0 \Longrightarrow x = 0 \vee y = 0$ *(Bounded)*
- $0 \otimes 0 = 0$ *(Grounded)*

Karsten Lehmann also shows [4], that every bounded t-norm is a fuzzy connector. In fact, we want to limit the fuzzy connector to bounded t-norms, to have *simi* and its expansions fulfil more formal properties.
With all these components defined, we can present the definition of *simi*. Itself is just a connection of an inner asymmetric CSM $simi_d$ used in both directions. For the connection, the fuzzy connector is used.

Definition 21 (simi)**.** *For* $C, D, E, F \in \mathcal{ELH}$, $\otimes$ *being a fuzzy connector,* pm^{asym} *a primitive measure, and* $g_{\mathcal{ELH}}$ *a weighting function, the concept similarity measure* $simi : \mathcal{C}(\mathcal{ELH})^2 \longrightarrow [0,1]$ *is defined as follows:*

$$simi(C,D) = simi_d(C,D) \otimes simi_d(D,C)$$

with:

$$simi_d(C,D) = \begin{cases} \dfrac{\sum_{C' \in \widehat{C}_\sqcap} [g(C') * \bigoplus_{D' \in \widehat{D}_\sqcap} simi_d(C',D')]}{\sum_{C' \in \widehat{C}_\sqcap} g_{\mathcal{ELH}}(C')} & \text{if } C \neq \top \text{ and } |\widehat{C}_\sqcap| > 1 \text{ or } |\widehat{D}_\sqcap| > 1 \\ 1 & \text{if } C = \top \\ pm(C,D) & \text{if } C, D \in N_C \\ pm(r,s)[w + (1-w) simi_d(E,F)] & \text{if } C = \exists r.E \text{ and } D = \exists s.F \\ 0 & \text{otherwise} \end{cases}$$

and a suitable $w > 0$.

Note that the only changes to the version of Karsten Lehmann are the weighting function $g_{\mathcal{ELH}}$ and the operator $\widehat{(\cdot)}$, which are just expanded to handle more constructors. So *simi* is still equal. Also note that it can be possible, that all participants of a conjunction are weighted *0*, the calculation will not be defined. For this paper we want to stay with $g_{\mathcal{DL},def}$, so this case is temporarily covered. Because *simi* is designed for multiple combinations of the previously introduced components, formally this components are specified in the following order:

$simi$[fuzzy connector, t-conorm, primitive measure, weighting function, ω]

For shorter formulas we want to refer *simi* and its expansions by dropping the brackets. If needed, we will refer the components in an other way. Also still referred to as fuzzy connector, please remember our limitation on the fuzzy connectors to bounded t-norms.

2.3.2 Formal Properties for CSMs

In the following we took the similarity-measure properties from [4] and defined them for a general CSM *sim*.

Definition 22 (similarity-measure properties)**.** *Let sim be a similarity measure on a specific $\mathcal{DL}$, $C, D, E, F, U, L \in \mathcal{DL}$, $(C_n)_n$ be sequences of concept description with $\forall i, j \in N, i \neq j : C_i \not\sqsubseteq C_j$ and lcs the least common subsumer. We define the following properties:*

- ***Triangle Inequality***: $1 + sim(E, D) \geq sim(E, C) + sim(C, D)$
- ***Equivalence Invariant***: $C \equiv D \Longrightarrow sim(C, E) = sim(D, E)$
- ***Equivalence Closed***: $sim(C, D) = 1 \Longleftrightarrow C \equiv D$
- ***Subsumption Preserving***: $C \sqsubseteq D \sqsubseteq E \Longrightarrow sim(C, D) \geq simi(C, E)$
- ***Reverse Subsumption Preserving***: $C \sqsubseteq D \sqsubseteq E \Longrightarrow sim(C, E) \leq sim(D, E)$
- ***Dissimilar Closed***: $C \not\equiv \top \wedge D \not\equiv \top \wedge lcs(C, D) \equiv \top \Longrightarrow sim(C, D) = 0$
- ***Bounded***: $C \not\equiv \bot \wedge D \not\equiv \bot \wedge lcs(C, D) \not\equiv \top \Longrightarrow sim(C, D) > 0$
- ***Structural Dependent for*** $\sqcap$*: for all sequences $(C_n)_n$ of atoms with $\forall i, j \in \mathbb{N}, i \neq j : C_i \not\sqsupseteq C_j$ and for all $E, D \in \mathcal{C}(\mathcal{DL})$ the concept descriptions*

  $$D_n := \sqcap_{i<n} C_i \sqcap D$$
  $$E_n := \sqcap_{i<n} C_i \sqcap E$$

  fulfil:

  $$\lim_{n\to\infty} sim(D_n, E_n) = 1$$

- ***Structural Dependent for*** $\sqcup$*: for all sequences $(C_n)_n$ of atoms with $\forall i, j \in \mathbb{N}, i \neq j : C_i \not\sqsupseteq C_j$ and for all $E, D \in \mathcal{C}(\mathcal{DL})$ the concept descriptions*

  $$D_n := \sqcup_{i<n} C_i \sqcup D$$
  $$E_n := \sqcup_{i<n} C_i \sqcup E$$

  fulfil:

$$\lim_{n\to\infty} sim(D_n, E_n) = 1$$

Since we will expand to the disjunction $\sqcup$ as a constructor, we changed Structural Dependent from [4] to Structural Dependent for $\sqcap$ and adapted it to a new property: Structural Dependent for $\sqcup$. We want to assume, a CSM working on a specific DL $\mathcal{DL}$ can just fulfil Structural Dependent for $\bigcirc \in \{\sqcap, \sqcup\}$, if $\bigcirc \in const(\mathcal{DL})$.
Karsten Lehmann shows that *simi* fulfils the properties Equivalence Invariant, Equivalence Closed, Subsumption Preserving and Bounded. For the assumption that the weighting function g maps every concept to a value bigger than *0*, *simi* is Structural Dependent for $\sqcup$. Furthermore for pm_{def}, *simi* is also Dissimilar Closed. He also shows, that *simi* can generalize the Jaccard Index, a similarity measure for sets.
Note that *simi* is specific designed to fulfil Subsumption Preserving. It benefits from *1* being the neutral element for every bounded t-norm and the use of the directed measure $simi_d$.
More detail to *simi* and its properties can be found in [4].

2.4 Jaccard Index

Karsten Lehmann introduced in [4] the Jaccard Index as a structural measure, that can be used to calculate similarities between sets of concept names. It actually is an adaptation of a set similarity measure from [3] and accomplished to fulfil almost all formal properties from 2.3.2. The essential idea of the Jaccard Index is to compare the shared information of both sets with the overall information in both sets. With the operator $\widehat{(\cdot)}$, defined by Karsten Lehmann, the Jaccard Index *Jacc* is defined as:

$$Jacc(C, D) := \frac{|\widehat{C} \cap \widehat{D}|}{|\widehat{C} \cup \widehat{D}|}$$

As already mentioned, we expand this operator to the first level $\bigcirc$-participant set operator. This allows us to use the Jaccard Index not only on sets of concept names, but also for sets of participants of a certain constructor $\bigcirc$. Such an usage needs some restrictions. So we have to ensure, that participants contained in both sets, effect the result with a *1* and all other participants influence the result with a *0*. By this, the calculation of the Jaccard Index will remain. In general, we will not define the rules in the way of the Jaccard Index. But if possible, we define them in a way, that enables us to to **break the calculation down to the Jaccard Index**, if some restrictions are fulfilled. Within this paper the Jaccard Index will stay a major tool to achieve the formal properties, especially Subsumption Preserving, Reverse Subsumption Preserving and the fulfilling of the Triangle Inequality. In fact, breaking the calculation down to the Jaccard Index will stay nearly the only sufficient way to achieve the Triangle Inequality.

2.5 Precondition Summary

For clarity we here summarize our preconditions for this paper. We will look at CSMs that are symmetric and deterministic. For those, that use a fuzzy connector like Karsten Lehmann does in *simi*, we assume this fuzzy connector to be a bounded t-norm. Those working on a specific DLs $\mathcal{DL}$ and using a weighting function, will in general use $g_{\mathcal{DL},def}$. The DLs the CSMs work on, will be restricted to the constructors $\neg, \sqcap, \sqcup, \exists, \forall$ and allow the constants $\top$ and $\bot$. Possible given TBoxes are assumed to be unfoldable, normalized and extended. Possible given RBoxes just contain simple role inclusion axioms. We assume every concept description to be extended with the terminological knowledge from the TBox, so they only contain primitive concept names. Also we assume them to be at least in a name unique (normal) form, to prevent trivial violations of Equivalence Invariance. If we need them to be in an unique normal form, we want to achieve this by using the rules in 2.1.4.

3 Simple Expansions and their Influence on the Formal Properties

In this chapter we introduce some simple expansions of *simi*. As new constructors there will be the disjunction $\sqcup$, the $\forall$-quantification and a lighter version of the negation, the primitive negation $(\neg)$. First we will distinguish between three different expansion types for *simi*. Then we investigate for all of the formal properties, under which circumstances they are fulfilled by one of the defined CSM, that works on a DL with certain constructors. This will give us a big database of concept descriptions for a later analysis of the formal properties.

3.1 Simple Expansions to more expressive DLs

We want to show three ways to expand CSM to more expressive DLs. The first one is an expansion directly from *simi*. As mentioned in the recap, *simi* is purposely designed to fulfil Subsumption Preserving. So this first expansion will be purposely designed to fulfil Subsumption Preserving for conjunctions. As dual case for this, the second expansion will revert this design and so purposely fulfil Reverse Subsumption Preserving for the disjunction. As third expansion, we show that it is also possible to drop the design with the inner asymmetric CSM, to get an on its own fully symmetric CSM.

Before we come to the expansion, we want to present definition and handling of the *primitive negation* $(\neg)$. As we already pointed out, $(\neg)$ is a lighter version of the negation. In particular, $(\neg)$ is a negation, that only occurs in front of primitive concept names. In all other aspects it is the same as the normal known negation.

Definition 23 ($N_{C_{pr}}$)**.** *For an acyclic TBox $\mathcal{T}$, $N_{C_{pr}}$ is the* set of all primitive concept names.

Definition 24 (primitive negation)**.** *The* primitive negation $(\neg)$ *is a negation, that can only be applied to primitive concept names $A \in N_{C_{pr}}$.*

Definition 25 ($N_{L_{pr}}$)**.** *We define the set of all* primitive literals *as follows:*

$$N_{L_{pr}} = N_{C_{pr}} \cup \{\neg A | A \in N_{C_{pr}}\}$$

This definitions could also be done for concept names of N_C. But since we assumed all concept description to just contain primitive concept names, we want to use the more elemental definitions.
Handling the primitive negation $(\neg)$ is rather easy, since we just can expand *pm*

to the set of all primitive literals. We will call such an expansion a *literal measure*. Since we will later introduce different literal measures, when referring to a literal measure in general, we want to use lm. So we to not have do introduce a rule for $(\neg)$. All rules for the behaviour between primitive literals can be covered in the literal measure.
For some $\mathcal{DL}$ with $\neg \in const(\mathcal{DL})$, we can also introduce a normal form, that needs negations to be in front of concept names. This allows us to simulate $\neg$ by $(\neg)$. Unfortunately it also requires, that for every other constructor, the corresponding dual constructor has to be in $const(\mathcal{DL})$ too. So if $\exists \in const(\mathcal{DL})$ also $\forall$ has to, if $\sqcup \in const(\mathcal{DL})$ also $\sqcap$ has to and the other ways around. An even bigger disadvantage is, that if $\sqcap, \sqcup \in const(\mathcal{DL})$ the complexity of the transformation form arbitrary concept description into such a normal is known to be exponential in the length of the concept description. So an investigation of $\neg$ will still bring benefit.

3.1.1 Expansions using a asymmetric Sub-Measure $simi_{dir}$

We want to give two simple ways to build CSMs that use an inner asymmetric sub-measure, like *simi* does. For the first one, we define a not necessarily symmetric literal measure.

Definition 26 (literal measure)**.** *A function* $lm^{asym} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$ *is a* literal measure, *if for all* $A, B \in N_{L_{pr}}$ *and* $r, s, t \in N_R$ *holds:*

- $lm^{asym}(A, B) = 1 \iff A = B$
- $A \equiv \neg B \Longrightarrow lm^{asym}(A, B) = 0$
- $lm^{asym}(r, s) = 1 \iff s \sqsubseteq_{\mathcal{R}} r$,
- $s \sqsubseteq_{\mathcal{R}} r \Longrightarrow lm^{asym}(s, r) > 0$ *and*
- $t \sqsubseteq_{\mathcal{R}} s \Longrightarrow lm^{asym}(r, s) \leq lm^{asym}(r, t)$

Note that lm^{asym} can be symmetric for $s \sqsubseteq_{\mathcal{R}} r$ holds $lm^{asym}(r, s) = lm^{asym}(s, r) = 1$. This is the case if $\mathcal{R}$ is empty, since so subsumption can only be achieved by equivalence. In other cases where $s \sqsubseteq_{\mathcal{R}} r$ without them being equivalent, a CSMs that actually handling role names, will lose Equivalence Closed. So those cases should be avoided.
This definition expands pm just by the second property. This property will later be important to assure Dissimilar Closed for some of the CSM. As for pm, we also define a default literal measure.

Definition 27 (default literal measure)**.** *We define the* default literal measure lm_{def}^{asym} :

$N^2_{L_{pr}} \cup N^2_R \longrightarrow [0,1]$, *for* $\rho \in (0,1]$ *as follows*

$$lm^{asym}_{def}(A,B) := \begin{cases} 0 & \textit{if } A \neq B \\ 1 & \textit{if } A = B \end{cases}$$

$$lm^{asym}_{def}(r,s) := \begin{cases} 1 & \textit{if } r = s \textit{ or } s \sqsubseteq r \\ 0 & \textit{if } s \not\sqsubseteq r \textit{ and } r \not\sqsubseteq s \\ \rho & \textit{if } r \sqsubseteq s \textit{ and } s \not\sqsubseteq r \end{cases}$$

$r, s \in N_R$ *and* $A, B \in N_{L_{pr}}$

Note that if ρ is *1*, lm^{asym}_{def} is symmetric and the observations of before hold. Also ρ can not be *0*, or the fourth property of Definition 26 would not hold. With lm^{asym}, we can define the CSM $simi_{asym}$. Since for $\sqcup$ and $\forall$ we just adapted the calculations from $\sqcap$ and $\exists$, this is just a simple expansion of *simi*.

Definition 28 ($simi_{asym}$)**.** *For a specific description logic* $\mathcal{DL}$*,* $C, D, E, F \in \mathcal{C}(\mathcal{DL})$*,* $\otimes$ *being a fuzzy connector,* $\oplus$ *being a t-conorm,* lm^{asym} *a literal measure, and g a weighting function, the concept similarity measure* $simi_{asym} : \mathcal{C}(\mathcal{DL})^2 \longrightarrow [0,1]$ *is defined as follows:*

$$simi_{asym}(C,D) = simi_{dir}(C,D) \otimes simi_{dir}(D,C)$$

with:

$$simi_{dir}(C,D) = \begin{cases} \dfrac{\sum_{C' \in \widehat{C}_\sqcap}[g(C') * \bigoplus_{D' \in \widehat{D}_\sqcap} simi_{dir}(C',D')]}{\sum_{C' \in \widehat{C}_\sqcap} g(C')} & \textit{if } C \neq \top \textit{ and } |\widehat{C}_\sqcap| > 1 \textit{ or } |\widehat{D}_\sqcap| > 1 \\ \dfrac{\sum_{C' \in \widehat{C}_\sqcup}[g(C') * \bigoplus_{D' \in \widehat{D}_\sqcup} simi_{dir}(C',D')]}{\sum_{C' \in \widehat{C}_\sqcup} g(C')} & \textit{if } C \neq \top \textit{ and } |\widehat{C}_\sqcup| > 1 \textit{ or } |\widehat{D}_\sqcup| > 1 \\ 1 & \textit{if } C = D = \top \textit{ or } C = D = \bot \\ lm^{asym}(r,s)[w + (1-w) simi_{dir}(E,F)] & \textit{if } C = \exists r.E \textit{ and } D = \exists s.F \textit{ or} \\ & C = \forall r.E \textit{ and } D = \forall s.F \\ lm^{asym}(A,B) & A, B \in N_{L_{pr}} \\ 0 & \textit{otherwise} \end{cases}$$

and a suitable $w > 0$.

This CSM covers the constructors $(\neg), \sqcap, \sqcup, \exists$ and $\forall$. So it can be used for every $\mathcal{DL}$ with $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap, \sqcup, \exists, \forall\}$ and even can use role hierarchies form an Rbox $\mathcal{R}$. As *simi*, $simi_{asym}$ is purposely designed to fulfil Subsumption Preserving for $\sqcap$. Unfortunately for $\sqcup$ Subsumption Preserving does not hold, because of the semantics of $\sqcup$ within DLs. We will go into more details, when we proof the formal properties for the constructors in 3.2.
Also $simi_{asym}$ is defined for different combinations of its components. So also here these components are formally specified in the order:

$simi_{asym}$[fuzzy connector, t-conorm, literal measure, weighting function, ω]

and also here we want to drop this formality, to get shorter formulas.
As $simi_{asym}$ is purposely designed to fulfil Subsumption Preserving for $\sqcap$, we can do the same with Reverse Subsumption Preserving and $\sqcup$. Therefore we first define a dual literal measure, that is the same as lm^{asym}, but the asymmetric properties for the role names are turned around.

Definition 29 (dual literal measure). *A function $lm^{dual} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$ is a* dual literal measure*, if for all $A, B \in N_{L_{pr}}$ and $r, s, t \in N_R$ holds:*

- $lm^{dual}(A,B) = 1 \iff A = B$
- $A \equiv \neg B \implies lm^{dual}(A,B) = 0$
- $lm^{dual}(r,s) = 1 \iff s \sqsupseteq_{\mathcal{R}} r$,
- $s \sqsupseteq_{\mathcal{R}} r \implies lm^{dual}(s,r) > 0$ *and*
- $r \sqsupseteq_{\mathcal{R}} t \implies lm^{dual}(s,r) \geq lm^{asym}(s,t)$

We can also receive the default measure by swapping around the asymmetric conditions for the role names.

Definition 30 (default dual literal measure). *We define the* default dual literal measure $lm_{def}^{dual} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$, *for $\rho \in (0,1]$ as follows*

$$lm_{def}^{dual}(A,B) := \begin{cases} 0 & \text{if } A \neq B \\ 1 & \text{if } A = B \end{cases}$$

$$lm_{def}^{dual}(r,s) := \begin{cases} 1 & \text{if } r = s \text{ or } s \sqsupseteq r \\ 0 & \text{if } s \not\sqsupseteq r \text{ and } r \not\sqsupseteq s \\ \rho & \text{if } r \sqsupseteq s \text{ and } s \not\sqsupseteq r \end{cases}$$

$r, s \in N_R$ *and* $A, B \in N_{L_{pr}}$

Definition 31 ($simi_{dual}$). *For a specific description logic $\mathcal{DL}$, $C, D, E, F \in \mathcal{C}()\mathcal{DL}$, $\otimes$ being a fuzzy connector, $\oplus$ being a t-conorm, lm^{dual} a dual literal measure, and g a weighting function $simi_{dual} : \mathcal{C}(\mathcal{DL})^2 \longrightarrow [0,1]$ is defined as follows:*

$$simi_{dual}(C,D) = simi_{dir^*}(C,D) \otimes simi_{dir^*}(D,C)$$

with:

$$simi_{dir^*}(C,D) = \begin{cases} \dfrac{\sum_{C' \in \widehat{C}_\sqcap}[g(C') * \bigoplus_{D' \in \widehat{D}_\sqcap} simi_{dir^*}(C',D')]}{\sum_{C' \in \widehat{C}_\sqcap} g(C')} & \text{if } C \neq \top \text{ and } |\widehat{C}_\sqcap| > 1 \text{ or } |\widehat{D}_\sqcap| > 1 \\ \dfrac{\sum_{C' \in \widehat{C}_\sqcup}[g(C') * \bigoplus_{D' \in \widehat{D}_\sqcup} simi_{dir^*}(C',D')]}{\sum_{C' \in \widehat{C}_\sqcup} g(C')} & \text{if } C \neq \top \text{ and } |\widehat{C}_\sqcup| > 1 \text{ or } |\widehat{D}_\sqcup| > 1 \\ 1 & \text{if } C = D = \top \text{ or } C = D = \bot \\ lm^{dual}(r,s)[w + (1-w)simi(E,F)] & \text{if } C = \exists r.E \text{ and } D = \exists s.F \text{ or} \\ & C = \forall r.E \text{ and } D = \forall s.F \\ lm^{dual}(A,B) & A, B \in N_{L_{pr}} \\ 0 & \text{otherwise} \end{cases}$$

and a suitable $w > 0$.

The CSM itself differs from $simi_{asym}$ just in the use of a different literal measure. So the introduces $simi_{dir^*}$ is just a $simi_{dir}$, that uses lm^{dual} instead of lm^{asym} and $simi_{dual}$ a $simi_{asym}$ that uses $simi_{dir^*}$ instead of $simi_{dir}$.
Here again the order of the formal specification of the components is:

$$simi_{dual}[\text{fuzzy connector, t-conorm, literal measure, weighting function, } \omega]$$

which we want to drop for shorter formulas.

3.1.2 A Fully symmetric Expansion

We also want to introduce a method for those CSMs that not uses an inner asymmetric CSM.

Definition 32 (symmetric literal measure). *A function $lm^{sym} : N^2_{L_{pr}}$ is a* symmetric literal measure, *if for all $A, B \in N_{L_{pr}}$ and $r, s, t \in N_R$ holds:*

- $lm^{sym}(A, B) = 1 \iff A = B$,
- $lm^{sym}(A, B) = lm^{sym}(B, A)$ *and* $lm^{sym}(r, s) = lm^{sym}(s, r)$
- $A \equiv \neg B \Longrightarrow lm^{sym}(A, B) = 0$
- $lm^{sym}(r, s) = 1 \iff s = r$,
- $s \sqsubseteq_{\mathcal{R}} r$ *or* $r \sqsubseteq_{\mathcal{R}} s \Longrightarrow lm^{sym(s,r)} > 0$ *(bounded)*

We can also define a default measure for lm^{sym}.

Definition 33 (symmetric default literal measure). *We define the* symmetric default literal measure $lm^{sym}_{def} : N^2_{L_{pr}} \cup N^2_R \longrightarrow [0, 1]$, *for $\rho \in (0, 1)$ as follows:*

$$lm^{sym}_{def}(A, B) := \begin{cases} 0 & \text{if } A \neq B \\ 1 & \text{if } A = B \end{cases}$$

$$lm^{sym}_{def}(r, s) := \begin{cases} 1 & \text{if } r = s \\ \rho & \text{if } s \sqsubseteq_{\mathcal{R}} r \text{ or } r \sqsubseteq_{\mathcal{R}} s \\ 0 & \text{otherwise} \end{cases}$$

To handle conjunctions and disjunctions, we define a symmetric operator, that adapts $\bigoplus_{D' \in \widehat{D}_{\sqcap}} simi_d(C', D')$ in a symmetric way. First of all, the recursive call of $simi_d$ should be the fully symmetric CSM. Therefore the new operator has to know some CSM $simi_{xx}$. Because we want to keep the property from [4], to be a generalisation of the Jaccard Index, for every first level participant that occurs in C and D, the operator should return *1*. All other first level participants should be recursively compared with the first level participant from the other concept description.

Definition 34 (*handle*$_{C,D}^{\bigcirc,simi_{xx}}$)**.** *Let* $C, D \in \mathcal{C}(\mathcal{DL})$, $E' \in \widehat{C}_{\bigcirc} \cup \widehat{D}_{\bigcirc}$, $\bigcirc$ *be a operator of* $\mathcal{DL}$ *and* $\oplus$ *be a t-conorm. We define* $handle_{C,D}^{\bigcirc,simi_{xx}}(E') : \mathcal{C}(\mathcal{DL}) \longrightarrow [0,1]$ *for an specific* $simi_{xx}$ *as follows:*

$$handle_{C,D}^{\bigcirc,simi_{xx}}(E') = \begin{cases} 1 & E' \in \widehat{C}_{\bigcirc} \cap \widehat{D}_{\bigcirc} \\ \bigoplus_{D' \in \widehat{D}_{\bigcirc}} simi_{xx}(E', D') & E' \in \widehat{C}_{\bigcirc} \backslash \widehat{D}_{\bigcirc} \\ \bigoplus_{C' \in \widehat{C}_{\bigcirc}} simi_{xx}(E', C') & E' \in \widehat{D}_{\bigcirc} \backslash \widehat{C}_{\bigcirc} \\ 0 & otherwise \end{cases}$$

Since this operator and lm^{sym} are symmetric, we can easily define a fully symmetric CSM $simi_{sym}$.

Definition 35 ($simi_{sym}$)**.** *For a specific description logic* $\mathcal{DL}$, $C, D \in \mathcal{C}(\mathcal{DL})$, lm^{asym} *a symmetric literal measure and g a weighting function and* $handle_{C,D}^{\bigcirc,simi_{xx}}(E')$ *using a t-conorm* $\oplus$, $simi_{sym} : \mathcal{C}(DL)^2 \longrightarrow [0,1]$ *is defined as follows:*

$$simi_{sym}(C,D) = \begin{cases} \dfrac{\sum_{E' \in \widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap}} [g(E') * handle_{C,D}^{\sqcap,simi_{sym}}(E')]}{\sum_{E' \in \widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap}} g(E')} & if\ |\widehat{C}_{\sqcap}| > 1\ or\ |\widehat{D}_{\sqcap}| > 1 \\ \dfrac{\sum_{E' \in \widehat{C}_{\sqcup} \cup \widehat{D}_{\sqcup}} [g(E') * handle_{C,D}^{\sqcup,simi_{sym}}(E')]}{\sum_{E' \in \widehat{C}_{\sqcup} \cup \widehat{D}_{\sqcup}} g(E')} & if\ |\widehat{C}_{\sqcup}| > 1\ or\ |\widehat{D}_{\sqcup}| > 1 \\ lm^{sym}(r,s)[w + (1-w)simi_{sym}(E,F)] & if\ C = \exists r.E\ and\ D = \exists s.F\ or \\ & C = \forall r.E\ and\ D = \forall s.F \\ 1 & if\ C = D = \top\ or\ C = D = \bot \\ lm^{sym}(A,B) & A, B \in N_{L_{pr}} \\ 0 & otherwise \end{cases}$$

and a suitable $w > 0$.

Since $simi_{sym}$ is symmetric by its own, there is no fuzzy connector needed. The formal specification of its components is as follows:

$simi_{sym}$[t-conorm, literal measure, weighting function, ω]

Like before, for the rest of this paper we want to drop this specification for shorter formulas.

3.1.3 Generalisation of a CSM Structure

Since the rules for the constructor will be essential to the fulfilling of the formal properties, we want to introduce a general quotation for those rules. This will allow us to analyse these rule, without doing vague references. The general structure of this kind of CSM is:

$$sim(C,D) = \begin{cases} connect_{\bigcirc}^{collect_{\bigcirc}(C,D)}(choose_{\bigcirc}) & \text{for } \bigcirc \in \{\sqcap, \sqcup\} \\ connect_{\bigcirc}(choose_{\bigcirc}^{R}, choose_{\bigcirc}^{C}) & \text{for } \bigcirc \in \{\forall, \exists\} \\ \text{fintely many special values} & \text{for finitely many special cases} \\ \text{default value} & \textit{otherwise} \end{cases}$$

So for the conjunction $\sqcap$ and the disjunction $\sqcup$, we have a $collect_{\bigcirc}(C,D)$ function, that builds out of C and D a set of concept descriptions. The $choose_{\bigcirc}$ function calculates for every concept description given to it a value between *0* and *1*. The $connect_{\bigcirc}^{collect_{\bigcirc}(C,D)}$ function takes the concepts of $collect_{\bigcirc}(C,D)$, forwards them to $choose_{\bigcirc}$ and calculates from the returns of $choose_{\bigcirc}$ the final similarity value. For shorter formulas, we also want to refer these $connect_{\bigcirc}^{collect_{\bigcirc}(C,D)}$ as $connect_{\sqcap}$ or $connect_{\sqcup}$.
For the $\exists$-quantification and $\forall$-quantification we have a $choose_{\bigcirc}^{R}$ function, which calculates a similarity between the role names and a $choose_{\bigcirc}^{C}$ that calculates a similarity between the concept descriptions the quantifications are put on. Note that in $choose_{\bigcirc}^{C}$ the ^{C} is no reference to the input concept C. It just declares, that this is the *choose*-function for the concept description. The function $connect_{\forall}$ or respectively $connect_{\exists}$ connects the returns of the both functions to a final similarity value.
The *finitely many special cases* will cover the handling of the constants $\top$ and $\bot$ and other special cases, that may occur. As usual the *otherwise* cover all cases, where none of the former rules can be applied to, with a default value. The specifications of the functions are:

$$\begin{aligned} choose_{\sqcup}/choose_{\sqcap} &: \mathcal{DL} \longrightarrow [0,1] \\ choose_{\forall}^{R}/choose_{\exists}^{R} &: N_R \longrightarrow [0,1] \\ choose_{\forall}^{C}/choose_{\exists}^{C} &: \mathcal{DL} \longrightarrow [0,1] \\ collect_{\bigcirc} &: \mathcal{DL}^2 \longrightarrow \mathcal{P}(\mathcal{DL}) \\ connect_{\bigcirc}^{collect_{\bigcirc}(C,D)} &: [0,1]^{|collect_{\bigcirc}(C,D)|} \longrightarrow [0,1] \\ connect_{\forall}/connect_{\exists} &: [0,1]^2 \longrightarrow [0,1] \end{aligned}$$

Note that for future work, a rule for the negation $\neg$ can be generalized in the same way. But since we do not cover the negation $\neg$ in this paper, we leave it out.

Example 1. *For $simi_{sym}$ we have:*

$$\begin{aligned}
connect_{\bigcirc}^{collect_{\bigcirc}(C,D)} &= mean_{[0,1]}^{g}(\{choose_{\bigcirc}(E') | E' \in collect_{\sqcup}(C,D)\}) \\
collect_{\sqcup}(C,D) &= \widehat{C}_{\sqcup} \cup \widehat{D}_{\sqcup} \\
collect_{\sqcap}(C,D) &= \widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap} \\
choose_{\sqcup}(E') &= handle_{C,D}^{\sqcup,simi_{sym}}(E') \\
choose_{\sqcap}(E') &= handle_{C,D}^{\sqcap,simi_{sym}}(E')
\end{aligned}$$

where $mean_{[0,1]}^{g}$ is the mean of with g weighted values between 0 and 1

$$\begin{aligned}
connect_{\bigcirc} &= choose_{\bigcirc}^{R}(r,s)[w + (1-w)choose_{\bigcirc}^{C}(E,F)] \\
choose_{\forall}^{R}(r,s) = choose_{\exists}^{R}(r,s) &= lm^{sym}(r,s) \\
choose_{\forall}^{C}(E,F) = choose_{\exists}^{C}(E,F) &= simi_{sym}(E,F)
\end{aligned}$$

special cases:

$$\begin{aligned}
C = D = \top \text{ or } C = D = \bot &\longrightarrow 1 \\
A, B \in N_{L_{pr}} &\longrightarrow lm^{sym}(A,B)
\end{aligned}$$

default value : 0

Example 1 gives us an idea, how this generalisation work. We can do it the same way for $simi_d$, $simi_{dir}$ and $simi_{dir^*}$. For *simi*, $simi_{asym}$ and $simi_{dual}$ we would additionally need a function, that connects the inner asymmetric CSMs. But since this will always be the fuzzy connector, we do not introduce a generalized name for it.

3.2 The Formal Properties and their Preservation by the Constructors

We now take a look on the formal properties and check for the literal measures and every constructor, under which conditions they preserve this property. The idea is, that if we have a $\mathcal{DL}$ and $const(\mathcal{DL})$, we can do an inductive proof over the structure of $\mathcal{DL}$. So we define

Definition 36. *Let $\mathcal{DL}$ be a specific description logic and $S \subseteq \mathcal{C}(\mathcal{DL})$ be a set of concept description, what fulfils the property $\circ^{prob}$ for a CSM sim on $\mathcal{DL}$. A constructor $\bigcirc \in const(\mathcal{DL})$* preserves $\circ^{prob}$ *for sim on $\mathcal{DL}$, if for all possible chosen S and $C_1, \cdots, C_n \in S$ holds:*

$$S' = \{\bigcirc(C_1, \cdots, C_n)\} \cup S \text{ fulfils } \circ^{prob}$$

this means, if we have a set S of concept description, that fulfil a property, and a constructor, that preserves this same property, we can expand S with the constructor and the resulting set S' will not violate the property. As induction base serves the literal measure. If the constructor preserves the property, we can do the induction step. Note that it can be possible, that a constructor preserves a property just under certain conditions. Then we have to check these conditions, to do the induction step.
For shorter formulas, we want to assume all role name relations to be with respect to $\mathcal{R}$. So we write $r \sqsubseteq s$ and $r \equiv s$ instead of $r \sqsubseteq_{\mathcal{R}} s$ and $r \equiv_{\mathcal{R}} s$. Also, most of the properties can be proofed easier for $simi_{sym}$, so we usually start with this expansion first.

3.2.1 Equivalence Invariant and Equivalence Closed

Since Equivalence Invariant can easily be achieved by a unique normal form, we will not investigate the relations between Equivalence Closed and the constructors here.

Theorem 1. *For every CSM sim is Equivalence Invariant if it uses a unique normal form.*

This theorem obviously is true, because with a unique normal form, two concept description that are equivalent have the same syntactic appearance. So they are semantically and syntactically equal. This means, calculating the similarity to a third concept description with the same deterministic CSM, will take the same steps and so return in both cases the same results.
Note that Equivalence Invariant can also be achieved in some other way. For $N_{L_{pr}}$ and N_R this is done by our assumption, that our normal forms are at least name unique. The rules of the constructors must be as interchangeable as they can be equivalently transformed into each other. For example, for $\neg C \sqcap \neg D \equiv \neg(C \sqcup D)$ must hold that calculation of the negation applied on the disjunction isomorph to the conjunction applied on the negations is. Also for distributive transformations like $\exists r.C \sqcup \exists r.D \equiv \exists r.(C \sqcup D)$ regarded. Especially the transformation $\exists r.C \sqcap \forall r.(C \sqcap D) \equiv \exists r.(C \sqcap D) \sqcap \forall r.(C \sqcap D)$ could make problems with this approach. Also note, that our restrictions for TBoxes to be unfoldable and that for equivalent role names, one is chosen for the syntax, we achieved that trivial violations of Equivalence Invariant are excluded. So it is not to possible violate this property just because we have two equivalent role or concept names, with different similarities to other role or concept names.

Equivalence Closed holds trivially for literal measure. Also the role names are no concept descriptions, we will still want the literal measures lm to be Equivalent Closed in N_R. To our advantage, this also holds trivially.
For the rest of the paper we want explicitly differ between a literal measures lm to fulfil a property within $N_{L_{pr}}$ and a literal measures lm to fulfil a property within N_R.

Lemma 1. *Every literal measure lm^{sym}, lm^{asym} or lm^{dual} is Equivalence Closed in $N_{L_{pr}}$. Every symmetric literal measure lm^{sym} is also Equivalence Closed in N_R.*

Proof 1 (Lemma 1)**.**

From the first property of Definitions 26, 29 and 32 follows immediately, that every both are Equivalence Closed in $N_{L_{pr}}$.
Additionally the rules of Definition 32 ensures, that lm^{sym} is also Equivalence Closed in N_R.

□

We can prove, that for $simi_{sym}$ every constructor preserves Equivalence Closed.

Theorem 2. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$. We can say for the constructors of $\mathcal{DL}$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ and disjunction $\sqcup$ preserve Equivalence Closed.*
2) *The $\exists$- and $\forall$-quantification preserve Equivalence Closed.*

Proof 2 (Theorem 2)**.**

1) Let $C = \bigsqcap_i C_i$ and $D = \bigsqcap_j D_j$. We have to show, that for

$$simi_{sym}(C_i, D_j) = 1 \Longleftrightarrow C_i \equiv D_j$$

holds:

$$simi_{sym}(C, D) = 1 \Longleftrightarrow C \equiv D$$

Let $simi_{sym}(C, D) = 1$. Since :

$$collect_{\sqcap}(C, D) = \widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap}$$

and for $n = |collect_{\sqcap}(C, D)|$ holds:

$$connect_{\sqcap}^{collect_{\bigcirc}(C,D)}(x_1, ..., x_n) = 1 \Longleftrightarrow x_1 = ... = x_n = 1$$

we have:

$$\begin{aligned}
&\forall E' \in collect_{\sqcap}(C, D) : choose_{\sqcap}(E') = 1 \\
&\Longleftrightarrow \forall E' \in collect_{\sqcap}(C, D) : E' \in \widehat{C}_{\sqcap} \cap \widehat{D}_{\sqcap} \\
&\Longleftrightarrow (\widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap}) \subseteq (\widehat{C}_{\sqcap} \cap \widehat{D}_{\sqcap}) \\
&\Longleftrightarrow (\widehat{C}_{\sqcap} \cup \widehat{D}_{\sqcap}) = (\widehat{C}_{\sqcap} \cap \widehat{D}_{\sqcap}) \\
&\Longleftrightarrow \widehat{C}_{\sqcap} = \widehat{D}_{\sqcap} \\
&\Longleftrightarrow C \equiv D
\end{aligned}$$

For the disjunction the argumentation is the same.

2) Let $C = \forall r.C'$ and $D = \forall s.D'$. We have to show, that for

$$simi_{sym}(C', D') = 1 \iff C' \equiv D'$$

and

$$lm^{sym}(r, s) = 1 \iff r = s$$

hold:

$$simi_{sym}(C, D) = 1 \iff C \equiv D$$

Let $simi_{sym}(C, D) = 1$. Since:

$$choose^{R}_{\forall}(r, s) = lm^{sym}(r, s)$$

$$choose^{C}_{\exists}(E, F) = simi_{sym}(E, F)$$

and it holds that:

$$connect_{\forall}(x_1, x_2) = 1 \iff x_1 = x_2 = 1$$

we have:

$$\begin{aligned} & choose^{R}_{\forall}(r, s) = choose^{C}_{\exists}(C', D') = 1 \\ \iff & lm^{sym}(r, s) = simi_{sym}(C', D') = 1 \\ \iff & r = s, C' \equiv D' \\ \iff & C \equiv D \end{aligned}$$

The argumentation for the $\exists$-quantification is the same.

□

Due to their asymmetric structure, the proof for $simi_{asym}$ is not so obvious. The following lemma states two properties for $simi_{dir}$, that will help us with the proof.

Lemma 2. *Let $simi_{dir}$ be on a specific $\mathcal{DL}$. For the constructors of $\mathcal{DL}$ and the property:*

$$simi_{dir}(C, D) = 1 \iff C \sqsupseteq D$$

we can say, that for $simi_{dir}$:

1) The literal measure lm^{asym} fulfils this property

2) The conjunction $\sqcap$ preserves this property

3) The $\exists$- and $\forall$-quantification preserve this property

Also it holds that:

4) the disjunction $\sqcup$ preserves at least:

$$C \equiv D \Longrightarrow simi_{dir}(C, D) = 1$$

The first property is taken from [4]. We just adapted it for our expansion. In fact, it would have been enough to show, that all those constructors prevent the second property, to proof Equivalence closed for $simi_{asym}$. But since the first one implies the second one and we will need the first one for later proofs, we will prove it right now.

Proof 3 (Lemma 2)**.**

1) Since on the level of literals, subsumption can just be achieved by equivalence, this follows directly from the Definition 26.

2) Let $C = \bigsqcap_i C_i$ and $D = \bigsqcap_j D_j$. We can argue:

$$\begin{aligned} simi_{dir}(C, D) = 1 &\Longleftrightarrow \forall C_i : \exists D_j : simi_{dir}(C_i, D_j) = 1 \\ &\Longleftrightarrow \forall C_i : \exists D_j : C_i \sqsupseteq D_j \\ &\Longleftrightarrow C \sqsupseteq D. \end{aligned}$$

3) Let $C = \forall r.C'$ and $D = \forall s.D'$. Since this property by definition applies to $lm^{asym}(r, s)$, we can argue:

$$\begin{aligned} simi_{dir}(C, D) = 1 &\Longleftrightarrow pm^{asym}(r, s) = 1 \land simi_{dir}(C', D') = 1 \\ &\Longleftrightarrow r \sqsupseteq s \land C' \sqsupseteq D' \\ &\Longleftrightarrow C \sqsupseteq D. \end{aligned}$$

The argumentation for the $\exists$-quantification is the same.

4) Let $C = \bigsqcup_i C_i$ and $D = \bigsqcup_j D_j$. We can argue:

$$\begin{aligned} C \equiv D &\Longrightarrow \forall C_i \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : C_i = D_j \\ &\Longleftrightarrow \forall C_i \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : simi_{dir}(C_i, D_j) = 1 \\ &\Longleftrightarrow simi_{dir}(C, D) = 1. \end{aligned}$$

□

This property allows our next theorem about Equivalence Closed and the constructors handled by $simi_{asym}$.

Theorem 3. *Let $simi_{asym}$ be on a specific $\mathcal{DL}$ and use the fuzzy connector $\otimes$. We can say for the constructors of $\mathcal{DL}$, that for $simi_{asym}$:*

1) The conjunction $\sqcap$, the $\exists$- and $\forall$-quantification preserve Equivalence Closed.

2) The disjunction $\sqcup$ preserves Equivalence Closed.

Proof 4 (Theorem 3)**.**

1) Since for $simi_{dir}$ all these constructors preserve the first property from Lemma 2, we can argue:

$$\begin{aligned}
simi_{asym}(C,D) = 1 &\iff simi_{dir}(C,D) \otimes simi_{dir}(D,C) = 1 \\
&\iff simi_{dir}(C,D) = 1 \wedge simi_{dir}(D,C) = 1 \\
&\iff C \sqsupseteq D \wedge D \sqsupseteq C \\
&\iff C \equiv D.
\end{aligned}$$

2) Because of a unique normal form we can argue:

$$\begin{aligned}
simi_{asym}(C,D) = 1 &\iff simi_{dir}(C,D) \otimes simi_{dir}(D,C) = 1 \\
&\iff simi_{dir}(C,D) = 1 \wedge simi_{dir}(D,C) = 1 \\
&\iff \forall C_i \in \widehat{C}_{\sqcup} : \exists D_j \in \widehat{D}_{\sqcup} : simi_{dir}(C_i, D_j) = 1 \\
&\qquad \wedge \forall D_o \in \widehat{D}_{\sqcup} : \exists C_p \in \widehat{C}_{\sqcup} : simi_{dir}(D_o, C_p) = 1 \\
&\iff \forall C_i \in \widehat{C}_{\sqcup} : \exists D_j \in \widehat{D}_{\sqcup} : C_i \equiv D_j \\
&\qquad \wedge \forall D_o \in \widehat{D}_{\sqcup} : \exists C_p \in \widehat{C}_{\sqcup} : D_o \equiv C_p \\
&\iff C \equiv D.
\end{aligned}$$

Since $simi_{dual}$ is nearly the same as $simi_{asym}$, we can adapt the proof, to show Equivalence Closed for $simi_{dual}$. For this adaption we have to remember, the only difference between $simi_{asym}$ and $simi_{dual}$ is the used literal measure. These again differ just in the direction of their role name related properties. This means this time, that $simi_{dir^*}(\exists r.C, \exists s.D) = 1$ if $C \sqsubseteq D$ and $r \sqsubseteq s$ (same for $\forall$). And instead for $\sqcap$, the adapted property holds for $\sqcup$.

Corollary 1. *We can adapt Proof 3 to show that lm^{dual} fulfils and for $simi_{dir^*}$ the constructors $\sqcup, \exists$ and $\forall$ preserve the following property:*

$$simi_{dir^*}(C,D) = 1 \iff C \sqsubseteq D$$

Furthermore we can adapt Proof 4 to show, that for $simi_{dual}$ the constructors $\sqcup, \sqcap, \exists$ and $\forall$ preserve Equivalence Closed.

3.2.2 Subsumption Preserving and Reverse Subsumption Preserving

The measures lm^{sym}, lm^{asym} and lm^{dual} are trivially Subsumption Preserving in $N_{L_{pr}}$. Same holds for Reverse Subsumption Preserving. To preserve the properties also for the quantifications, we have to set some requirements to the relations of the role names. While for lm^{sym} the definitions of Subsumption Preserving and Reverse Subsumption Preserving can easily be taken on to N_R, lm^{asym} and lm^{dual} need some precise definitions.

Definition 37. *We define lm^{dual} to be Subsumption Preserving in N_R if for all $r, s, t \in N_R$ holds:*

$$s \sqsupseteq r \Longrightarrow lm^{dual}(s,t) \leq lm^{dual}(r,t)$$
$$s \sqsupseteq r \Longrightarrow lm^{dual}(t,s) \leq lm^{dual}(t,s)$$

Since in $simi_{asym}$ the quantifications preserve Subsumption Preserving due to its special design, we do not need lm^{asym} to fulfil additional requirements. For Reverse Subsumption Preserving it will be the other way around. Here lm^{dual} will not need to fulfil additional requirements.

Definition 38. *We define lm^{asym} to be Reverse Subsumption Preserving in N_R if for all $r, s, t \in N_R$ holds:*

$$s \sqsupseteq r \Longrightarrow lm^{asym}(s,t) \geq lm^{asym}(r,t)$$
$$s \sqsupseteq r \Longrightarrow lm^{asym}(t,s) \geq lm^{asym}(t,s)$$

The definitions for lm^{sym} to be Subsumption Preserving or Reverse Subsumption Preserving in N_R stay as usual, since it is fully symmetric. With the first of these definitions we formulate the following lemma for Subsumption Preserving.

Lemma 3. *Every literal measure lm^{sym}, lm^{asym} or lm^{dual} is Subsumption Preserving in $N_{L_{pr}}$. Also lm^{sym} and lm^{dual} can be constructed to fulfil Subsumption Preserving in N_R. We denote those by adding the index $_{SP}$: lm^{sym}_{SP}, lm^{dual}_{SP}.*

Proof 5 (Lemma 3)**.**

Since on the level of concept names subsumption can only be achieved by equivalence, we have for every $A_1 \sqsubseteq A_2 \sqsubseteq A_3, A_i \in N_{L_{pr}}$ that $lm^{asym}(A_1, A_2) = lm^{asym}(A_1, A_3) = 1$. So lm^{asym} Subsumption Preserving is for literals.
The argumentations for lm^{sym} and lm^{dual} are the same.
Since non of the properties of lm^{sym} and lm^{dual} contradict Subsumption Preserving in N_R, both can be constructed to fulfil this property.

□

This proof can also be used for Reverse Subsumption Preserving.

Corollary 2. *Every literal measure lm^{sym},lm^{asym} or lm^{dual} is Reverse Subsumption Preserving in $N_{L_{pr}}$. Also lm^{sym} and lm^{asym} can can be constructed to fulfil Reverse Subsumption Preserving in N_R. We denote those by adding $\cdot_{RSP}$: lm^{sym}_{RSP}, lm^{asym}_{RSP}*

For the in 3.1 defined lm^{sym}_{def} we have, that if $s \sqsupseteq r$, every $t \sqsubseteq r$ is also subsumed by s. So lm^{sym}_{def} gives r and t the same similarity as s and t, except $r = t$ or $s = r = t$. Same argumentation can be done for $t' \sqsupseteq s$. So lm^{sym}_{def} by definition fulfils both properties.

Corollary 3. *Every lm^{sym}_{def} Subsumption Preserving and Reverse Subsumption Preserving in N_R.*

The next theorem shows under which conditions the constructors preserve Subsumption Preserving for $simi_{sym}$.

Theorem 4. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$.We can say for the constructors of $\mathcal{DL}$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ not generally preserves Subsumption Preserving.*
2) *The disjunction $\sqcup$ not generally preserves Subsumption Preserving.*
3) *The $\exists$- and $\forall$-quantification not generally preserve Subsumption Preserving.*

Furthermore holds:

4) *The conjunction $\sqcap$ preserves Subsumption Preserving, if for all possible participants C_i of the conjunctions additionally holds:*
$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$
5) *The disjunction $\sqcup$ preserves Subsumption Preserving, if for all possible participants C_i of the disjunctions additionally holds:*
$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$
and
$$simi_{sym}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$
6) *The $\exists$- and $\forall$-quantification preserves Subsumption Preserving, for lm_{SP}^{sym}.*

Note that *4)* and *5)* take advantage of lm^{sym} being Equivalence Closed. Because in both cases, we want subsumptions automatically be equivalences and lm^{sym} is Equivalence Closed, this results in every pair of each other subsuming participants having an similarity value of *1*. Note also, that a more general restriction for *5)* is, that just equivalences between the possible participants of the disjunctions return *1* and every other pair *0*. Essentially this is the same, except that this also allows subsumption beside equivalences. Therefore it also has no possible knowledge about concept description that subsume each other. Anyway, both cases break the calculation down to the Jaccard Index [4].

Proof 6 (Theorem 4)**.**

1) Let $E = E_1, D = D_1$ and $C = C_1 \sqcap C_2$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. By condition $simi_{sym}(C_1, D_1) \geq simi_{sym}(C_1, E_1)$. Since there are no further restrictions, it can be possible that
$$simi_{sym}(C_2, E_1) > 2 * simi_{sym}(C_2, D_1)$$
and
$$simi_{sym}(C_2, D_1) > simi_{sym}(C_1, D_1)$$
So we have:
$$simi_{sym}(C, E) > \frac{2}{3} * simi_{sym}(C_2, E_1) > \frac{4}{3} * simi_{sym}(C_2, D_1) > simi_{sym}(C, D)$$
Thus the conjunction $\sqcap$ not generally preserves Subsumption Preserving.

2) Let $E = E_1 \sqcup E_2, D = D_1$ and $C = C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. By assumption $simi_{sym}(C_1, D_1) \geq simi_{sym}(C_1, E_1)$. Since there are no further restrictions, it can be possible that:

$$simi_{sym}(C_1, E_2) > 2 * simi_{sym}(C_1, D_1)$$

and

$$simi_{sym}(C_1, E_1) = simi_{sym}(C_1, D_1)$$

So we have:

$$simi_{sym}(C, E) > \frac{2}{3} * simi_{sym}(C_1, E_2) > \frac{4}{3} * simi_{sym}(C_1, D_1) > simi_{sym}(C, D)$$

Thus the disjunction $\sqcup$ not generally preserves Subsumption Preserving.

3) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. By assumption $simi_{sym}(C_1, D_1) \geq simi_{sym}(C_1, E_1)$. Since there are no further restrictions, it can be possible that:

$$\begin{aligned}
&lm^{sym}(r_C, r_E) > lm^{sym}(r_C, r_D) \frac{\omega + (1-\omega) simi_{sym}(C_1, D_1)}{\omega + (1-\omega) simi_{sym}(C_1, E_1)} \\
\Longrightarrow\ & lm^{sym}(r_C, r_E) * [\omega + (1-\omega) simi_{sym}(C_1, E_1)] \\
&\quad > lm^{sym}(r_C, r_D) * [\omega + (1-\omega) simi_{sym}(C_1, D_1) \\
\Longrightarrow\ & simi_{sym}(C, E) > simi_{sym}(C, D)]
\end{aligned}$$

Thus the $\forall$-quantification not generally preserves Subsumption Preserving. The argumentation for the $\exists$-quantification is the same.

4) Let $E = \bigsqcap_i E_i, D = \bigsqcap_j D_j$ and $C = \bigsqcap_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a conjunction this means:

$$\begin{aligned}
D \sqsupseteq C &\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \sqsupseteq C_k \\
&\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \equiv C_k \\
&\Longrightarrow \widehat{D}_\sqcap \subseteq \widehat{C}_\sqcap
\end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. Since so clearly $\widehat{E}_\sqcap \subseteq \widehat{D}_\sqcap \subseteq \widehat{C}_\sqcap$ and $connect_\sqcap^{collect_\sqcap(C,D)}$ is monotone, we can argue:

$$\begin{aligned}
&\forall E_i \in \widehat{E}_\sqcap : \exists D_j \in \widehat{D}_\sqcap : E_i \equiv D_j \\
\Longleftrightarrow\ & \forall C_k \in \widehat{C}_\sqcap, \forall E_i \in \widehat{E}_\sqcap : \exists D_j \in \widehat{D}_\sqcap : simi_{sym}(C_k, E_i) = simi_{sym}(C_k, D_j) \\
\Longrightarrow\ & \forall C_k \in \widehat{C}_\sqcap, \forall E_i \in \widehat{E}_\sqcap : \exists D_j \in \widehat{D}_\sqcap : simi_{sym}(C_k, E_i) \leq simi_{sym}(C_k, D_j) \\
\Longleftrightarrow\ & simi_{sym}(C, E) \leq simi_{sym}(C, D)
\end{aligned}$$

So Subsumption Preserving is fulfilled.

5) Let $E = \bigsqcup_i E_i, D = \bigsqcup_j D_j$ and $C = \bigsqcup_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a disjunction this means:

$$\begin{aligned} D \sqsupseteq C &\iff \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \sqsupseteq C_k \\ &\iff \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \equiv C_k \\ &\implies \widehat{C}_\sqcup \subseteq \widehat{D}_\sqcup \end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. By the additional assumptions and $\widehat{C}_\sqcup \sqsubset \widehat{D}_\sqcup \sqsubseteq \widehat{E}_\sqcup$ we have:

$$simi_{sym}(C, E) = \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{C}_\sqcup \cup \widehat{E}_\sqcup|} = \frac{|\widehat{C}_\sqcup|}{|\widehat{E}_\sqcup|} \leq \frac{|\widehat{C}_\sqcup|}{|\widehat{D}_\sqcup|} = \frac{|\widehat{C}_\sqcup \cap \widehat{D}_\sqcup|}{|\widehat{C}_\sqcup \cup \widehat{D}_\sqcup|} = simi_{sym}(C, D)$$

So Subsumption Preserving is fulfilled.

6) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. According to the additional assumption along with $simi_{sym}(C_1, D_1) \geq simi_{sym}(C_1, E_1)$ also $lm^{sym}_{SP}(r_C, r_D) \geq lm^{sym}_{SP}(r_C, r_E)$ holds. Since $connect_\forall$ is monotone, this directly implies that $simi_{sym}(C, D) \geq simi_{sym}(C, E)$. So Subsumption Preserving is fulfilled.
The argumentation for the $\exists$-quantification is the same.

$\square$

So for $simi_{sym}$ the conjunction $\sqcap$ and disjunction $\sqcup$ just preserve Subsumption Preserving for special conditions. We can synthesize these conditions with lm^{sym}_{def} and limitations of $const(\mathcal{DL})$.

Corollary 4. *For $\mathcal{DL}$, $simi_{sym}$ and Subsumption Preserving we have:*

1) *The conjunction $\sqcap$ preserves Subsumption Preserving for lm^{sym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$*

2) *The disjunction $\sqcup$ preserves Subsumption Preserving for lm^{sym}_{def} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$*

Before we proof Subsumption Preserving for $simi_{asym}$, we introduce another lemma with a property for $simi_{dir}$. This lemma is again taken from [4] and expanded to lm^{asym} and our additional constructors.

Lemma 4. *Let $simi_{dir}$ be on a specific $\mathcal{DL}$. For $const(\mathcal{DL})$ and the property:*

$$E \sqsupseteq D \implies \forall C : simi_{dir}(C, E) \leq simi_{dir}(C, D)$$

we can say that for $simi_{dir}$:

1) *The literal measure lm^{asym} fulfils this property.*

2) *The conjunction $\sqcap$, the $\exists$- and $\forall$-quantification preserve this property.*

This property is the cause, $simi_{asym}$ preserves Subsumption Preserving for most constructors. In particular, the asymmetric properties for role names in lm^{asym} make the quantifications fulfil this property for $simi_{dir}$. Because for the conjunction holds, that $\sqcap_i D_i \sqsupseteq \sqcap_j C_j$ if and only if for every D_i there exists a C_j that is subsumed by D_i (see [4]), it also preserves this property. For the disjunction this is not possible.

Proof 7 (Lemma 4)**.**

1) Since on the level of literals, subsumption can only be achieved by equivalence, lm^{asym} fulfils this property.

2) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_C.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. By assumption, this property follows directly the fourth property of Definition 26.
Let $E = \sqcap_i E_i, D = \sqcap_j D_j$ and $C = \sqcap_k C_k$. Since this is a conjunction, we have:

$$\begin{aligned} E \sqsupseteq D &\iff \forall E_i \in \widehat{E}_\sqcap : \exists D_J \in \widehat{D}_\sqcap : E_i \sqsupseteq D_j \\ &\iff \forall C_K \in \widehat{C}_\sqcap, \forall E_i \in \widehat{E}_\sqcap : \exists D_J \in \widehat{D}_\sqcap : simi_{dir}(C_k, E_i) \leq simi_{dir}(C_k, D_j) \\ &\iff simi_{dir}(C, E) \leq simi_{dir}(C, D) \end{aligned}$$

The argumentation for the $\exists$-quantification is the same.

$\square$

This property effects, that for the constructors $(\neg), \sqcap, \exists$ and $\forall$ and $C \sqsubseteq D \sqsubseteq E$, one direction of $simi_{dir}$ is *1*. For the quantifications the last rule of Definition 26 guarantees that for the other direction Subsumption Preserving holds. Since in conjunctions, the subsuming term are always shorter or equal to the subsumed ones, for conjunctions the other directions also are Subsumption Preserving. So with *1* being the neutral element for the t-norm, the Subsumption Preserving preserving values will be returned for $simi_{asym}$. Here our limitation of the fuzzy connectors to bounded t-norm takes effect. We need the neutral element *1* for the fuzzy connector, to get Subsumption Preserving for $simi_{asym}$. But note that also other fuzzy connectors with the neutral element *1* will fulfil this purpose.

Theorem 5. *Let $simi_{asym}$ be on a specific $\mathcal{DL}$.We can say for $const(\mathcal{DL})$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$, the $\exists$- and $\forall$-quantification preserve Subsumption Preserving.*

2) *The disjunction $\sqcup$ not generally preserves Subsumption Preserving.*

Furthermore holds:

3) The disjunction $\sqcup$ preserves Subsumption Preserving, if for all possible participants C_i of the disjunction additionally holds:

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

and

$$simi_{dir}(C_1, C_2) = 0 \wedge simi_{dir}(C_2, C_1) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

Note that here *3)* takes advantage of lm_{asym} being Equivalence Closed. Also here, only equivalence returning *1* and everything else *0*, could be more general condition.

Proof 8 (Theorem 5)**.**

1) With Lemma 2 and the identity property of $\otimes$ holds:

$$\begin{aligned} simi_{asym}(C, D) &= simi_{dir}(C, D) \otimes simi_{dir}(D, C) \\ &= simi_{dir}(C, D) \otimes 1 \\ &= simi_{dir}(C, D) \end{aligned}$$

The same holds for $simi_{asym}(C, E)$. By Lemma 4 and $E \sqsupseteq D$, it follows that:

$$simi_{asym}(C, E) = simi_{dir}(C, E) \leq simi_{dir}(C, D) = simi_{asym}(C, D)$$

So Subsumption Preserving is fulfilled.

2) Let $E = E_1 \sqcup E_2, D = D_1$ and $C = C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. Since there are no further restrictions, it can be possible that:

$$\begin{aligned} simi_{dir}(C_1, D_1) &= simi_{dir}(C_1, E_1) \\ &= simi_{dir}(C_1, E_2) \\ &= 1 \\ simi_{dir}(E_2, C_1) &> 2 * simi_{dir}(D_1, C_1) \end{aligned}$$

and

$$simi_{dir}(E_1, C_1) = simi_{dir}(D_1, C_1)$$

So we have:

$$\begin{aligned} simi_{dir}(E, C) > \frac{1}{2} simi_{dir}(E_2, C_1) > simi_{dir}(D_1, C_1) = simi_{dir}(D, C) \\ simi_{dir}(C, E) = \ 1 = simi_{dir}(C, D) \end{aligned}$$

what, along with 1 being the neutral element of $\otimes$, implies:

$$simi_{asym}(C, E) = simi_{dir}(E, C) > simi_{dir}(D, C) = simi_{asym}(C, D)$$

3) Let $E = \bigsqcup_i E_i, D = \bigsqcup_j D_j$ and $C = \bigsqcup_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a disjunction this means:

$$D \sqsupseteq C \Longleftrightarrow \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \sqsupseteq C_k$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. So we have $|\widehat{C}_\sqcup| \leq |\widehat{D}_\sqcup| \leq |\widehat{E}_\sqcup|$ Because of the additional assumption and $simi_{dir}$ being Equivalence Closed, holds:

$$simi_{dir}(C, E) = \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{C}_\sqcup|} = \frac{|\widehat{C}_\sqcup|}{|\widehat{C}_\sqcup|} = \frac{|\widehat{C}_\sqcup \cap \widehat{D}_\sqcup|}{|\widehat{C}_\sqcup|} = simi_{dir}(C, D)$$

$$simi_{dir}(E, C) = \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{E}_\sqcup|} = \frac{|\widehat{C}_\sqcup|}{|\widehat{E}_\sqcup|} \leq \frac{|\widehat{C}_\sqcup|}{|\widehat{D}_\sqcup|} = \frac{|\widehat{C}_\sqcup \cap \widehat{D}_\sqcup|}{|\widehat{D}_\sqcup|} = simi_{dir}(E, D)$$

From 1 being the neutral element for $\otimes$, follows

$$simi_{asym}(C, E) \leq simi_{asym}(C, D)$$

So Subsumption Preserving is fulfilled.

$\square$

With a limitation of $const(\mathcal{DL})$ and lm^{sym}_{def}, the requirements for *3)* can be implemented.

Corollary 5. *For $\mathcal{DL}$, $simi_{asym}$ and Subsumption Preserving, the disjunction $\sqcup$ preserves Subsumption Preserving for lm^{sym}_{def} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$.*

For $simi_{sym}$ and Reverse Subsumption Preserving the theorem is pretty much the same as for Subsumption Preserving. Just the requirements for $\sqcap$ and $\sqcup$ to fulfil it are exchanged. So we can say, that for Reverse Subsumption Preserving the disjunction is the one, that is more likely to preserve it.

Theorem 6. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$.We can say for $const(\mathcal{DL})$, that for $simi_{sym}$:*

1) The conjunction $\sqcap$ not generally preserves Reverse Subsumption Preserving.

2) The disjunction $\sqcup$ not generally preserves Reverse Subsumption Preserving.

3) The $\exists$- and $\forall$-quantification not generally preserve Reverse Subsumption Preserving.

Furthermore holds:

4) The conjunction $\sqcap$ preserves Reverse Subsumption Preserving, if for all possible participants C_i of the conjunction additionally holds:

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

and

$$simi_{sym}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

5) *The disjunction $\sqcup$ preserves Reverse Subsumption Preserving, if for all possible participants C_i of the disjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

6) *The $\exists$- and $\forall$-quantification preserves Reverse Subsumption Preserving, for lm^{sym}_{RSP}.*

Again *4)* and *5)* take advantage of lm^{sym} being Equivalence closed. This time, for *4)* a more general condition could be, that only equivalence returns *1* and everything else *0*.

Proof 9 (Theorem 6)**.**

1) Let $E = E_1, D = D_1$ and $C = C_1 \sqcap C_2$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. By condition $simi_{sym}(C_1, E_1) \leq simi_{sym}(D_1, E_1)$. Since there are no further restrictions, it can be possible that

$$simi_{sym}(C_2, E_1) > 2 * simi_{sym}(D_1, E_1)$$

and

$$simi_{sym}(E_1, D_1) = simi_{sym}(E_1, C_1)$$

So we have:

$$simi_{sym}(C, E) > \frac{2}{3} * simi_{sym}(C_2, E_1) > \frac{4}{3} * simi_{sym}(D_1, E_1) > simi_{sym}(D, E)$$

Thus the conjunction $\sqcap$ not generally preserves Reverse Subsumption Preserving.

2) Let $E = E_1 \sqcup E_2, D = D_1$ and $C = C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. By assumption $simi_{sym}(C_1, E_1) \leq simi_{sym}(D_1, E_1)$. Since there are no further restrictions, it can be possible that:

$$simi_{sym}(C_1, E_2) > 2 * simi_{sym}(D_1, E_1)$$

and

$$simi_{sym}(D_1, E_2) > simi_{sym}(D_1, E_1)$$

So we have:

$$simi_{sym}(C, E) > \frac{2}{3} * simi_{sym}(C_1, E_2) > \frac{4}{3} * simi_{sym}(D_1, E_1) > simi_{sym}(D, E)$$

Thus the disjunction $\sqcup$ not generally preserves Reverse Subsumption Preserving.

3) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. By assumption $simi_{sym}(D_1, E_1) \geq simi_{sym}(C_1, E_1)$. Since

there are no further restrictions, it can be possible that:

$$\begin{aligned}
&lm^{sym}(r_C, r_E) > lm^{sym}(r_D, r_E)\frac{\omega + (1-\omega)simi_{sym}(D_1, E_1)}{\omega + (1-\omega)simi_{sym}(C_1, E_1)} \\
\Longrightarrow\ & lm^{sym}(r_C, r_E)[\omega + (1-\omega)simi_{sym}(C_1, E_1)] \\
& > lm^{sym}(r_C, r_D)[\omega + (1-\omega)simi_{sym}(D_1, E_1)] \\
\Longrightarrow\ & simi_{sym}(C, E) > simi_{sym}(D, E)
\end{aligned}$$

Thus the $\forall$-quantification not generally preserves Reverse Subsumption Preserving.
The argumentation for the $\exists$-quantification is the same.

4) Let $E = \bigsqcap_i E_i, D = \bigsqcap_j D_j$ and $C = \bigsqcap_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a conjunction this means:

$$\begin{aligned}
D \sqsupseteq C &\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \sqsupseteq C_k \\
&\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \equiv C_k \\
&\Longrightarrow \widehat{D}_\sqcap \subseteq \widehat{C}_\sqcap
\end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. By the additional assumption and $\widehat{C}_\sqcup \sqsupseteq \widehat{D}_\sqcup \sqsupseteq \widehat{E}_\sqcup$ we have:

$$simi_{sym}(C, E) = \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{C}_\sqcup \cup \widehat{E}_\sqcup|} = \frac{|\widehat{E}_\sqcup|}{|\widehat{C}_\sqcup|} \leq \frac{|\widehat{E}_\sqcup|}{|\widehat{D}_\sqcup|} = \frac{|\widehat{D}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{D}_\sqcup \cup \widehat{E}_\sqcup|} = simi_{sym}(D, E)$$

So Reverse Subsumption Preserving is fulfilled.

5) Let $E = \bigsqcup_i E_i, D = \bigsqcup_j D_j$ and $C = \bigsqcup_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a disjunction this means:

$$\begin{aligned}
D \sqsupseteq C &\Longleftrightarrow \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \sqsupseteq C_k \\
&\Longleftrightarrow \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \equiv C_k \\
&\Longrightarrow \widehat{C}_\sqcup \subseteq \widehat{D}_\sqcup
\end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. Since so clearly $\widehat{E}_\sqcup \supseteq \widehat{D}_\sqcup \supseteq \widehat{C}_\sqcup$ and $connect_\sqcup^{collect_\sqcup}$ is monotone, we can argue:

$$\begin{aligned}
& \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : C_k \equiv D_j \\
\Longleftrightarrow & \forall E_i \in \widehat{E}_\sqcup, \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : simi_{sym}(C_k, E_i) = simi_{sym}(D_j, E_i) \\
\Longrightarrow & \forall E_i \in \widehat{E}_\sqcup, \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : simi_{sym}(C_k, E_i) \leq simi_{sym}(D_j, E_i) \\
\Longleftrightarrow & simi_{sym}(C, E) \leq simi_{sym}(D, E)
\end{aligned}$$

So Reverse Subsumption Preserving is fulfilled.

6) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. According to the additional assumption, along with $simi_{sym}(D_1, E_1) \geq simi_{sym}(C_1, E_1)$ also $lm^{sym}_{RSP}(r_D, r_E) \geq lm^{sym}_{RSP}(r_C, r_E)$ holds. Since $connect_\forall$ is monotone, this directly implies that $simi_{sym}(D, E) \geq simi_{sym}(C, E)$. So Reverse Subsumption Preserving is fulfilled.
The argumentation for the $\exists$-quantification is the same.

$\square$

Like for Subsumption Preserving, we can synthesise the requirements for $\sqcap$ and $\sqcup$ to fulfil Reverse Subsumption Preserving by limiting $const(\mathcal{DL})$ and lm^{sym}_{def}.

Corollary 6. *For $\mathcal{DL}$, $simi_{sym}$ and Reverse Subsumption Preserving we have:*

1) *The conjunction $\sqcap$ preserves Reverse Subsumption Preserving for lm^{sym}_{def} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$*

2) *The disjunction $\sqcup$ preserves Reverse Subsumption Preserving for lm^{sym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$*

For Reverse Subsumption Preserving $simi_{asym}$ has no advantage towards $simi_{sym}$.

Theorem 7. *Let $simi_{asym}$ be on a specific $\mathcal{DL}$.We can say for $const(\mathcal{DL})$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ not generally preserves Reverse Subsumption Preserving.*

2) *The disjunction $\sqcup$ not generally preserves Reverse Subsumption Preserving.*

3) *The $\exists$- and $\forall$-quantification not generally preserve Reverse Subsumption Preserving.*

Furthermore holds:

4) *The conjunction $\sqcap$ preserves Reverse Subsumption Preserving, if for all possible participants C_i of the conjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

and

$$simi_{dir}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

5) *The disjunction $\sqcup$ preserves Reverse Subsumption Preserving, if for all possible participants C_i of the disjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

6) *The $\exists$- and $\forall$-quantification preserve Reverse Subsumption Preserving, for lm^{asym}_{RSP}.*

Also here *4)* and *5)* take advantage of lm^{asym} being Equivalence Closed and again for *4)* a more general condition could be, that only equivalence returns *1* and everything else *0*.

Proof 10 (Theorem 7)**.**

1) Let $E = E_1, D = D_1$ and $C = C_1 \sqcap C_2$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. Since there are no further restrictions, it can be possible that

$$\begin{aligned}
simi_{dir}(E_1, D_1) &= simi_{dir}(E_1, C_1)\\
&= simi_{dir}(E_2, C_1)\\
&= 1\\
simi_{dir}(C_2, E_1) &> 2 * simi_{dir}(D_1, E_1)
\end{aligned}$$

and

$$simi_{dir}(C_2, E_1) = simi_{dir}(C_1, E_1)$$

So we have:

$$\begin{aligned}
simi_{dir}(C, E) > \frac{1}{2} * simi_{dir}(C_2, E_1) = simi_{dir}(D_1, E_1) &> simi_{dir}(D, E)\\
simi_{dir}(E, C) = \ 1 = simi_{dir}(E, D)
\end{aligned}$$

what, along with 1 being the neutral element of $\otimes$, implies:

$$simi_{asym}(C, E) = simi_{dir}(C, E) > simi_{dir}(D, E) = simi_{asym}(D, E)$$

2) Let $E = E_1 \sqcup E_2, D = D_1$ and $C = C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$. Since there are no further restrictions, it can be possible that:

$$\begin{aligned}
simi_{dir}(D_1, E_1) &= simi_{dir}(C_1, E_1)\\
&= simi_{dir}(C_1, E_2)\\
&= 1\\
simi_{dir}(E_2, C_1) &> 2 * simi_{dir}(E_1, D_1)
\end{aligned}$$

and

$$simi_{dir}(E_1, D_1) = simi_{dir}(E_1, C_1)$$

So we have:

$$\begin{aligned}
simi_{dir}(E, C) > \frac{1}{2} simi_{dir}(E_2, C_1) > simi_{dir}(D_1, C_1) = simi_{dir}(D, C)\\
simi_{dir}(C, E) = \ 1 = simi_{dir}(D, E)
\end{aligned}$$

what, along with 1 being the neutral element of $\otimes$, implies:

$$simi_{asym}(C, E) = simi_{dir}(E, C) > simi_{dir}(D, C) = simi_{asym}(C, D)$$

3) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. By assumption $simi_{dir}(D_1, E_1) \geq simi_{dir}(C_1, E_1)$. By the second property of Definition 26 we have $lm^{asym}dir(r_E, r_C) = lm^{asym}(r_E, r_D) = 1$.

Since there are no further restrictions, it can be possible that $sim_{dir}(E, C) = sim_{dir}(E, D) = 1$, which implies by identity of $\otimes$ with 1, that:

$$\begin{aligned} simi_{asym}(E, D) &= simi_{dir}(E, D) \otimes simi_{dir}(D, E) \\ &= 1 \otimes simi_{dir}(D, E) \\ &= simi_{dir}(D, E) \end{aligned}$$

The same can be shown for $simi_{asym}(E, C)$. Additionally it is possible that:

$$\begin{aligned} & lm^{asym}(r_C, r_E) > lm^{asym}(r_D, r_E) \frac{\omega + (1-\omega) simi_{dir}(D_1, E_1)}{\omega + (1-\omega) simi_{dir}(C_1, E_1)} \\ \Longrightarrow\ & lm^{asym}(r_C, r_E)[\omega + (1-\omega) simi_{dir}(C_1, E_1)] \\ & \quad > lm^{asym}(r_D, r_E)[\omega + (1-\omega) simi_{dir}(D_1, E_1)] \\ \Longrightarrow\ & simi_{dir}(C, E) > simi_{dir}(D, E) \end{aligned}$$

Thus the $\forall$-quantification not generally preserves Reverse Subsumption Preserving.
The argumentation for the $\exists$-quantification is the same.

4) Let $E = \bigsqcap_i E_i, D = \bigsqcap_j D_j$ and $C = \bigsqcap_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a conjunction this means:

$$\begin{aligned} D \sqsupseteq C &\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \sqsupseteq C_k \\ &\Longleftrightarrow \forall D_j \in \widehat{D}_\sqcap : \exists C_k \in \widehat{C}_\sqcap : D_j \equiv C_k \\ &\Longrightarrow \widehat{D}_\sqcap \subseteq \widehat{C}_\sqcap \end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. By the additional assumption and $\widehat{C}_\sqcup \sqsupseteq \widehat{D}_\sqcup \sqsupseteq \widehat{E}_\sqcup$ we have:

$$\begin{aligned} simi_{dir}(C, E) &= \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{C}_\sqcup|} = \frac{|\widehat{E}_\sqcup|}{|\widehat{C}_\sqcup|} \leq \frac{|\widehat{E}_\sqcup|}{|\widehat{D}_\sqcup|} = \frac{|\widehat{D}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{D}_\sqcup|} = simi_{dir}(D, E) \\ simi_{dir}(E, C) &= \frac{|\widehat{C}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{E}_\sqcup|} = \frac{|\widehat{E}_\sqcup|}{|\widehat{E}_\sqcup|} = \frac{|\widehat{D}_\sqcup \cap \widehat{E}_\sqcup|}{|\widehat{E}_\sqcup|} = simi_{dir}(E, D) \end{aligned}$$

Because $\otimes$ is monotone, also $simi_{asym}(C, E) \leq simi_{asym}(D, E)$. So Reverse Subsumption Preserving is fulfilled.

5) Let $E = \bigsqcup_i E_i, D = \bigsqcup_j D_j$ and $C = \bigsqcup_k C_k$ with $E \sqsupseteq D \sqsupseteq C$. Since this is a disjunction this means:

$$\begin{aligned} D \sqsupseteq C &\Longleftrightarrow \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \sqsupseteq C_k \\ &\Longleftrightarrow \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : D_j \equiv C_k \\ &\Longrightarrow \widehat{C}_\sqcup \subseteq \widehat{D}_\sqcup \end{aligned}$$

The same can be shown for $E \sqsupseteq C$ and $E \sqsupseteq D$. Since so clearly $\widehat{E}_\sqcup \supseteq \widehat{D}_\sqcup \supseteq \widehat{C}_\sqcup$ and $connect_\sqcup^{collect_\sqcup}$ is monotone, we can argue:

$$\begin{aligned}
&\forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : C_k \equiv D_j \\
\Longleftrightarrow &\forall E_i \in \widehat{E}_\sqcup, \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : simi_{dir}(C_k, E_i) = simi_{dir}(D_j, E_i) \\
\Longrightarrow &\forall E_i \in \widehat{E}_\sqcup, \forall C_k \in \widehat{C}_\sqcup : \exists D_j \in \widehat{D}_\sqcup : simi_{dir}(C_k, E_i) \leq simi_{dir}(D_j, E_i) \\
\Longleftrightarrow &simi_{dir}(C, E) \leq simi_{dir}(D, E)
\end{aligned}$$

The same way we can show that $simi_{dir}(E, C) \leq simi_{dir}(E, D)$. Together with the monotonicity of $\otimes 1$ follows $simi_{asym}(C, E) \leq simi_{asym}(D, E)$. So Reverse Subsumption Preserving is fulfilled.

6) Let $E = \forall r_E.E_1, D = \forall r_D.D_1$ and $C = \forall r_D.C_1$ with $E_1 \sqsupseteq D_1 \sqsupseteq C_1$ and $r_E \sqsupseteq r_D \sqsupseteq r_C$. This implies by Lemma 2, that

$$simi_{dir}(E_1, D_1) = simi_{dir}(E_1, C_1) = 1$$

According to the additional assumption holds

$$\begin{aligned}
simi_{dir}(D_1, E_1) \otimes simi_{dir}(E_1, D_1) &\geq simi_{dir}(C_1, E_1) \otimes simi_{dir}(E_1, C_1) \\
simi_{dir}(D_1, E_1) \otimes 1 &\geq simi_{dir}(C_1, E_1) \otimes 1 \\
simi_{dir}(D_1, E_1) &\geq simi_{dir}(C_1, E_1)
\end{aligned}$$

and

$$\begin{aligned}
lm_{RSP}^{asym}(r_D, r_E) &\geq lm_{RSP}^{asym}(r_C, r_E) \\
lm_{RSP}^{asym}(r_E, r_D) &= lm_{RSP}^{asym}(r_E, r_C) = 1
\end{aligned}$$

Since $\otimes$ and $connect_\forall$ are monotone, this implies that

$$simi_{asym}(D, E) \geq simi_{asym}(C, E)$$

So Reverse Subsumption Preserving is fulfilled.
The argumentation for the $\exists$-quantification is the same.

□

Corollary 7. *For $\mathcal{DL}$, $simi_{asym}$ and Reverse Subsumption Preserving we have:*

1) *The conjunction $\sqcap$ preserves Reverse Subsumption Preserving for lm_{def}^{asym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$*

2) *The disjunction $\sqcup$ preserves Reverse Subsumption Preserving for lm^{asym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$*

As mentioned, $simi_{dual}$ is designed dual to $simi_{asym}$. This means, that the asymmetric properties of the role names in lm^{asym} let the quantification term relate to each other in an inverted way. Because of this, they preserve Reverse Subsumption Preserving. Instead of the conjunction for $simi_{asym}$ the disjunction happens to preserve Reverse Subsumption Preserving as well. As drawback, $simi_{dual}$ has for Subsumption Preserving no advantages towards $simi_{sym}$.
The reverse design of $simi_{dual}$ starts with the additional property, that later can be used to proof that almost all constructors preserve Reverse Subsumption Preserving.

Corollary 8. *With the property from Corollary 1, we can adapt Proof 7 to show that $simi_{dir^*}$ fulfils and the constructor $\sqcup, \forall$ and $\exists$ preserve the property:*

$$E \sqsubseteq D \Longrightarrow \forall C : simi_{dir^*}(C, E) \leq simi_{dir^*}(C, D)$$

For Reverse Subsumption Preserving $simi_{dual}$ got the advantages, that $simi_{asym}$ got for Subsumption Preserving, except that the roles of the conjunction $\sqcap$ and the disjunction $\sqcup$ are interchanged.

Corollary 9. *With the property form Corollary 8, we can adapt Proof 8 to show that for $simi_{dual}$:*

1) *The disjunction $\sqcup$, the $\exists$- and $\forall$-quantification preserve Reverse Subsumption Preserving.*

2) *The conjunction $\sqcap$ not generally preserves Reverse Subsumption Preserving.*

and furthermore:

3) *The conjunction $\sqcap$ preserves Reverse Subsumption Preserving, if for all possible participants C_i of the conjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

and

$$simi_{dir^*}(C_1, C_2) = 0 \land simi_{dir^*}(C_2, C_1) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

Corollary 10. *For $\mathcal{DL}$, $simi_{dual}$ and Reverse Subsumption Preserving, the conjunction $\sqcap$ preserves Subsumption Preserving for lm^{dual}_{def} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$.*

For Subsumption Preserving $simi_{dual}$ got the disadvantages, that $simi_{asym}$ got for Reverse Subsumption Preserving, except that again the roles of the conjunction $\sqcap$ and the disjunction $\sqcup$ are interchanged.

Corollary 11. *We can adapt Proof 10 to show that for $simi_{dual}$:*

1) *The disjunction $\sqcup$ not generally preserves Subsumption Preserving.*

2) *The conjunction $\sqcap$ not generally preserves Subsumption Preserving.*

3) *The $\exists$- and $\forall$-quantification not generally preserve Subsumption Preserving.*

Furthermore holds:

4) *The disjunction $\sqcup$ preserves Subsumption Preserving, if for all possible participants C_i of the disjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

and

$$simi_{dir^*}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

5) *The conjunction $\sqcap$ preserves Subsumption Preserving, if for all possible participants C_i of the conjunction additionally holds:*

$$C_1 \sqsupseteq C_2 \Longleftrightarrow C_1 \equiv C_2$$

6) *The $\exists$- and $\forall$-quantification preserve Subsumption Preserving, for lm^{dual}_{SP}.*

Corollary 12. *For $\mathcal{DL}$, $simi_{dual}$ and Subsumption Preserving we have:*

1) *The disjunction $\sqcup$ preserves Subsumption Preserving for lm^{dual}_{def} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$*

2) *The conjunction $\sqcap$ preserves Subsumption Preserving for lm^{dual} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$*

So $simi_{asym}$ and $simi_{dual}$ have an advantage towards $simi_{sym}$ in the properties Subsumption Preserving or Reverse Subsumption Preserving. Because of their semantic, we also can say that the conjunction $\sqcap$ is more likely to preserve Subsumption Preserving and the disjunction $\sqcup$ is more likely to preserve Reverse Subsumption Preserving.

3.2.3 Dissimilar Closed and Bounded

For the formal properties Dissimilar Closed and Bounded we have to distinguish two cases. The first case is if $\sqcup \in const(\mathcal{DL})$. For this case we trivially have that $lcs(C, D) = C \sqcup D$. From this follows that:

$$lcs(C, D) = \top \Longleftrightarrow C \sqsupseteq \neg D$$

This is not only harder to check, but also brings major trouble for fulfilling $lcs(C, D) = \top$ with quantifications. In fact, with just the primitive negation $(\neg)$, it is impossible to produce $\top$ within a disjunction containing a quantification.
The second case is, that $\sqcup \notin const(\mathcal{DL})$. This was the case in the design of *simi*. For this holds, that

$$lcs(C, D) = \top \Longleftrightarrow C \not\sqsupseteq D \wedge D \not\sqsupseteq C$$

For literal and quantifications, this can easily be assured by a proper literal measure.
So we first define the conditions for the literal measure to be Dissimilar Closed.

Lemma 5. *Every literal measure lm^{sym},lm^{asym} or lm^{dual} is Dissimilar Closed for a specific $\mathcal{DL}$, if $\sqcup \in const(\mathcal{DL})$.*
If $\sqcup \notin const(\mathcal{DL})$, then lm^{sym} is still Dissimilar Closed, if for all $A, B \in N_{L_{pr}}$ and $r, s \in N_R$ holds:

$$A \not\equiv B \Longrightarrow lm^{sym}(A, B) = 0$$
$$r \not\sqsupseteq s \wedge s \not\sqsupseteq r \Longrightarrow lm^{sym}(r, s) = 0$$

and lm^{asym} and lm^{dual} are still Dissimilar Closed, if for all $A, B \in N_{L_{pr}}$ and $r, s \in N_R$ holds:

$$A \not\equiv B \Longrightarrow lm^*(A, B) \otimes lm^*(B, A) = 0$$
$$r \not\sqsupseteq s \wedge s \not\sqsupseteq r \Longrightarrow lm^*(r, s) = lm^*(s, r) = 0$$

with $lm^* \in \{lm^{asym}, lm^{dual}\}$. We want to denote those by adding $\cdot_{DiCl}$: lm^{sym}_{DiCl}, lm^{asym}_{DiCl}, lm^{dual}_{DiCl} .

The role name related rules are not needed to make the literal measures Dissimilar Closed. But we will later need these, to proof that the quantifications preserve Dissimilar Closed, so we include them right away.

Proof 11 (Lemma 5)**.**

If $\sqcup \in const(\mathcal{DL})$, then for $A, B \in N_{L_{pr}}$ the $lcs(A, B) = A \sqcup B$. So:

$$lcs(A, B) \equiv \top \Longleftrightarrow A \sqsupseteq \neg B$$

Because on the level of literals supsumtion can only be achieved by equivalence, this means:

$$lcs(A, B) \equiv \top \Longleftrightarrow A \equiv \neg B$$

By definition, this is 0. So Dissimilar Closed is fulfilled for this case.
If $\sqcup \notin const(\mathcal{DL})$:

$$\begin{aligned} lcs(A, B) \not\equiv \top &\Longleftrightarrow A \sqsupseteq B \\ &\Longleftrightarrow A \equiv B \end{aligned}$$

So lm^{sym} and lm^{asym} no longer fulfil Dissimilar Closed. But if they fulfil the additional assumption, they obviously fulfil Dissimilar Closed again.

□

Corollary 13. *Every lm^{sym}_{def}, lm^{asym}_{def} or lm^{dual}_{def} is Dissimilar Closed, even if $\sqcup \notin const(\mathcal{DL})$.*

The following theorem is about $simi_{sym}$ and Dissimilar Closed. It will refer to the cases of $\sqcup \notin const(\mathcal{DL})$ and $\sqcup \in const(\mathcal{DL})$, if the are of relevance.

Theorem 8. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$. We can say for const($\mathcal{DL}$), that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ preserves Dissimilar Closed.*

2) *The disjunction $\sqcup$ preserves Dissimilar Closed, if $\{\neg, (\neg)\} \cap const(\mathcal{DL}) = \varnothing$.*

3) *If $\sqcup \in const(\mathcal{DL})$, then the $\exists$- and $\forall$-quantification preserve Dissimilar Closed. If $\sqcup \notin const(\mathcal{DL})$, then the $\exists$- and $\forall$-quantification preserve Dissimilar Closed for lm^{sym}_{DiCl}.*

Proof 12 (Theorem 8)**.**

1) Let $D = \bigsqcap_i D_i$ and $C = \bigsqcap_j C_j$. If $\sqcup \notin const(\mathcal{DL})$, since

$$\begin{aligned} & connect_{\sqcap}^{collect_{\sqcap}(C,D)}(choose_{\sqcap}) = 0 \\ \iff & \forall E' \in collect_{\sqcap}(C,D) : choose_{\sqcap}(E') = 0 \end{aligned}$$

and there are no other n-ary constructors, we can argue:

$$\begin{aligned} lcs(C,D) \equiv \top & \iff \forall C_j \in \widehat{C}_{\sqcap}, \forall D_i \in \widehat{D}_{\sqcap} : C_j \not\sqsupseteq D_i \wedge D_i \not\sqsupseteq C_j \\ & \iff \forall C_j \in \widehat{C}_{\sqcap}, \forall D_i \in \widehat{D}_{\sqcap} : lcs(C_j, D_i) \equiv \top \\ & \implies \forall C_j \in \widehat{C}_{\sqcap}, \forall D_i \in \widehat{D}_{\sqcap} : simi_{sym}(C_j, D_i) = 0 \\ & \iff simi_{sym}(C,D) = 0 \end{aligned}$$

If $\sqcup \in const(\mathcal{DL})$:

$$\begin{aligned} lcs(C,D) \equiv \top & \iff C \sqsupseteq \neg D \\ & \iff \bigsqcap_j C_j \sqsupseteq \neg \bigsqcap_i D_i \\ & \iff \bigsqcap_j C_j \sqsupseteq \bigsqcup_i \neg D_i \\ & \iff \forall D_i \in \widehat{D}_{\sqcap} : \bigsqcap_j C_j \sqsupseteq \neg D_i \\ & \iff \forall D_i \in \widehat{D}_{\sqcap}, \forall C_j \in \widehat{C}_{\sqcap} : C_j \sqsupseteq \neg D_i \\ & \iff \forall D_i \in \widehat{D}_{\sqcap}, \forall C_j \in \widehat{C}_{\sqcap} : lcs(C_j, D_i) \equiv \top \\ & \implies \forall D_i \in \widehat{D}_{\sqcap}, \forall C_j \in \widehat{C}_{\sqcap} : simi_{sym}(C_j, D_i) = 0 \\ & \iff simi_{sym}(C,D) = 0 \end{aligned}$$

So Dissimilar Closed is preserved in both cases.

2) Let $D = \bigsqcup_i D_i$ and $C = \bigsqcup_j C_j$. We can argue:

$$\begin{aligned} lcs(C,D) \equiv \top & \iff C \sqsupseteq \neg D \\ & \iff \bigsqcup_j C_j \sqsupseteq \neg \bigsqcup_i D_i \\ & \iff \bigsqcup_j C_j \sqsupseteq \bigsqcap_i \neg D_i \\ & \implies \exists S \subseteq \widehat{C}_{\sqcap}, \exists M \subseteq \widehat{D}_{\sqcap},\ S, M \neq \varnothing : \forall D_m \in M, \forall C_s \in S : \bigsqcup_s C_s \sqsupseteq \bigsqcap_m \neg D_m \end{aligned}$$

So it is possible. that between $C_j \in \widehat{C}_\sqcap \backslash S$ and $D_i \in \widehat{D}_\sqcap \backslash M$ a similarity bigger than 0 is. Since:

$$\begin{aligned} &connect_{\sqcup}^{collect_\sqcup(C,D)}(choose_\sqcup) = 0 \\ \Longleftrightarrow\ &\forall E' \in collect_\sqcup(C,D) : choose_\sqcup(E') = 0 \end{aligned}$$

this means, that $\sqcup$ not generally preserves Dissimilar Closed. But if $\{\neg, (\neg)\} \cap const(\mathcal{DL}) = \varnothing$, the sets S and M are also empty. So the premiss is never fulfilled and therefore Dissimilar Closed trivially preserved.

3) Let $C = \forall r_C.C_1$. If D is a conjunction or disjunction, the rules for those should be applied first. If is not of the form $\forall r_D.D_1$, the result is 0 and so Dissimilar Closed trivially preserved.
Let $D = \forall r_D.D_1$. If $\sqcup \notin const(\mathcal{DL})$, since:

$$connect_\forall(choose_\forall^R, choose_\forall^C) = 0 \Longleftrightarrow choose_\forall^R = 0$$

we can argue:

$$\begin{aligned} lcs(C,D) \equiv \top &\Longleftrightarrow r_C \not\sqsupseteq r_D \wedge r_D \not\sqsupseteq r_C \\ &\Longleftrightarrow lm_{DiCl}^{sym}(r_C, r_D) = 0 \\ &\Longleftrightarrow simi_{sym}(C,D) = 0 \end{aligned}$$

If $\sqcup \in const(\mathcal{DL})$, $lcs(C,D) = (\forall r_C.C_1) \sqcup (\forall r_D.D_1) \not\equiv \top$, so the premiss is never fulfilled and therefore Dissimilar Closed trivially preserved.
The argumentation for the $\exists$-quantification is the same.

□

The only times the relation of the role names are relevant for this proof, we use the requirements for lm^{sym} being Dissimilar Closed. This allows us to adapt the hole proof easily to verify the same statement for $simi_{asym}$ and $simi_{dual}$.

Corollary 14. *Since the argumentations of Proof 12 can be adapted to either $simi_{dir}(C,D)$ or $simi_{dir}(D,C)$ and 0 is absorbing for the fuzzy connector $\otimes$, we can use the propositions of Theorem 8 also for $simi_{asym}$.*
The same holds for $simi_{dir^*}$ and $simi_{dual}$.

Note that for this corollary it is necessary, that the fuzzy connector has *0* as the absorbing element. Our limitation of the fuzzy connectors to bounded t-norm covers this.
In the next lemma we define the conditions for a literal measure to be Bound.

Lemma 6. *Every literal measure lm^{sym}, lm^{asym} or lm^{dual} is Bounded, if $\sqcup \notin const(\mathcal{DL})$. If $\sqcup \in const(\mathcal{DL})$, lm^{sym} is still Bounded if for $A, B \in N_{L_{pr}}$ holds:*

$$A \not\equiv \neg B \Longrightarrow lm^{sym}(A,B) > 0$$

and lm^{asym} and lm^{asym} are still Bounded if for $A, B \in N_{L_{pr}}$ holds:

$$A \not\equiv \neg B \Longrightarrow lm^*(A,B) \otimes lm^*(B,A) > 0$$

with $lm^* \in \{lm^{asym}, lm^{dual}\}$. We want to denote those literal measure by adding $\cdot_{Bou}$: lm^{sym}_{Bou},lm^{asym}_{Bou},lm^{dual}_{Bou}.

Proof 13 (Lemma 6)**.**

If $\sqcup \notin const(\mathcal{DL})$ we can argue :

$$\begin{aligned} lcs(A,B) \not\equiv \top &\Longleftrightarrow A \equiv B \\ &\Longleftrightarrow lm^{sym}(A,B) = 1 > 0 \end{aligned}$$

So Bounded is fulfilled.
If $\sqcup \in const(\mathcal{DL})$, then it is possible that $lm^{sym}(A,B) = 0$ for $A \not\equiv B$, but $lcs(A,B) = A \sqcup B$ is not necessarily equal to $\top$. But with the additional assumption that for all $A, B \in N_{L_{pr}}$: $lm^{sym}_{Bou}(A,B) > 0$, we have that the conclusion of Bounded always holds. So it is fulfilled for this literal measure.
The argumentations for lm^{asym} and lm^{dual} are the same.

□

For Bounded the trivial case is $\sqcup \notin const(\mathcal{DL})$. The other case brings a major trouble. If $\sqcup \notin const(\mathcal{DL})$, then the lcs between a quantification and an other concept is never equal to $\top$. So the premise of Bounded is always fulfilled, if a quantification participates. Unfortunately for those cases the similarity value is not always *0*.

Theorem 9. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$. We can say for the constructors of $\mathcal{DL}$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ preserves Bounded.*
2) *The disjunction $\sqcup$ preserves Bounded.*
3) *The $\exists$- and $\forall$-quantification preserve Bounded, if $\sqcup \notin const(\mathcal{DL})$.*

Proof 14 (Theorem 9)**.**

1) Let $D = \bigsqcap_i D_i$ and $C = \bigsqcap_j C_j$. If $\sqcup \notin const(\mathcal{DL})$, because:

$$\begin{aligned} &connect_{\sqcap}^{collect_{\sqcap}(C,D)}(choose_{\sqcap}) = 0 \\ \Longleftrightarrow\ &\forall E' \in collect_{\sqcap}(C,D) : choose_{\sqcap}(E') = 0 \end{aligned}$$

we can argue:

$$\begin{aligned} lcs(C,D) \not\equiv \top &\Longleftrightarrow \exists C_j \in \widehat{C}_{\sqcap}, \exists D_i \in \widehat{D}_{\sqcap} : C_j \sqsupseteq D_i \vee D_i \sqsupseteq C_j \\ &\Longleftrightarrow \exists C_j \in \widehat{C}_{\sqcap}, \exists D_i \in \widehat{D}_{\sqcap} : lcs(C_j, D_i) \not\equiv \top \\ &\Longrightarrow \exists C_j \in \widehat{C}_{\sqcap}, \exists D_i \in \widehat{D}_{\sqcap} : simi_{sym}(C_j, D_i) > 0 \\ &\Longleftrightarrow simi_{sym}(C,D) > 0 \end{aligned}$$

If $\sqcup \in const(\mathcal{DL})$:

$$\begin{aligned}
lcs(C,D) \not\equiv \top &\iff C \not\sqsupseteq \neg D \\
&\iff \textstyle\bigsqcap_j C_j \not\sqsupseteq \neg \bigsqcap_i D_i \\
&\iff \textstyle\bigsqcap_j C_j \not\sqsupseteq \bigsqcup_i \neg D_i \\
&\iff \exists D_i \in \widehat{D}_\sqcap : \textstyle\bigsqcap_j C_j \not\sqsupseteq \neg D_i \\
&\iff \exists D_i \in \widehat{D}_\sqcap, \exists C_j \in \widehat{C}_\sqcap : C_j \not\sqsupseteq \neg D_i \\
&\iff \exists D_i \in \widehat{D}_\sqcap, \exists C_j \in \widehat{C}_\sqcap : lcs(C_j, D_i) \not\equiv \top \\
&\implies \exists D_i \in \widehat{D}_\sqcap, \exists C_j \in \widehat{C}_\sqcap : simi_{sym}(C_j, D_i) > 0 \\
&\iff simi_{sym}(C,D) > 0
\end{aligned}$$

So Bounded is preserved in both cases.

2) Let $D = \bigsqcup_i D_i$ and $C = \bigsqcup_j C_j$. Since:

$$\begin{aligned}
&connect_\sqcup^{collect_\sqcup(C,D)}(choose_\sqcup) = 0 \\
\iff &\forall E' \in collect_\sqcup(C,D) : choose_\sqcup(E') = 0
\end{aligned}$$

we can argue:

$$\begin{aligned}
lcs(C,D) \not\equiv \top &\iff D \sqcup C \not\equiv \top \\
&\iff \forall D_i \in \widehat{D}_\sqcup, \forall C_j \in \widehat{C}_\sqcup : lcs(D_i, C_j) \not\equiv \top \\
&\iff \forall D_i \in \widehat{D}_\sqcup, \forall C_j \in \widehat{C}_\sqcup : simi_{sym}(C_j, D_i) > 0
\end{aligned}$$

3) Let $C = \forall r_C.C_1$. We can rephrase that $simi_{sym}$ is Peserving if:

$$simi_{sym}(C,D) = 0, C \not\equiv \top, D \not\equiv \top \implies lcs(C,D) \equiv \top$$

If $\sqcup \notin const(\mathcal{DL})$ this can only happen, if $C \sqsupseteq D$ or $D \sqsupseteq C$. For any D not of the form $D = \forall r_D.D_1$, this can never happen, so Peserving is preserved trivially. If D is of the form $D = \forall r_D.D_1$. We can use the the argumentation from Proof 12 to show that the $\forall$-quantification preserves Peserving.
If $\sqcup \in const(\mathcal{DL})$, unfortunately the conclusion of this rephrasing never holds. So in this case the $\forall$-quantification never preserves Peserving.
The argumentation for the $\exists$-quantification is the same.

□

Note that the problem of *3)* with $\sqcup \in const(\mathcal{DL})$ also applies to cases of calculating the similarity between a quantification and a primitive concept name. So just modifying the quantification rule, to always return some value bigger *0* would not solve this problem. We also would have to modify the special cases.

Like for Dissimilar Closed, this proof can be used for $simi_{asym}$ and $simi_{dual}$ as well.

Corollary 15. *Since the argumentations of Proof 14 can be adapted to either* $simi_{dir}(C, D)$ *or* $simi_{dir}(D, C)$ *and* 0 *is absorbing for* $\otimes$*, we can use the propositions of Theorem 9 also for* $simi_{asym}$*.*
The same holds for $simi_{dir^*}$ *and* $simi_{dual}$*.*

3.2.4 Structural Dependent

Both structural dependencies apply to a specific constructor. In each case, the other constructors have nearly no impact. As long as they all provide that $C = D \implies sim(C, D) = 1$, they will have no more impact at all. This is, because of the formulation of Structural Dependent, which speaks of a sequence of concept descriptions, implying that they are syntactical equal. Being Equivalence Closed covers this for $simi_{sym}$, $simi_{asym}$ and $simi_{dual}$. Furthermore the weighting function should not weight the equivalent participants with *0*.

Theorem 10. *Let* $simi_{sym}$ *be on a specific* $\mathcal{DL}$*. If the constructor* $\bigcirc \in const(\mathcal{DL})$*,* $\bigcirc \in \{\sqcap, \sqcup\}$*, then* $simi_{sym}$ *is Structural Dependent for* $\bigcirc$ *if the weight function* g *maps every concept description* $C \in \mathcal{C}(\mathcal{DL})$ *to a value bigger than* 0*.*

Proof 15 (Theorem 10)**.**

For all sequences $(C_n)_n$ of atoms with $\forall i, j \in \mathbb{N}, i \neq j : C_i \not\sqsupseteq C_j$, let $D_n := \bigcirc_{i<n} C_i \bigcirc D$ and $E_n := \bigcirc_{i<n} C_i \bigcirc E$. If the weight function g maps every concept description $C \in \mathcal{C}(\mathcal{DL})$ to a value bigger than 0, this assures that $connect_{\bigcirc}^{collect_{\bigcirc}(C,D)}(choose_{\bigcirc})$ is monotone in all arguments. Because for $simi_{sym}$ furthermore holds:

- $\widehat{D}_{\bigcirc} \cap \widehat{C}_{\bigcirc} \subseteq collect_{\bigcirc}(C, D)$, and
- $E' \in \widehat{D}_{\bigcirc} \cap \widehat{C}_{\bigcirc} \implies choose_{\bigcirc}(E') = 1$

we can define $simi_{sym}$ recursively:

$$f(0) \in [0, 2], simi_{sym}(D_0, E_0) = \frac{simi_{sym}(D, E) + g(C_0)}{f(0)}$$
$$f(n+1) = f(n) + g(C_n), simi_{sym}(D_{n+1}, E_{n+1}) = \frac{simi_{sym}(D_n, E_n) + g(C_n)}{f(n+1)}$$

where $f(n)$ collects the weights of the iteration, starting with the weights for $simi_{sym}(D, E)$ and C_0 in $f(0)$. Since every $g(C_n) > 0$, there exists for every $n > 0$ a $\varepsilon_n > 0$ with:

$$\varepsilon_n = mean(\{g(C_i) | i = 1, ..., n\})$$

what allows us to say:

$$simi_{sym}(D_n, E_n) = \frac{simi_{sym}(D, E) + g(C_0) + n * \varepsilon_n}{f(0) + n * \varepsilon_n}$$

with:

$$\lim_{n\to\infty} \frac{simi_{sym}(D,E) + g(C_0) + n * \varepsilon_n}{f(0) + n * \varepsilon_n} = \frac{\varepsilon_n}{\varepsilon_n} = 1$$

□

Because D_n and E_n are always the same length, we can easily adapt this proof $simi_{dir}(D_n, E_n)$, $simi_{dir}(E_n, D_n)$, $simi_{dir^*}(D_n, E_n)$ and $simi_{dir^*}(E_n, D_n)$. So if the fuzzy connector is monotone, we can also prove this lemma for $simi_{asym}$ and $simi_{dual}$. Once again we provided this by limiting the fuzzy connector to bounded t-norm.

Corollary 16. *Let $simi_{asym}$ be on a specific $\mathcal{DL}$. If the constructor $\bigcirc \in const(\mathcal{DL})$, $\bigcirc \in \{\sqcap, \sqcup\}$, then $simi_{asym}$ is Structural Dependent for $\bigcirc$ if the weight function g maps every concept description $C \in \mathcal{C}(\mathcal{DL})$ to a value bigger than 0.*

Corollary 17. *Let $simi_{dual}$ be on a specific $\mathcal{DL}$. If the constructor $\bigcirc \in const(\mathcal{DL})$, $\bigcirc \in \{\sqcap, \sqcup\}$, then $simi_{dual}$ is Structural Dependent for $\bigcirc$ if the weight function g maps every concept description $C \in \mathcal{C}(\mathcal{DL})$ to a value bigger than 0.*

3.2.5 Triangle Inequality

For our look at the Triangle Inequality, we assume the weighting function to be $g_{\mathcal{DL},def}$ the hole time. Any other weighting function could unpredictably distort the calculation for the conjunction and disjunction. Also we want to adapt the proof for the Jaccard Index to fulfil the Triangle Inequality like in [4]. Therefore we will need $g_{\mathcal{DL},def}$.
At the end, achieving one of our CSM to fulfil the Triangle Inequality is quite hard. This is, because it is a strong property. For literals a quantifications it can be achieved by a suitable literal measure. For the conjunction and disjunction it is not so easy.
To proof some lemmata for triple of values, we introduce an operator to calculate the similarity values within such a triple.

Definition 39 (Triangle Inequality for triples)**.** *Let (x_1, x_2, x_3) be a triple of values $x_i \in [0,1], i = 1,2,3$. We want to say that this triple fulfils the Triangle Inequality, if:*

$$\begin{aligned} 1 + x_1 &\geq x_2 + x_3 \\ 1 + x_2 &\geq x_1 + x_3 \\ 1 + x_3 &\geq x_1 + x_2. \end{aligned}$$

Let sim be a CSM on a specific $\mathcal{DL}$. We define for $C, D, E \in \mathcal{C}(\mathcal{DL})$ the function $Tri_{sim} : \mathcal{DL}^3 \longrightarrow [0,1]^3$ with:

$$Tri_{sim}(C,D,E) = (sim(C,D), sim(C,E), sim(D,E))$$

Further more we want to say that a triple (C, D, E) of $\mathcal{DL}$ concept descriptions fulfils the Triangle Inequality within sim iff $Tri_{sim}(C, D, E)$ fulfils the Triangle Inequality.

The following lemmatas will help us to prove that for special cases, the Triangle Inequality is fulfilled.

Lemma 7. *Let the triple (x_1, x_2, x_3) be a triple, that fulfil the Triangle Inequality. Then, for $\omega \in [0,1]$, the triple $(\omega + (1-\omega)x_1, \omega + (1-\omega)x_2, \omega + (1-\omega)x_3)$ also fulfils the Triangle Inequality as well.*

Proof 16 (Lemma 7)**.**

We can argue:

$$\begin{aligned}
1 + x_1 &\geq x_2 + x_3 \\
\frac{1-\omega}{1-\omega} + x_1 &\geq x_2 + x_3 \\
\frac{1}{1-\omega} + x_1 &\geq x_2 + x_3 + \frac{\omega}{1-\omega} \\
1 + (1-\omega)x_1 &\geq (1-\omega)x_2 + (1-\omega)x_3) + \omega \\
1 + \omega + (1-\omega)x_1 &\geq \omega + (1-\omega)x_2 + \omega + (1-\omega)x_3
\end{aligned}$$

what fulfils the Triangle Inequality for this case. Because this can be done for every possible order:

$$\begin{aligned}
1 + x_i &\geq x_j + x_k \\
i &\neq j \neq k \\
i, j, k &\in \{1, 2, 3\}
\end{aligned}$$

the Triangle Inequality is preserved.

□

Lemma 8. *Let the triple (x_1, x_2, x_3) and (y_1, y_2, y_3) be triples, that fulfil the Triangle Inequality, then the triple (x_1y_1, x_2y_2, x_3y_3) also fulfils the Triangle Inequality as well.*

Proof 17 (Lemma 8)**.**

For $x_1, x_2, x_3, y_1, y_2, y_3 \in [0,1]$ hold:

$$\begin{aligned}
1 + x_1 &\geq x_2 + x_3 \\
1 + y_1 &\geq y_2 + y_3.
\end{aligned}$$

We consider the case:

$$1 + x_1y_1 = x_2y_2 + x_3y_3$$

where w.l.o.g. $x_1 = y_1 = x_2 = y_2$ and $x_3 = y_3 = 1$. As for the hypothesis, the prerequisite are equalities. We now show, without violating the prerequisite, there is no way to increase the right hand side or decrease the left hand side of this case.

In in altering the x_2, we have approaches to increase the right hand side:

- If we try to increase x_2 by ε for decreasing x_3 by ε, this result in an increase of εy_2 for a decrease of εy_3. But since clearly $y_2 \leq y_3$, this will have no change or a decrease on the right hand side.
- If we try to increase x_2 by ε for increase of x_1 by ε, this results in an increase of the right hand side by εy_2 for an increase of the left hand side by εy_1. Since $y_1 = y_2$, both side rise in the same value, so the equality will still remain.

Decreasing the left hand side by altering x_1 by ε for a increase of $x_i, i \in \{2,3\}$ by ε, will result in an equal or higher decrease of the right hand side, than the left hand side. The reasons are the same as for the atering of x_2.
For altering y_1 or y_2, the argumentations are the same as for x_1 and x_2.
Because every possible configuration for the x_i and y_i, can be achieved by doing the mentioned alternations to this case or the case that $x_1 = y_1 = x_3 = y_3$ and $x_2 = y_2 = 1$, for all of this configurations at least the equality hold. So hold:

$$1 + x_1y_1 \leq x_2y_2 + x_3y_3$$

Because the mentioned prerequisite hold for every possible order:

$$\begin{aligned} 1 + x_i &\geq x_j + x_k \\ 1 + y_l &\geq y_m + y_n \\ &i \neq j \neq k \\ &l \neq m \neq n \\ &i, j, k, l, m, n \in \{1, 2, 3\} \end{aligned}$$

this proof can be done for all

$$1 + x_iy_l = x_my_2 + x_ny_3$$

so the triple (x_1y_1, x_2y_2, x_3y_3) fulfils the Triangle Inequality.

□

Again we start with the proof for $simi_{sym}$. There for we first define the requirements for lm^{sym} to fulfil the Triangle Inequality.

Definition 40. *We want to denote every symmetric literal measure lm^{sym} that, for $A, B, C \in N_{L_{pr}}$ and $r, s, t \in N_R$, fulfils the Triangle Inequality:*

$$\begin{aligned} 1 + lm^{sym}(A, B) &\geq lm^{sym}(A, C) + lm^{sym}(B, C) \\ 1 + lm^{sym}(r, s) &\geq lm^{sym}(r, t) + lm^{sym}(s, t) \end{aligned}$$

by adding $\cdot_{TrIn}$: lm^{sym}_{TrIn}:

In this definition are already the requirements for the quantifications included. They are not necessary for lm^{sym} to fulfil the Triangle Inequality in $N_{L_{pr}}$. But we included them anyway, so we do not have to bring a second definition for lm^{sym} to fulfil the Triangle Inequality in N_R. We just have to keep in mind, that these requirements can occur independently, when we later look for counter examples in the analysis of the properties.

Theorem 11. *Let $simi_{sym}$ be on a specific $\mathcal{DL}$. We can say for the constructors of $\mathcal{DL}$, that for $simi_{sym}$:*

1) *The conjunction $\sqcap$ and the disjunction $\sqcup$ preserve Triangle Inequality if additionally for all possible participants C_i of the conjunction/disjunction holds:*

$$simi_{sym}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

2) *The $\exists$- and $\forall$-quantification preserve the Triangle Inequality for lm^{sym}_{TrIn}.*

Proof 18 (Theorem 11)**.**

1) By the additional assumption in $simi_{sym}$ for two disjunctions $C, D \in \mathcal{C}(\mathcal{DL}), C = \bigsqcup_i C_i, D = \bigsqcup_j D_j$ holds:

$$simi_{sym}(C, D) = \frac{|\widehat{C}_{\sqcup} \cap \widehat{D}_{\sqcup}|}{|\widehat{C}_{\sqcup} \cup \widehat{D}_{\sqcup}|}$$

   Since this is similar to the Jaccard Index ([4]), we can adapt the prove of the Triangle Inequality for the Jaccard Index (see [5] and [4]).
   For the conjunction, the argumentation is the same.

2) Let $C, D, E \in \mathcal{C}(\mathcal{DL})$, $C = \forall r_C.C_1$ and $Tri_{sym}(C, D, E) = (x_1, x_2, x_3)$. Due to the treatment of the $\forall$-quantification in $simi_{sym}$, if at least one of D and E is no $\forall$-quantification, then the triple (x_1, x_2, x_3) contains at least two zeros and so the Triangle Inequality is fulfilled.
   If $D = \forall r_D.D_1$ and $E = \forall r_E.E_1$, then because $Tri(C_1, D_1, E_1)$ fulfils the Triangle Inequality, Lemma 7 assures that

$$(\omega + (1-\omega)x_1, \omega + (1-\omega)x_2, \omega + (1-\omega)x_3) = (y_1, y_2, y_3)$$

   fulfils the Triangle Inequality as well. In conclusion to this, Lemma 8 assures that furthermore

$$(lm^{sym}_{TrIn}(r_C, r_D)y_1, lm^{sym}_{TrIn}(r_C, r_E)y_1, lm^{sym}_{TrIn}(r_D, r_E)y_1)$$

   fulfils the Triangle Inequality and so the property is preserved.
   The argumentation for the $\exists$-quantification is the same.

$\square$

Example 2. *Let $C = C_1 \sqcup C_2, D, E$ with $C_1, C_1, D, E \in N_{L_{pr}}$, $lm^{sym}_{TrIn}(C_1, D) = lm^{sym}_{TrIn}(C_2, E) > \frac{3}{4}$ and all other similarities between primitive concept names are 0. This lm^{sym}_{TrIn} clearly fulfils the Triangle Inequality, but:*

$$1 + simi_{sym}(E, D) < simi_{sym}(C, D) + simi_{sym}(C, E)$$

because

$$1 + 0 = \frac{\frac{3}{4} + \frac{3}{4} + 0}{3} + \frac{\frac{3}{4} + \frac{3}{4} + 0}{3} < simi_{sym}(C, D) + simi_{sym}(C, E)$$

So $Tri(C, D; E)$ fulfils not the Triangle Inequality.

Example 2 shows, why for $simi_{sym}$ the conjunction and disjunction not preserve the Triangle Inequality for general. Note that we proved an implication. It could be possible, that there are more general restrictions to make the conjunction and disjunction preserve the Triangle Inequality. For now we keep with the ones, we can synthesise by simple limitations of $const(\mathcal{DL})$ and lm_{def}^{sym}.

Corollary 18. *For $\mathcal{DL}$, $simi_{sym}$ and the Triangle Inequality we have:*

1) *The conjunction $\sqcap$ preserves the fulfilling of the Triangle Inequality for lm_{def}^{sym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcap\}$*

2) *The disjunction $\sqcup$ preserves the fulfilling of the Triangle Inequality for lm_{def}^{sym} and $const(\mathcal{DL}) \subseteq \{(\neg), \sqcup\}$*

Because of the asymmetric structure of $simi_{asym}$ and $simi_{dual}$, lm^{asym} and lm^{dual} are not necessarily symmetric. So their definition to fulfil the Triangle Inequality, has to regard both ways they can be applied.

Definition 41. *We define an asymmetric literal measure lm^{asym} or lm^{dual} to fulfil the Triangle Inequality, if for $A, B, C \in N_{L_{pr}}$, $r, s, t \in N_R$ holds:*

$$\begin{aligned}
1 + lm^*(A,B) &\geq lm^*(A,C) + lm^*(B,C)\\
1 + lm^*(B,A) &\geq lm^*(C,A) + lm^*(C,B)\\
1 + lm^*(r,s) &\geq lm^*(r,t) + lm^*(s,t)\\
1 + lm^*(s,r) &\geq lm^*(t,r) + lm^*(t,s)
\end{aligned}$$

with $lm^* \in \{lm^{asym}, lm^{dual}\}$. We want to denote those by adding $_{TrIn}$: lm_{TrIn}^{asym}, lm_{TrIn}^{dual}.

Also the fuzzy connector will have a major impact. Therefore we also define a version, that regards the used fuzzy connector.

Definition 42. *We define an asymmetric literal measure lm^{asym} or lm^{dual} to fulfil the Triangle Inequality for fuzzy connector $\otimes$, if for $A, B, C \in N_{L_{pr}}$, $r, s, t \in N_R$ holds:*

$$\begin{aligned}
&1 + lm^*(A,B) \otimes lm^*(B,A)\\
&\geq lm^*(A,C) \otimes lm^*(C,A) + lm^*(B,C) \otimes lm^*(C,B)\\
&1 + lm^*(r,s) \otimes lm^*(s,r)\\
&\geq lm^*(r,t) \otimes lm^*(t,r) + lm^*(s,t) \otimes lm^*(t,s)
\end{aligned}$$

with $lm^* \in \{lm^{asym}, lm^{dual}\}$. We want to denote it by adding $_{TrIn(\otimes)}$: $lm_{TrIn(\otimes)}^{asym}$, $lm_{TrIn(\otimes)}^{asym}$.

For the Product t-norm it is easy to show, that the first definition implies the second one.

Corollary 19. *Lemma 8 implies, that for the Product t-norm $\otimes_{prod}$ every asymmetric literal measure lm_{TrIn}^{asym} is also a $lm_{TrIn(\otimes_{prod})}^{asym}$.*
The same holds for lm_{TrIn}^{dual}.

The big disadvantage of $simi_{asym}$ and $simi_{dual}$ is, that the fuzzy connector is an additional unpredictable variable. Even with our restriction to bounded t-norm, we can only conclude:

$$x \otimes y \leq min(x, y)$$

This brings the problem, that in some cases every fuzz connector must be checked separately.

Theorem 12. *Let $simi_{asym}$ be on a specific $\mathcal{DL}$, using the fuzzy connector $\otimes$. We can say for the constructors of $\mathcal{DL}$, that for $simi_{asym}$:*

1) *The conjunction $\sqcap$ and the disjunction $\sqcup$ preserve Triangle Inequality if additionally for all possible participants C_i of the conjunction/disjunction holds:*

$$simi_{dir}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

and the fuzzy connector used by $simi_{asym}$ is the Hamacher product $\otimes_{H_0}$.

2) *The $\exists$- and $\forall$-quantification preserve the Triangle Inequality for lm^{asym}_{TrIn} and the fuzzy connector used by $simi_{asym}$ is the Product t-norm $\otimes_{prod}$.*

Proof 19 (Theorem 11)**.**

1) By the additional assumption in $simi_{sym}$ for two disjunctions $C, D \in \mathcal{C}(\mathcal{DL}), C = \bigsqcup_i C_i, D = \bigsqcup_j D_j$ holds:

$$simi_{sym}(C, D) = \frac{|\widehat{C}_\sqcup \cap \widehat{D}_\sqcup|}{|\widehat{C}_\sqcup|}$$

Since this is similar to $simi_d$ from ([4]) and also $\otimes_{H_0}$ is assumed as fuzzy connector, we can adapt the proof of $simi_d[\cdot, pm_{def}, g_{def},]$ with $\otimes_{H_0}$ being the Jaccard Index ([4]) to prove the fulfilling of the Triangle Inequality.
For the conjunction, the argumentation is the same.

2) Let $C = \forall r.C_1$, $D = \forall s.D_1$ and $E = \forall t.E_1$. For shorter formulas, we denote:

$$x_a = lm^{asym}_{TrIn}(r, s), x_{a'} = lm^{asym}_{TrIn}(s, r)$$
$$x_b = lm^{asym}_{TrIn}(r, t), x_{b'} = lm^{asym}_{TrIn}(t, r)$$
$$x_c = lm^{asym}_{TrIn}(s, t), x_{c'} = lm^{asym}_{TrIn}(t, s)$$
$$x_1 = simi_{asym}(C_1, D_1), x_{1'} = simi_{asym}(D_1, C_1)$$
$$x_2 = simi_{asym}(C_1, E_1), x_{2'} = simi_{asym}(E_1, C_1)$$
$$x_3 = simi_{asym}(D_1, E_1), x_{3'} = simi_{asym}(E_1, D_1)$$
$$x_\alpha = \omega + (1 - \omega)x_1, x_{\alpha'} = \omega + (1 - \omega)x_{1'}$$
$$x_\beta = \omega + (1 - \omega)x_2, x_{\beta'} = \omega + (1 - \omega)x_{2'}$$
$$x_\gamma = \omega + (1 - \omega)x_3, x_{\gamma'} = \omega + (1 - \omega)x_{3'}$$

From Corollary 19 follows, that the literal measure is also a $lm^{asym}_{TrIn(\otimes_{prod})}$. For shorter formulas, we assume for the rest of the proof $\otimes = \otimes_{prod}$.
We can argue:

$$\begin{aligned}
2 + x_1 + x_{1'} &\geq x_2 + x_{2'} + x_3 + x_{4'} \\
2\frac{\omega - \omega^2}{\omega - \omega^2} + x_1 + x_{1'} &\geq x_2 + x_{2'} + x_3 + x_{3'} \\
\frac{2\omega}{\omega - \omega^2} + x_1 + x_{1'} &\geq x_2 + x_{2'} + x_3 + x_{3'} + \frac{\omega^2}{\omega - \omega^2} \\
2\omega + (\omega - \omega^2)x_1 + (\omega - \omega^2)x_{1'} &\geq (\omega - \omega^2)x_2 + (\omega - \omega^2)x_{2'} \\
&\quad + (\omega - \omega^2)x_3 + (\omega - \omega^2)x_{3'} + 2\omega^2
\end{aligned}$$

Because of $lm^{asym}_{TrIn(\otimes)}$ we can argue:

$$\begin{gathered}
1 + x_1 \otimes x_{1'} \geq x_2 \otimes x_{2'} + x_3 \otimes x_{3'} \\
(1-\omega)^2 + (1-\omega)^2[x_1 \otimes x_{1'}] \geq (1-\omega)^2[x_2 \otimes x_{2'}] + (1-\omega)^2[x_3 \otimes x_{3'}]
\end{gathered}$$

what implies with the previous formula:

$$\begin{aligned}
&2\omega + (\omega - \omega^2)x_1 + (\omega - \omega^2)x_{1'} + (1-\omega)^2 + (1-\omega)^2 x_1 \otimes (1-\omega)^2 x_{1'} \\
\geq\; &(\omega - \omega^2)x_2 + (\omega - \omega^2)x_{2'} + (\omega - \omega^2)x_3 + (\omega - \omega^2)x_{3'} + 2\omega^2 \\
&+ (1-\omega)^2[x_2 \otimes x_{2'}] + (1-\omega)^2[x_3 \otimes x_{3'}]
\end{aligned}$$

Looking on each side of this inequality separately, we can argue for the left hand side:

$$\begin{aligned}
&2\omega + (\omega - \omega^2)x_1 + (\omega - \omega^2)x_{1'} \\
&\quad + (1-\omega)^2 + (1-\omega)^2 x_1 \otimes (1-\omega)^2 x_{1'} \\
&= 2\omega - \omega^2 + \omega^2 + \omega(1-\omega)x_1 + \omega(1-\omega)x_{1'} \\
&\quad + (1-\omega)^2 + (1-\omega)^2 x_1 \otimes (1-\omega)^2 x_{1'} \\
&= 2\omega - \omega^2 + (1-\omega)^2 + ([\omega + (1-\omega)x_1] \otimes [\omega + (1-\omega)x_{1'}]) \\
&= 1 + (x_\alpha \otimes x_{\alpha'})
\end{aligned}$$

and for the right hand side:

$$\begin{aligned}
&(\omega - \omega^2)x_2 + (\omega - \omega^2)x_{2'} + (\omega - \omega^2)x_3 + (\omega - \omega^2)x_{3'} + 2\omega^2 \\
&\quad + (1-\omega)^2[x_2 \otimes x_{2'}] + (1-\omega)^2[x_3 \otimes x_{3'}] \\
&= \omega^2 + \omega(1-\omega)x_2 + \omega(1-\omega)x_{2'} + (1-\omega)^2[x_2 \otimes x_{2'}] \\
&\quad + \omega^2 + \omega(1-\omega)x_3 + \omega(1-\omega)x_{3'} + (1-\omega)^2[x_3 \otimes x_{3'}] \\
&= ([\omega + (1-\omega)x_2] \otimes [\omega + (1-\omega)x_{2'}]) + \\
&\quad ([\omega + (1-\omega)x_3] \otimes [\omega + (1-\omega)x_{3'}]) \\
&= (x_\beta \otimes x_{\beta'}) + (x_\gamma \otimes x_{\gamma'})
\end{aligned}$$

So holds:

$$1 + (x_\alpha \otimes x_{\alpha'}) \geq (x_\beta \otimes x_{\beta'}) + (x_\gamma \otimes x_{\gamma'})$$

This can be argued for every possible order of the values and implies that the triple $((x_\alpha \otimes x_{\alpha'}), (x_\beta \otimes x_{\beta'}), (x_\gamma \otimes x_{\gamma'}))$ fulfils the Triangle Inequality. Since $\otimes$ is the Product t-norm, we can say that by Lemma 8 holds:

$$1 + (x_a \otimes x_\alpha x_{a'} \otimes x_{\alpha'}) \geq (x_b \otimes x_\beta x_{b'} \otimes x_{\beta'}) + (x_c \otimes x_\gamma x_{c'} \otimes x_{\gamma'})$$

that can be transposed to:

$$1 + (x_a x_\alpha \otimes x_{a'} x_{\alpha'}) \geq (x_b x_\beta \otimes x_{b'} x_{\beta'}) + (x_c x_\gamma \otimes x_{c'} x_{\gamma'})$$

and so implies, that in this case the $\forall$-quantification preserves the fulfilling of the Triangle Inequality.
The argumentation for the $\exists$-quantification is the same.

$\square$

Note that *1)* does not hold for $\otimes_{min}$ or $\otimes_{prod}$. Also *2)* does not hold for $\otimes_{min}$ or $\otimes_{H_0}$. A counterexample for *2)* and $\otimes_{min}$ is

$$\begin{aligned}
1 + min\{x_a x_1, x_{a'} x_{1'}\} &< min\{x_b x_2, x_{b'} x_{2'}\} + min\{x_c x_3, x_{c'} x_{3'}\} \\
x_a &= x_b = x_c = x_3 = x_{c'} = x_{3'} = x_{2'} = (1 - \eta) \\
x_1 &= x_{a'} = x_{1'} = x_2 = x_{b'} = (1 - \varepsilon) \\
\varepsilon &> \eta \\
\varepsilon + \varepsilon\eta + \eta^2 &> \varepsilon^2 + 3\eta \\
\eta &\longrightarrow 0
\end{aligned}$$

Here η and ε are artificial differences to assure that the values can be different to *1*. This is needed to allow Dissimilar Closed. If $\eta = 0$, this example would violate Dissimilar Closed.

Corollary 20. *We can adapt Proof 19 to show that for $simi_{dual}$:*

1) The conjunction $\sqcap$ and the disjunction $\sqcup$ preserve Triangle Inequality if additionally for all possible participants C_i of the conjunction/disjunction holds:

$$simi_{dir*}(C_1, C_2) = 0 \Longleftrightarrow C_1 \not\equiv C_2$$

and the fuzzy connector used by $simi_{dual}$ is the Hamacher product $\otimes_{H_0}$.

2) The $\exists$- and $\forall$-quantification preserve the Triangle Inequality for lm^{dual}_{TrIn} and the fuzzy connector used by $simi_{dual}$ is the Product t-norm $\otimes_{prod}$.

4 Modifications to fulfil more Formal Properties

Based on Chapter 3, in this chapter we want to show some approaches and modification for our current CSMs to purposely change their fulfilling of the formal properties. We will also use these modified CSMs for the analysis in the next chapter. First we want to take a look at the literal measures. We define some lm to fulfil as much formal properties as possible. Then we will take a look at some modifications for quantification rules and the rules of conjunction and disjunction.

4.1 Alternative Literal Measures

4.1.1 Special Literal Measure

In Chapter 3 we introduced requirements for the literal measure to fulfil certain formal properties. We now want to implement as much of these requirement as possible in one literal measure. Except for Dissimilar Closed and Peserving this will work out fine. Depending on, whether $\sqcup \in const(\mathcal{DL})$, one of both properties is trivial fulfilled and for the other the requirement must be true. Unfortunately both requirements can not be applied to the same literal measure. Therefore the following definitions will distinguish two literal measures.

Definition 43. *The literal measure $lm^{asym}_{\not\sqcup} : N^2_{L_{pr}} \cup N^2_R \longrightarrow [0,1]$, with $\rho \in (0, \frac{1}{2}]$ and:*

$$lm^{asym}_{\not\sqcup}(A,B) := \begin{cases} 1 & \text{if } A = B \\ 0 & \text{if } A \neq B \end{cases}$$

$$lm^{asym}_{\not\sqcup}(r,s) := \begin{cases} 1 & \text{if } r = s \text{ or } s \sqsubseteq r \\ \rho & \text{if } s \not\sqsubseteq r \text{ or } r \sqsubseteq s \\ 0 & \text{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Reverse Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \notin const(\mathcal{DL})$.

The literal measure $lm_{\sqcup}^{asym} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$*, with* $\rho \in (0, \frac{1}{2}]$ *and:*

$$lm_{\sqcup}^{asym}(A,B) := \begin{cases} 1 & \textit{if } A = B \\ 0 & \textit{if } A = \neg B \\ \rho & \textit{otherwise} \end{cases}$$

$$lm_{\sqcup}^{asym}(r,s) := \begin{cases} 1 & \textit{if } r = s \textit{ or } s \sqsubseteq r \\ \rho & \textit{if } s \not\sqsubseteq r \textit{ or } r \sqsubseteq s \\ 0 & \textit{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Reverse Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \in const(\mathcal{DL})$*.*

Note that for these literal measure Subsumption Preserving is not a required property, since it is achieved by design.

Definition 44. *The dual literal measure* $lm_{\not\sqcup}^{dual} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$*, with* $\rho \in (0, \frac{1}{2}]$ *and:*

$$lm_{\not\sqcup}^{dual}(A,B) := \begin{cases} 1 & \textit{if } A = B \\ 0 & \textit{if } A \neq B \end{cases}$$

$$lm_{\not\sqcup}^{dual}(r,s) := \begin{cases} 1 & \textit{if } r = s \textit{ or } s \sqsupseteq r \\ \rho & \textit{if } s \not\sqsupseteq r \textit{ or } r \sqsupseteq s \\ 0 & \textit{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \notin const(\mathcal{DL})$*.*
The dual literal measure $lm_{\sqcup}^{dual} : N_{L_{pr}}^2 \cup N_R^2 \longrightarrow [0,1]$*, with* $\rho \in (0, \frac{1}{2}]$ *and:*

$$lm_{\sqcup}^{dual}(A,B) := \begin{cases} 1 & \textit{if } A = B \\ 0 & \textit{if } A = \neg B \\ \rho & \textit{otherwise} \end{cases}$$

$$lm_{\sqcup}^{dual}(r,s) := \begin{cases} 1 & \textit{if } r = s \textit{ or } s \sqsupseteq r \\ \rho & \textit{if } s \not\sqsupseteq r \textit{ or } r \sqsupseteq s \\ 0 & \textit{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \in const(\mathcal{DL})$*.*

Note that for these literal measures Reverse Subsumption Preserving is not a required property, since it is achieved by design.

Definition 45. *The symmetric literal measure* $lm^{sym}_{\not\sqcup} : N^2_{L_{pr}} \cup N^2_R \longrightarrow [0,1]$*, with* $\rho \in (0, \frac{1}{2}]$ *and:*

$$lm^{sym}_{\not\sqcup}(A,B) := \begin{cases} 1 & \text{if } A = B \\ 0 & \text{if } A \neq B \end{cases}$$

$$lm^{sym}_{\not\sqcup}(r,s) := \begin{cases} 1 & \text{if } r = s \\ \rho & \text{if } s \sqsubseteq r \text{ or } r \sqsubseteq s \\ 0 & \text{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Subsumption Preserving, Reverse Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \notin const(\mathcal{DL})$*.*
The symmetric literal measure $lm^{sym}_{\sqcup} : N^2_{L_{pr}} \cup N^2_R \longrightarrow [0,1]$*, with* $\rho \in (0, \frac{1}{2}]$ *and:*

$$lm^{sym}_{\sqcup}(A,B) := \begin{cases} 1 & \text{if } A = B \\ 0 & \text{if } A = \neg B \\ \rho & \text{otherwise} \end{cases}$$

$$lm^{sym}_{\sqcup}(r,s) := \begin{cases} 1 & \text{if } r = s \\ \rho & \text{if } s \sqsubseteq r \text{ or } r \sqsubseteq s \\ 0 & \text{otherwise} \end{cases}$$

is by construction Equivalent Invariant, Equivalent Closed, Subsumption Preserving, Reverse Subsumption Preserving, Dissimilar Closed, Peserving and fulfils the Triangle Inequality, for $\sqcup \in const(\mathcal{DL})$*.*

The claimed properties are hold, since these literal measures fulfil the requirements we defined in Chapter 3. By $\rho \leq \frac{1}{2}$ the Triangle Inequality in N_R is assured trivially, since so the sum of two similarity values is never bigger than *1*, except two primitive literals are equal. If the right hand side is never bigger than *1*, then the left hand side *1* of the Triangle Inequality will hold the statement. If at least two of the participating primitive literal are equal, this means that on the right hand side a *1* can appear. Equivalent Closed assures, that this can only happen for equality of this literal. Furthermore Equivalent Invariant assures that the other similarity values are the same, so the Triangle Inequality holds again. The following example illustrates this.

Example 3.

case 1: $x_1, x_2, x_3 \in (0, \frac{1}{2}]$

$$1 + x_1 \geq x_2 + x_3$$

case 2: $x_1, x_3 \in (0, \frac{1}{2}] \cup \{1\}, x_2 = 1$

$$\begin{aligned} \textit{Equivalent Closed} + \textit{Equivalent Invariant} &\Longrightarrow x_1 = x_3 \\ &\Longrightarrow 1 + x_1 \geq x_2 + x_3 \end{aligned}$$

Note that this trivial case to fulfil the Triangle Inequality does not cover all possibilities to do that.

4.1.2 Combining $simi_{asym}$ and $simi_{dual}$

For an attempt to combine $simi_{asym}$ and $simi_{dual}$, there must hold for the new combined literal measure lm_{comb}:

$$\forall r \in N_R : t \sqsubseteq s \Longrightarrow lm_{comb}(r, s) = lm_{comb}(r, t)$$

Since this also holds for $r \in \{s, t\}$, we can conclude that

$$lm_{comb}(s, s) = lm_{comb}(s, t) = lm_{comb}(t, t)$$

This would mean, that either $\mathcal{R}$ only contains equivalences between role names, or Equivalence Closed no longer holds.

4.2 Modifying Quantifier Rules

4.2.1 Equivalence exclusive Quantifier Rules

One way to provide Subsumption Preserving for the disjunction and Reverse Subsumption Preserving for the conjunction, even with quantifications for constructors, is to modify the rules for the quantifications. If the rule just return *1* for the case of absolute equivalence, and *0* otherwise, the more general requirements for our disjunction to preserve Subsumption Preserving or for our conjunction to preserve Reverse Subsumption Preserving are fulfilled.

Definition 46 (equivalence exclusive quantifier rule)**.** *Let* $\bigcirc \in \{\forall, \exists\}$*. For a specific description logic* $\mathcal{DL}$*,* $r, c \in N_R$ *and* $C, C', D, D' \in \mathcal{C}(\mathcal{DL})$ *we define an* equivalence exclusive rule *for* $\bigcirc$*-quantification:*

If $C = \bigcirc r.C', D = \bigcirc s.D'$

$$sim(C, D) = \begin{cases} 1 & C' \equiv D', r \equiv s \\ 0 & \textit{otherwise} \end{cases}$$

This allows us to also use $\exists$- and $\forall$-quantifications in cases like Corollary 4 *2)* or Corollary 5. Note that this does not allow us to use conjunction and disjunction in the same DL and let the corresponding CSM Subsumption Preserving or Reverse Subsumption Preserving. Depending on whether we have a conjunction of disjunctions or a disjunction of conjunctions as normal form, this would need the inner of both constructors to have an equivalence exclusive rule in the CSM. This again would eliminate every similarity knowledge we could gain from this layer.

A disadvantage of this rule is, that for the quantifications and all deeper layers of the concept descriptions, also every similarity knowledge is lost. That is why, for future work we suggest a Structural Dependence like property for the quantifications. The following definition could give a hint:

Definition 47 (Quantifier Role-Name Dependency). *Let* $\bigcirc \in \{\forall, \exists\}$*. A CSM sim with an literal measure lm is* Role-Name Dependent *for* $\bigcirc$ *if for* $C = \bigcirc r.C', D = \bigcirc s.D', E = \bigcirc t.E'$ *hold:*

$$C' \equiv D' \equiv E', lm(r,s) > lm(r,t) \Longrightarrow sim(C,D) > sim(C,E)$$

This definition can also easily be adapted to asymmetric CSMs or a Quantifier Concept Dependency property. Since we have already enough properties to cover in this paper, we will leave these properties to future works.
Note that by design, these equivalence exclusive rules also automatically violate Peserving and preserve Equivalence Invariant, Equivalence Closed and Dissimilar Closed.

4.3 Modifying Conjunction and Disjunction Rules

4.3.1 Equivalence exclusive Conjunction or Disjunction Rule

As for the quantifications, we can introduce for conjunction and disjunction equivalence exclusive rules:

Definition 48 (equivalence exclusive rule). *Let* $\bigcirc \in \{\sqcap, \sqcup\}$*. For a specific description logic* $\mathcal{DL}$*, and* $C, C_i, D, D_j \in \mathcal{C}(\mathcal{DL})$ *we define an* equivalence exclusive rule *for* $\bigcirc$*:*

If $C = \bigcirc_i C_i, D = \bigcirc_j D_j$

$$sim(C,D) = \begin{cases} 1 & \forall C_i \in \widehat{C}_{\bigcirc} : \exists D_j \in \widehat{D}_{\bigcirc} : C_i \equiv D_j \text{ and} \\ & \forall D_j \in \widehat{D}_{\bigcirc} : \exists C_i \in \widehat{C}_{\bigcirc} : C_i \equiv D_j \\ 0 & \text{otherwise} \end{cases}$$

At this time, this function is the only rule that provides us the possibility to have $\{\sqcap, \sqcup\} \in const(\mathcal{DL})$ and still be Subsumption Preserving or Reverse Subsumption Preserving. As mentioned, such a function eliminates every similarity knowledge of the concept descriptions deeper layers (except for equivalence). Since so subsumptions from the deeper layers will not be respected, this rule automatically violates Peserving. It also is not Structural Dependent, because as long as one $\bigcirc$-participant is not equivalent, it still returns 0. Furthermore it is possible, that a normal form ,that do not choose uniquely one out of several equivalent primitive concept names, then two equivalent concept names can have different similarities to other primitive concept names. So Equivalence Invariant could be threatened. For the most uses, this will not be the case, since using this rule is motivated by passing *1* and *0* to the upper layers. So also the deeper layers, especially $N_{L_{pr}}$, should also act equivalence exclusive.
On the other hand, the rule trivially provides Dissimilar Closed to the upper layers of the concept description. So even if the deeper layers did not provide Dissimilar Closed, this rule will obtain it for the higher layers. The same holds for Equivalence Closed. The Triangle Inequality, Subsumtion Preserving and Reverse Subsumtion Preserving are provided trivially.

4.3.2 Trivial Structural Dependent with max

A trivial way to make the conjunction or disjunction rule Structural Dependent, is the maximum function max. As long as holds $C = D \Longrightarrow sim(C, D) = 1$, we have can just use it as $connect_{\bigcirc}, \bigcirc \in \{\sqcap, \sqcup\}$.

Definition 49. *Let $\bigcirc \in \{\sqcap, \sqcup\}$. We define a CSM $sim_{max(\bigcirc)}$, that uses the maximum function max for the $\bigcirc$-rule by replacing the corresponding rule as follows:*

If $C = \bigcirc_i C_i, D = \bigcirc_j D_j$

$$\begin{aligned} sim_{max(\bigcirc)}(C, D) &= connect_{\bigcirc}^{collect_{\bigcirc}(C,D)}(choose_{\bigcirc}) \\ &= max\{x | x = choose_{\bigcirc}(E'), E' \in collect_{\bigcirc}(C, D)\} \end{aligned}$$

As long as we can guarantee that at least one pair of the equivalent participants of C and D is in $collect_{\bigcirc}(C, D)$ and $choose_{\bigcirc}$ maps to *1*, this rule provides Structural Dependence for $\bigcirc$. Furthermore it preserves Subsumption Preserving, Reverse Subsumption Preserving, Dissimilar Closed (broken again by $\{\sqcup, (\neg)\} \subseteq const(\mathcal{DL})$), Peserving and Equivalence Invariant. On the down side, it does not preserve Equivalence Closed, automatically violates the Triangle Inequality and may not fit to the uses of a CSM.
If we want to prevent such a trivial Structural Dependence in future works, we suggest a stronger requirement. The $connect_{\bigcirc}$ function should be strictly monotone in all its elements. Still the $\lim_{n \to \infty} sim(D_n, E_n) = 1$ is important, to prevent the function form converging to a different limiting value. At this point, also note that our actual Structural Dependence property is independent from Equivalence Invariant. Its premise just demands a sequence of concept descriptions, so they are already syntactical equal. If we want to strengthen the definition of Structural Dependence in future works, we should also think about changing this to semantic equality. For this paper we stay with the actual definition from Chapter 2.

4.4 Trivial Triangular Inequality

As mentioned in 4.1.1, the Triangle Inequality is trivially achieved, if all possible similarity measures are in $[0, \frac{1}{2}]$, if equivalence closed even for $[0, \frac{1}{2}] \cup \{1\}$. This allows us to change every rule of a concept similarity measure to preserve Triangle Inequality. So for a CSM sim on a specific $\mathcal{DL}$, $C, D \in \mathcal{C}(\mathcal{DL})$ and a rule $*_{rule}$ of sim achieved by the function $f_{rule} : \mathcal{C}(\mathcal{DL})^2 \longrightarrow [0, 1]$

If in sim the rule $*_{rule}$ applies on C and D

$$sim(C, D) = f_{rule}(C, D)$$

we can bring a new rule $*_{rule'}$ to replace $*_{rule}$. This $*_{rule'}$ will be achieved by

$f_{rule'} : \mathcal{C}(\mathcal{DL})^2 \longrightarrow [0,1]$, which we define as:

$$f_{rule'}(C,D) = \begin{cases} 1 & \text{if } f_{rule}(C,D) = 1 \\ \frac{1}{2} f_{rule}(C,D) & \textit{otherwise} \end{cases}$$

Note that if we do not need Equivalence Closed we can just drop the first case of $f_{rule'}$. This function preserves Equivalent Invariant, Subsumption Preserving, Reverse Subsumption Preserving, Dissimilar Closed and Peserving if f_{rule} does. Subsumption Preserving and Reverse Subsumption Preserving are taken on, because the factor $\frac{1}{2}$ does just scale the values, so the relations between the similarity values stay untouched. The only value not scaled, are the *1*s from equivalences. But since they already were the highest values before, also their relations within subsumptions the the same. Since multiplication with $\frac{1}{2}$ also has *0* as absorbing element, Dissimilar Closed and Peserving are taken on from f_{rule}. Equivalence Invariant is taken on form f_{rule} because f_{rule} and $f_{rule'}$ are isomorphic to each other.
For $\sqcap$ and $\sqcup$, this modification is not Structural dependent, since the limiting value is at best $\frac{1}{2}$.

5 Formal Concept Analysis of the Formal Properties

In this chapter we will use the method of the FCA to analyse the relation between the formal properties. The FCA is a method to generate from a set of objects with attributes a set of implication that hold between those attributes. Note that the word *concept* in FCA is independent of the concepts of DLs. With the FCA we try to generate implications between the different possible combinations of the formal properties. For this we will use the dozens of CSMs we can generate with the knowledge from Chapter 3 and Chapter 4. Since the FCA just generates implications that hold within its context, we later have to proof those implication for the general case.
To perform the analysis we use the *Concept Explorer* (*ConExp* v1.3 [6]) tool, which provides all basic functionalities. It was downloaded from:

http://conexp.sourceforge.net/download.html

We first set up our analysis by generation the CSMs. Then we perform the attribute exploration of *ConExp*. At last we test the results for their generality.

More detailed information concerning FCA can be found in [2].

5.1 Set up ConExp for the FCA

The *ConExp* tool implements the basic functionality for a FCA. All we need, is a sufficient data base of objects with their attributes. In *ConExp* this is realised in form of a table, where each row identifies an object and each column an attribute. The entries of the table mark, whether the object has this attribute or not. As objects we have the different CSMs and as attributes the formal properties.
With five constructors for the DLs, three versions of CSMs, of which the literal measure can fulfil put to seven requirements ($lm_{TrIn}, lm_{SP}, lm_{RSP} \cdots$) and the modification from Chapter 4 we theoretically can generate thousands of objects for the FCA. But since we may not need all of them, we agree on a starting database and use the other knowledge for the attribute exploration. In particular we want to use all possible combinations of the constructors and $simi_{asym}$, $simi_{dual}$ and $simi_{sym}$ with different literal measures. We want to apply the dependency properties just to those CSMs, where the relevant constructors occur in the constructor set.
As literal measures for the CSMs we want to use one, that fulfils none of the in Chapter 3 introduces requirements. For every of these requirements we want

to use one literal measure that fulfils exactly this requirement and at last the six special version of the literal measure that we defined in 4.1.1.

5.2 Attribute Exploration with ConExp

We now want to present the steps of the attribute exploration with *ConExp*. After setting up our starting data base we can simply start it with the *Start Attribute Exploration* button. The program systematically poses possible implications concluded from the database. We have to confirm those implications or provide a counterexample. Providing a counterexample updates the database and may lead to new conclusions. As counterexample we always try to provide one, that primary has the least properties from the conclusion of the posed implication and secondary the most of all other properties. This will spare us lots of implications, that are just slight variations.
For our presentation of the attribute exploration we show the implications in the order they are posed. To each implication they also show the counterexample we provided. If we found no counterexample, we state a short comment. The proofs for generality of confirmed implications will follow in the next section. The following table introduces acronyms for the formal properties.

Equivalence Invariant	*EqInv*
Equivalence Closed	*EqClo*
Subsumption Preserving	*SubPre*
Reverse Subsumption Preserving	*ReSubPre*
Dissimilar Closed	*DisClo*
Peserving	*Bou*
Structural Dependent for $\bigcirc$	*StrucDep*($\bigcirc$)
Triangle Inequality	*TriInEq*

Attribute Exploration Steps

(i) **statement:** $true \Longrightarrow EqInv + EqClo$

counterexample: the CSM presented in $ref\bigstar\ FD06$.

properties: *EqInv*,*SubPre*,*ReSubPre*

(ii) **statement:** $true \Longrightarrow EqInv$

counterexample: the CSM presented in $ref\bigstar\ Jan06$.

properties: *StrucDep*($\sqcap$)

(iii) **statement:** $StrucDep(\sqcup) \Longrightarrow EqInv + EqClo$

counterexample: a version of $simi_{sym}$ on a $\mathcal{DL}$with $const(\mathcal{DL}) = \{\sqcup, \sqcap\}$ and no unique normal form given.

properties: *StrucDep*($\sqcap$),*StrucDep*($\sqcup$),*Bou*,*DisClo*

(iv) **statement:** $ReSubPre \Longrightarrow EqInv$

confirmed: holds within this context. Our restriction to name unique normal forms cause, that CSMs preserving *ReSubPre* automatically preserve *EqInv*.

(v) **statement:** $SubPre \Longrightarrow EqInv$

confirmed: holds within this context. Our restriction to name unique normal forms cause, that CSMs preserving *SubPre* automatically preserve *EqInv*.

(vi) **statement:** $EqClo \Longrightarrow EqInv$

counterexample: just assume no unique normal form and distinguish whether or not both arguments are equivalent or not. If equivalent return *1*, if not calculate some value, that could not get *1*, except the arguments are equivalent. Note that because the Structural Dependencies are independent from *EqClo*, this CSM can fulfil *StrucDep* for both constructors.

properties: *EqClo*, *StrucDep(⊓)*, *StrucDep(⊔)*, *Bou*, *DisClo*

(vii) **statement:** $EqInv + StrucDep(\sqcup) \Longrightarrow EqClo$

counterexample: the CSM $simi_{sym,max(\sqcap),max(\sqcup)}$ on $\mathcal{DL}$ using the literal measure lm^{sym}_{Bou} and $const(\mathcal{DL}) = \{\sqcap, \sqcup\}$.

properties: *EqInv*, *StrucDep(⊓)*, *StrucDep(⊔)*, *Bou*, *DisClo*, *SubPre*, *ReSubPre*

(viii) **statement:** $EqInv + ReSubPre + StrucDep(\sqcap)$
$\Longrightarrow SubPre + DisClo + Peserving + StrucDep(\sqcup)$

counterexample: the CSM $simi_{sym,max(\sqcap)}$ on $\mathcal{DL}$ using a literal measure that is not *DisClo* and $const(\mathcal{DL}) = \{\sqcap\}$.

properties: *EqInv*, *StrucDep(⊓)*, *Bou*, *SubPre*, *ReSubPre*

(ix) **statement:** $EqInv + ReSubPre + StrucDep(\sqcap)$
$\Longrightarrow SubPre + Peserving$

counterexample: the CSM $simi_{sym,max(\sqcap),max(\sqcup)}$ on $\mathcal{DL}$ using the literal measure is not *Bou*.

properties: *EqInv*, *StrucDep(⊓)*, *StrucDep(⊔)*, *DisClo*, *SubPre*, *ReSubPre*

(x) **statement:** $EqInv + ReSubPre + StrucDep(\sqcap) \Longrightarrow SubPre$

confirmed: holds within this context. In our research we could not find a function, that preserves *ReSubPre* for ⊓ without automatically preserving *SubPre*.

formulierung überdenken, ist Struc dep nicht auhc ein einflussfaktor?

(xi) **statement:** $EqInv + SubPre + StrucDep(\sqcup)$
$\Longrightarrow ReSubPre + DisClo + StrucDep(\sqcap)$

counterexample: the CSM $simi_{sym,max(\sqcup)}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{Bou} and $const(\mathcal{DL}) = \{(\neg), \sqcup\}$.

properties: *EqInv*, *StrucDep(⊔)*, *Bou*, *SubPre*, *ReSubPre*

(xii) **statement:** $EqInv + SubPre + StrucDep(\sqcup) \Longrightarrow ReSubPre$

confirmed holds within this context. In our research we could not find a function, that preserves $SubPre$ for $\sqcup$ without automatically preserving $ReSubPre$.

(xiii) **statement:** $EqInv + SubPre + StrucDep(\sqcup) + StrucDep(\sqcap) \Longrightarrow DisClo$

counterexample: the CSM $simi_{sym,max(\sqcup),max(\sqcap)}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{Bou}.

properties: *EqInv*, *StrucDep(⊔)*, *StrucDep(⊓)*, *Bou*, *SubPre*, *ReSubPre*

(xiv) **statement:** $EqInv + SubPre + StrucDep(\sqcup) + DisClo \Longrightarrow StrucDep(\sqcap)$

counterexample: the CSM $simi_{sym,max(\sqcup)}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{Bou} and $const(\mathcal{DL}) = \{\sqcup\}$.

properties: *EqInv*, *StrucDep(⊔)*, *Bou*, *DisClo*, *SubPre*, *ReSubPre*

(xv) **statement:** $EqInv + SubPre + StrucDep(\sqcap) + DisClo \Longrightarrow StrucDep(\sqcup)$

counterexample: the CSM $simi_{sym,max(\sqcap)}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{DiCl} and $const(\mathcal{DL}) = \{\sqcap\}$.

properties: *EqInv*, *StrucDep(⊓)*, *Bou*, *DisClo*, *SubPre*, *ReSubPre*

(xvi) **statement:** $EqInv + EqClo + SubPre \Longrightarrow Peserving$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{def} and $const(\mathcal{DL}) = \{\sqcup\}$.

properties: *EqInv*, *EqClo*, *StrucDep(⊔)*, *DisClo*, *SubPre*, *ReSubPre*, *TriInEq*

(xvii) **statement:** $EqInv + EqClo + SubPre + StrucDep(\sqcap) \Longrightarrow Peserving$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{def}, an equivalence exclusive $\sqcup$-rule and $const(\mathcal{DL}) = \{\sqcup, \sqcap\}$.

properties: *EqInv*, *EqClo*, *StrucDep(⊓)*, *DisClo*, *SubPre*, *ReSubPre*, *TriInEq*

(xviii) **statement:** $EqInv + EqClo + SubPre + ReSubPre + StrucDep(\sqcup)$
$\Longrightarrow DisClo + TriInEq$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{def} and $const(\mathcal{DL}) = \{(\neg), \sqcap\}$.

properties: *EqInv*, *EqClo*, *StrucDep(⊔)*, *SubPre*, *ReSubPre*, *TriInEq*

(xix) **statement:** $EqInv + EqClo + SubPre + ReSubPre + StrucDep(\sqcup) \Longrightarrow TriInEq$

confirmed: holds within this context. In our research we could not find a function, that preserves $SubPre$ and $EqClo$ for $\sqcup$, without automatically preserve $ReSubPre$ and $TriInEq$.

(xx) **statement:** $EqInv + EqClo + SubPre + ReSubPre + StrucDep(\sqcap)$
$\Longrightarrow DisClo + TriInEq$

confirmed: holds within this context. In our research we could not find a function, that preserves $ReSubPre$, $EqClo$ and $StrucDep$ for $\sqcap$, without automatically preserve $SubPre$ and $TriInEq$. Additionally form being $ReSubPre$ and $StrucDep(\sqcap)$ follows $DisClo$.

(xxi) **statement:** $TriInEq \Longrightarrow EqInv + EqClo$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{DiCl}, $const(\mathcal{DL}) = \{\sqcap\}$ the $\sqcap$-rule modified to trivially fulfil $TriInEq$ without $EqClo$ and no assumed unique normal form properties: *DisClo*, *Bou*, *TriInEq*

(xxii) **statement:** $TriInEq + StrucDep(\sqcup)$
$\implies EqInv + EqClo + SubPre + ReSubPre$

confirmed: holds within this context. In our research we could not find a way to assure $TriInEq$ and $StrucDep$ other than breaking the functions down to the Jaccard Index

(xxiii) **statement:** $TriInEq + StrucDep(\sqcap)$
$\implies EqInv + EqClo + SubPre + ReSubPre + DisClo$

confirmed: holds within this context. In our research we could not find a way to assure $TriInEq$ and $StrucDep$ other than breaking the functions down to the Jaccard Index

(xxiv) **statement:** $TriInEq + EqClo \implies EqInv$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{def}, $const(\mathcal{DL}) = \{\sqcap\}$ the $\sqcap$-rule modified to trivially fulfil $TriInEq$ and no assumed unique normal form

properties: *EqClo*, *DisClo*, *Bou*, *TriInEq*

(xxv) **statement:** $TriInEq + EqInv \implies EqClo$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{DiCl}, $const(\mathcal{DL}) = \{\sqcap, \sqcup\}$, a normal form that is a conjuction of disjunctions, the $\sqcap$-rule modified to trivially fulfil $TriInEq$ without $EqClo$ and the $\sqcup$-rule being equivalence exclusive

properties: *EqInv*, *DisClo*, *Bou*, *TriInEq*, *SubPre*, *ReSubPre*,

(xxvi) **statement:** $TriInEq + EqInv + DisClo \implies SubPre + ReSubPre$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{DiCl}, $const(\mathcal{DL}) = \{\sqcap\}$ the $\sqcap$-rule modified to trivially fulfil $TriInEq$

properties: *EqInv*, *EqClo*, *DisClo*, *Bou*, *TriInEq*

(xxvii) **statement:** $TriInEq + EqInv + ReSubPre \implies SubPre$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{Bou}, $const(\mathcal{DL}) = \{\sqcup\}$ the $\sqcap$-rule modified to trivially fulfil $TriInEq$

properties: *EqInv*, *EqClo*, *DisClo*, *Bou*, *ReSubPre*, *TriInEq*

(xxviii) **statement:** $TriInEq + EqInv + SubPre \implies ReSubPre$

counterexample: the CSM $simi_{sym}$ on $\mathcal{DL}$ using a literal measure lm^{sym}_{DiCl}, $const(\mathcal{DL}) = \{\sqcap\}$ the $\sqcap$-rule modified to trivially fulfil $TriInEq$

properties: *EqInv*, *EqClo*, *DisClo*, *Bou*, *SubPre*, *TriInEq*

(xxix) **statement:** $TriInEq + EqInv + EqClo + SubPre + ReSubPre + Bou$
$+ StrucDep(\sqcup) \implies DisClo + StrucDep(\sqcap)$

confirmed: holds within this context. In our research we could not find a way to assure $TriInEq, SubPre, ReSubPre, Bou$ and $StrucDep$ for conjunction or disjunction at the same time.

(xxx) **statement:** $TriInEq + EqInv + EqClo + SubPre + ReSubPre + DisClo$
$+ StrucDep(\sqcup) + StrucDep(\sqcap) \implies Peserving$

confirmed: holds within this context. In our research we could not find a way to assure *SubPre*, *ReSubPre*, *EqClo* and *StrucDep* for conjunction and disjunction at the same time.

(xxxi) **statement:** $TriInEq + EqInv + EqClo + SubPre + ReSubPre + DisClo + Bou + StrucDep(\sqcap) \implies StrucDep(\sqcup)$

confirmed: holds within this context. In our research we could not find a way to assure *TriInEq*, *SubPre*, *ReSubPre*, *Bou* and *StrucDep* for conjunction or disjunction at the same time.

5.3 FCA Results

5.3.1 The Statements (iv) and (v)

To prove the statements from (iv) and (v), we will take advantage from our restriction to name unique (normal) forms. So for equivalent primitive concept names and equivalent role names, we have chosen one of them, to represent all its equivalents in all concept descriptions. This makes the CSM Equivalence Invariant in $N_{L_{pr}}$ and N_R. Expanding all concept description with the knowledge of $\mathcal{T}$ and $\mathcal{R}$ additionally assures, that only primitive literal occur in them.
Before we state theorem and proof, we need the following preparations. First we show, that $\top$ can not violate Equivalence Closed while preserving Subsumption Preserving and $\bot$ can not violate Equivalence Closed while preserving Reverse Subsumpion Preserving.

Lemma 9. *Let sim be a symmetric deterministic CSM on a specific $\mathcal{DL}$, using a name unique normal form. Then $\top$ can not violate Equivalence Invariant while preserving Subsumption Preserving and $\bot$ can not violate Equivalence Invariant while preserving Reverse Subsumpion Preserving.*

Proof 20 (Lemma 9)**.**

The only way for $\top$ and $\bot$ to violate Equivalence Invariant are alternative representations like $A \in \mathcal{C}(\mathcal{DL}) : \top \equiv A \sqcup \neg A, \bot \equiv A \sqcap \neg A$. So for $\top$ there must be $\{\neg, \sqcup\} \subseteq const(\mathcal{DL})$ and for $\bot$ there must be $\{\neg, \sqcap\} \subseteq const(\mathcal{DL})$. Since equivalence implies subumption, we can state for all $A, B, C \in \mathcal{C}(\mathcal{DL}), A, B, C \not\equiv \top$:

$$(A \sqcup \neg A) \sqsupseteq (B \sqcup \neg B) \sqsupseteq C$$
$$(B \sqcup \neg B) \sqsupseteq (A \sqcup \neg A) \sqsupseteq C$$

Thus the similarities from all representations of $\top$ to all concept descriptions should be the same, or Subsumption Preserving is violated. The same holds for $C \equiv \top$, since then it also can be flipped in the subsumption chain.
For $\bot$ we can state for all $A, B, C \in \mathcal{C}(\mathcal{DL}), A, B, C \not\equiv \bot$:

$$C \sqsupseteq (A \sqcap \neg A) \sqsupseteq (B \sqcap \neg B)$$
$$C \sqsupseteq (B \sqcap \neg B) \sqsupseteq (A \sqcap \neg A)$$

and do the same Argumentation for Reverse Subsumpion Preserving.

$\square$

This allows us to show, that if a conjunction or disjunction C specifically violates Equivalence Invariant, for the needed cases, there exist a subsuming and a subsumed one, that have different similarities to the equivalent representations.

Lemma 10. *Let sim be a symmetric deterministic CSM on a specific $\mathcal{DL}$, using a name unique normal form, $\bigcirc \in \{\sqcap, \sqcup\}$, $\bigcirc \in const(\mathcal{DL})$ and $C = \bigcirc_i C_i, i = 1, ..., n, n > 1$ with none of the C_i violating Equivalence Invariant. If C violates Equivalence Invariant, we want to say, that C violates this property* initially. *For $C \not\equiv \top$, $C \not\equiv \bot$ and C violating Equivalence Invariant initially, there holds, that:*

$$\begin{aligned} \exists C', D, E : & E \sqsupseteq C \sqsupseteq D, C \equiv C' \\ & sim(C, E) \neq sim(C', E) \\ & sim(C, D) \neq sim(C', D) \end{aligned}$$

Proof 21 (Lemma 10)**.**

The existence of C' is trivially given by C violating Equivalence Invariant. For the conjunction, an E can be constructed by dropping one of the C_i from C. Since $n > 1$, this is possible. A D can be received from disjunctive connecting an $A \not\sqsubseteq C$ to C. Since $C \not\equiv \bot$, such an A exists in $\mathcal{C}(\mathcal{DL})$.
For the disjunction, an D can be constructed by dropping one of the C_i from C. Since $n > 1$, this is possible. A E can be received from conjunctive connecting an $A \not\sqsubseteq C$ to C. Since $C \not\equiv \top$, such an A exists in $\mathcal{C}(\mathcal{DL})$.
Since C initially violates Equivalence Invariant, the syntax of C must be relevant for calculating the similarity with the corresponding $\bigcirc$-rule of *sim*. Because *sim* is symmetric, this implies that also the syntax of the other concept is relevant. So for the constructed E and D holds: $sim(C, E) \neq sim(C', E)$ and $sim(C, D) \neq sim(C', D)$.

$\square$

We also want to introduce the value $\overline{x}_{sim}^{C,D}$. It will represent the highest achievable similarity, by a CSM *sim* and equivalent versions of C to D.

Definition 50. *For sim being a symmetric concept similarity measure on a specific description logic $\mathcal{DL}$, $C, C', D \in \mathcal{C}(\mathcal{DL})$ we define:*

$$\overline{x}_{sim}^{C,D} = max\{x | C' \equiv C, x = sim(C', D)\}$$

With this preparations we can state the following theorem and prove it.

Theorem 13. *Let sim be a deterministic symmetric concept similarity measure using a name unique normal form. Them holds, that if sim is Subsumption Preserving, sim is also Equivalence Invariant.*

Proof 22 (Theorem 13)**.**

Let *sim* be on a specific DL $\mathcal{DL}$. Assume *sim* is Subsumption Preserving but not Equivalence Invariant. This means there exists $C, C', D \in \mathcal{C}(\mathcal{DL})$ with $C \equiv C'$ and $sim(C, D) \neq sim(C', D)$. We want C and C' to be the concept description, violating Equivalence Invariant initially and not just by containing an other concept description that already violates Equivalence Invariant. Since *simi* use a name unique normal form, *sim* is Equivalence Invariant in $N_{L_{pr}}$ and N_R. This means C and C' can not be primitive literals or quantifications. So they must be conjunctions or disjunctions with at least two participants.
So $\bigcirc \in \{\sqcap, \sqcup\}$, $C = \bigcirc_i C_i, i = 1, ..., n, n > 1$ with none of the C_i violating Equivalence Closed. By Lemma 10 for the conjunction exists a E with $E \sqsubseteq C$ and $sim(C, E) \neq sim(C', E)$. Since equivalence implies subsumption, we can argue:

$$C' \sqsupseteq C \sqsupseteq E$$
$$C \sqsupseteq C' \sqsupseteq E$$

So in one or the other way, Subsumption Preserving is violated.
By Lemma 10 for the disjunction exists a E with $E \sqsubseteq C$, $E \not\equiv \top$ and $sim(C, E) \neq sim(C', E)$. Lemma 9 states, that therefore C not have to be equivalent to $\top$. Since equivalence implies subsumption, we can argue:

$$C' \sqsupseteq C \sqsupseteq E$$
$$C \sqsupseteq C' \sqsupseteq E$$

So in one or the other way, Subsumption Preserving is violated.

Since in both cases, Subsumption Preserving is violated when violating Equivalence Closed, we can conclude, that Subsumption Preservingd implies Equivalence Close.

□

Lemma 9 and 10 state enough to adapt Theorem 13 to Reverse Subsumption Preserving. So we can conclude the following corollary.

Corollary 21. *Let sim be a deterministic symmetric concept similarity measure using a name unique normal form. Them holds, that if sim is Reverse Subsumption Preserving, sim is also Equivalence Invariant.*

5.3.2 The Statements (x), (xii), (xix), (xx), (xxii) and (xxiii)

These statements follow from our rule-functions just insufficiently preserving Triangle Inequality, Subsumption Preserving and Reverse Subsumption Preserving. As said, the Structural dependence for the conjunction needs the conjunction to be a constructor in the used DL $\mathcal{DL}$. So statement (x) holds within our context, because the only way to make the conjunction $\sqcap$ Reverse Subsumtion Preserving, are trivial rule function, like *max* and the equivalence exclusive rule, or breaking the calculation down to the Jaccard Index. The latter is, what we do with our

restrictions to $simi_{asym}$ and $simi_{sym}$ in 3.2.2. But this methods automatically preserve Subsumption Preserving for $\sqcap$. This is be because of the semantic relation between $\sqcap$ and subsumption.
We want to take a look at a CSM $simi_{rstr}$ on a DL $\mathcal{DL}$with the assumed restrictions that $const(\mathcal{DL}) \cap \{\neg, \sqcup, \exists, \forall\} = \varnothing$ and for all concept descriptions $D \in \mathcal{C}(\mathcal{DL})$ holds $simi_{rstr}(D, D) = 1$. Further let:

$$A \equiv A_1 \sqcap \ldots \sqcap A_n$$
$$B \equiv B_1 \sqcap \ldots \sqcap B_m$$
$$C \equiv C_1 \sqcap \ldots \sqcap C_l$$

with $A \sqsupseteq B \sqsupseteq C$, so there holds that for every conjunction participant $A_i, i \in \{1, ..., n\}$ there exists a $B_j, j \in \{1, ..., m\}$ with $A_i \sqsupseteq B_j$, for every $B_i, i \in \{1, ..., m\}$ there exists a $C_j, j \in \{1, ..., l\}$ with $B_i \sqsupseteq C_j$ and consequently the same for $A_i, i \in \{1, ..., n\}$ subsuming some $C_j, j \in \{1, ..., l\}$. Equivalence Invariant and the restriction to $\mathcal{DL}$ in this case assure, that subsumption between the conjunction participants of the three concept descriptions can only be by equivalence. Since Equivalence Invariant allows us to rearrange nested conjunctions to one conjunction without changing the similarity values and besides the conjunction $\sqcap$ only the primitive negation $(\neg)$ is as possible constructor, but not able to express a subsumption, all these conjunction participants have to be in $N_{L_{pr}}$. Together with the just established subsumption condition for conjunctions we can conclude that $\widehat{C}_\sqcap \supseteq \widehat{B}_\sqcap \supseteq \widehat{A}_\sqcap$. Because we can state $A \sqsupseteq A \sqsupseteq B$, we can conclude that the conjunction participants form $\widehat{B}_\sqcap / \widehat{A}_\sqcap$ influence $simi_{rstr}(A, B)$ by no change or reducing it in comparison to $simi_{rstr}(A, A)$. In fact, if there would exist a $B_i \in \widehat{B}_\sqcap / \widehat{A}_\sqcap$ that would have an in any way increasing influence to $simi_{rstr}(A, B)$, then we could say that by leaving out this B_i in a DL:

$$A \sqsupseteq \textstyle\bigsqcap_{j \in S_j} B_j \sqsupset B, S_j = \{1, ..., m\}/i$$

there must hold $simi_{rstr}(A, B) > simi_{rstr}(A, \bigsqcap_{j \in S_j} B_j)$, what would contradict Reverse Subsumption Preserving. So the addition of a semantically relevant conjunction participant to a concept description D results in a new concept description D' that has a equal or lower similarity to all concept description subsuming both of them. From this follows $simi_{rstr}(A, C) \leq simi_{rstr}(A, A)$ and $simi_{rstr}(B, C) \leq simi_{rstr}(B, B)$, because C is just A or B with additional conjunction participants. Also by assumption holds that $simi_{rstr}(A, A) = simi_{rstr}(B, B) = simi_{rstr}(C, C) = 1$. Now we can say that removing a semantically relevant conjunction participant from a concept description D results in a new concept description D' that has a equal or lower similarity to all concept description that are subsumed by both of them.
But unfortunately we can not argue this way, when we allow other constructors in the DL. This allows subsumption to be realised in other way than equivalence. While the simple subsumption between one A_i and one B_j may be easily handled by assuming the conjunction participants already fulfil statement (x) and

making a proof over the structure of $\mathcal{DL}$, a more serious problem are **multi-subsumptions**. As multi-subsumptions we want to see cases, where either one conjunction participant subsumes or is subsumed by multiple conjunction participants of the other concept description. Such multi-subsumptions enable much bigger varieties of possible structures of the descriptions A, B and C. Unfortunately within the research of this paper these varieties were not invested enough to bring adequate results in either proving or disproving statement (x) for general.

Note that our restriction in 3.2.2 to make the conjunction $\sqcap$ Subsumption Preserving in $simi_{sym}$ or $simi_{dual}$ is actually that subsumption is only possible by equivalence. The same problem with multi-subsumptions occurs for statement (xii). In fact there it is a kind of dual case with $\sqcup$ Reverse Subsumption Preserving. Also statements (xix) and (xx) hold within this context because of this problem. Without further investigations of the cases with multi-subsumption, the only reliable method to ensure Subsumption Preserving, Reverse Subsumption Preserving and Equivalence Closed while containing $\sqcap$ or $\sqcup$ in the constructor set is breaking the calculation down to the Jaccard Index. But this also automatically ensures the Triangle Inequality. The statements (xxii) and (xxiii) hold within this context, because we could use other ways to ensure the the Triangle Inequality, than is breaking the calculation down to the Jaccard Index or the method to trivially achieve Triangle Inequality (see 4.4). Because the second method do not preserve Structural Dependence, this implies that by the state of the researches the premise of (xxii) and (xxiii) can only be achieved by breaking the calculation down to the Jaccard Index. And this automatically ensures the conclusion of these statements.

5.3.3 The Statements (xxix), (xxx) and (xxxi)

The premisses of these statements are not achievable within the research of this paper, so at this state these statements hold trivially. The premisses of (xxix) and (xxxi) are not achievable because we have no other way to fulfil the Triangle Inequality and Structural Dependence is breaking the calculation down to the Jaccard Index. Unfortunately for (xxix), Structural Dependence for the disjunction requires it to be in the constructor set of the DL, what makes it impossible to be Peserving at the same time as fulfilling the Triangle Inequality. For the premiss of (xxxi), we need additionally Peserving and Dissimilar Closed to be fulfilled at the same time. Therefore the conjunction has to be in the constructor set of the DL. But now we have both conjunction and disjunction in the constructor set. So to obtain Subsumption Preserving and Reverse Subsumption Preserving at the same time, one of this two constructors have to use an equivalence exclusive rule, what results in Peserving being violated. This same problem makes it impossible to fulfil the premiss of (xxx).

So for now this statements may serve no purpose nor information, but the should still be mentioned, because further research in the properties Subsumption Preserving, Reverse Subsumption Preserving and the fulfilling of the Triangle Inequality may allow in the future explorations starting from these statements.

6 Conclusion

We extended *simi* from [4] in three different ways. One ($simi_{asym}$) is a direct extension of *simi* by just adding rules for $(\neg)$, $\sqcup$ and $\forall$, which for the most cases preserves *simi*'s design to fulfil Subsumption Preserving. The second one ($simi_{dual}$) is a dual construction to $simi_{asym}$, which changes the design around to fulfil Reverse Subsumption Preserving for the most cases. The third extension ($simi_{sym}$) has a fully symmetric design, which gives disadvantages in fulfilling Subsumption Preserving and Reverse Subsumption Preserving, but enables an easier fulfilling of the Triangle Inequality.

For every of this extensions we investigated under which circumstances the rules of the constructor provide the different formal properties. With this knowledge an additional modification for the rule, we achieved a huge number of CSMs and additionally the formal properties they fulfil. Used as a database for a FCA, we were able to conclude several implications, that hold within the context of our expansions and their modifications.

Also we could only prove two of them for generality, the context still is a part of the universality. So implications that are disproved in this context are also disproved for the generality.

6.1 Open Problems

A major task for future work is to do further researches of the in 5.3.2 mentioned multi-subsumptions. An deeper investigation of the behaviour of CSM if these multi-subsumptions are enabled could bring more detailed answers to the question, how to achieve Subsumption Preserving or Reverse Subsumption Preserving. Also results for the Triangle Inequality may be achieved by this, because multi-subsumptions also cause problems in these field.

The Triangle Inequality remains an open problem. Since our trivial rule to achieve it is not a suitable way, the Jaccard Index remains the best attempt. Instead of its calculation by concept names or participants of an constructor, an other approach is to adapt it to concept description could be equivalence classes. We could associate concept descriptions with sets of equivalence classes. In the calculation we compare the number of equivalence classes they have in common with the number of equivalence classes they have together.

Also the negation remains an open problem. As mentioned in 3.1 we could use the primitive negation and a normal form that only allows negations in front of primitive concepts, but the exponential complexity to achieve such a normal form for $\mathcal{ALC}$is a big disadvantage. Besides achieving Equivalence Invariant for equivalent phrasings of $\sqcup$ and $\sqcap$ terms with negation would be of interest.

Of course, with the progress of the already mentioned open problem, the FCA of this paper could be completed. Further more the investigation may be expanded to asymmetric CSM or the usage of weighting functions other than $g_{\mathcal{DL},def}$.

The investigation of non-deterministic CSM or DLs that do not use a name unique (normal) form may also be of interest. But because of the unpredictable nature of those, a practical application seems unlikely.

Bibliography

[1] Franz Baader. Description Logics. 2009.

[2] Rudolf Wille Bernhard Ganter. ”Formal Concept Analysis - Mathematical Foundations”,. 1999.

[3] Paul Jaccard. Etude comparative de la distribution orale dans une portion des alpes et des jura. 1901.

[4] Karsten Lehmann. A Framework for Semantic Invariant Similarity Measures for $\mathcal{ELH}$ Concept Descriptions. 07.

[5] A. H. Lipkus. A proof of the triangle inequality for the tanimoto distance. 1999.

[6] Serhiy A. Yevtushenko. System of data analysis ”Concept Explorer”. (In Russian). Proceedings of the 7th national conference on Artificial Intelligence KII-2000,.