STRUCTURED ANALOGICAL ARGUMENTATION: FRAMEWORKS AND ALGORITHMS

BY

MR. TEERADAJ RACHARAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (ENGINEERING AND TECHNOLOGY)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2018

STRUCTURED ANALOGICAL ARGUMENTATION: FRAMEWORKS AND ALGORITHMS

BY

MR. TEERADAJ RACHARAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR
OF PHILOSOPHY (ENGINEERING AND TECHNOLOGY)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2018

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

DISSERTATION

BY

MR. TEERADAJ RACHARAK

ENTITLED

STRUCTURED ANALOGICAL ARGUMENTATION:
FRAMEWORKS AND ALGORITHMS

was approved as partial fulfillment of the requirements for
the degree of Doctor of Philosophy (Engineering and Technology)

on July 26, 2019

Chairman
(Associate Professor Ekawit Nantajeewarawat, D.Eng.)

Member and Advisor
(Assistant Professor Nguyen Duy Hung, Ph.D.)

Member and Co-advisor
(Professor Satoshi Tojo, Ph.D.)

Member
(Professor Thanaruk Theeramunkong, D.Eng.)

Member
(Prachya Boonkwan, Ph.D.)

Member
(Boontawee Suntisrivaraporn, D.Eng.)

Director
(Professor Pruettha Nanakorn, D.Eng.)

Thesis Title	STRUCTURED ANALOGICAL ARGUMENTATION: FRAMEWORKS AND ALGORITHMS
Author	Mr. Teeradaj Racharak
Degree	Doctor of Philosophy (Engineering and Technology)
Faculty/University	Sirindhorn International Institute of Technology/ Thammasat University
Thesis Advisor	Assistant Professor Nguyen Duy Hung, Ph.D.
Thesis Co-Advisor	Professor Satoshi Tojo, Ph.D.
Academic Years	2018

ABSTRACT

Analogical reasoning is a complex process based on a comparison between two pairs of concepts and states of affairs (*aka.* the source and the target) for characterizing certain features from one to another. Arguments which employ this process to support their claims are called *analogical arguments*. Our goals are to study the structure and the computation for their defeasibility in light of the argumentation theory. We outline the results of our study as comprising two parts in the following.

First, analogical reasoning involves in understanding the notion of similarity. To address this problem, we first take a look into the literature of similarity models. The most basic (but useful) one was introduced by (Tversky, 1977). In Tversky's model, an object is considered as a set of features. Then, the similarity of two objects is measured by the relationship between a number of common features and a number of different features. Nonetheless, not every feature need to be cited in analogical arguments, the studies in (Hesse, 1965; Waller, 2001; Weinreb, 2016) reported that features used by the comparison should be 'relevant' for characterizing certain features from one to another. As part of the study, we formally investigate the characteristics of a 'similarity notion' for analogical arguments in this dissertation. Though our similarity

models are posed in a general structure, our running examples are shown w.r.t. description logic formalism.

Second, analogical reasoning involves in understanding the structure of analogical arguments and computing their nature of defeasibility. These problems are indeed related to the study of structured argumentation and the acceptability of analogical arguments. As their results, we formally introduce a general framework in structured argumentation called *assumption-based argumentation with predicate similarity* $ABA^{(p)}$ framework. This framework can be seen as an extension of assumption-based argumentation framework (ABA), in which not only assumptions can be used but also similarity of predicates (w.r.t. by the proposed similarity notion) are used to support a claim. $ABA^{(p)}$ labels each argument tree with an analogical degree and different ways to aggregate numerical values are studied toward gullible/skeptical characteristics in agent reasoning. The acceptability of analogical arguments is evaluated w.r.t. Dung-styled semantics.

Finally, we demonstrate how our study can benefit the area of service science. Realistic examples are analyzed and a diverse range of applications is discussed.

Keywords: Analogical Reasoning, Metaphorical Reasoning, Description Logic, Assumption-based Argumentation, Persuasive Reasoning

ACKNOWLEDGEMENTS

First and Foremost, I would like to express my deepest gratitude to my advisor and co-advisor, Asst. Prof. Nguyen Duy Hung and Prof. Satoshi Tojo, for their guidance, support, patience, and encouragement throughout my doctoral course. Their technical and editorial advice were essential to the completion of this thesis. They have taught me innumerable lessons on academic researching. I am and will always be most appreciative of their kindness to help me learn, grow, and move forward.

Grateful acknowledgement is extended to Dr. Boontawee Suntisrivaraporn, who serve as a member of doctoral committees and my former advisor, for his useful and valuable guidance at the early stage of my doctoral study. Cordial gratitude is continued to Dr. Prachya Boonkwan for his kind proofreading on several of our publications. His suggestions and correction of writing style have improved my writing skill. Many thanks to Dr. Prachya Boonkwan, Assoc. Prof. Ekawit Nantajeewarawat, and Prof. Thanaruk Theeramunkong, who also serve as members of doctoral committees for their interest, comments, and support throughout the study. In addition, I would like to thank Prof. Ken Satoh, my external examiner, for his constructive suggestions.

I am also obliged to JAIST-NECTEC-SIIT dual degree program for the scholarship which facilitated my doctoral study at Sirindhorn International Institute of Technology, Thammasat University.

Last but not least, I would like to express my warmest appreciation to my family, Mr. Boontham Racharak and Mrs. Supatra Racharak, for their love, care, patience, understanding, and spiritual supports.

Mr. Teeradaj Racharak

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
$\mathcal{A}$	An ABox
AA	Abstract Argumentation
ABA	Assumption-based Argumentation
$\mathcal{AL}$	Attributive Language, a Description Logic providing atomic concept, the top concept $(\top)$, the bottom concept $(\bot)$, atomic negation, conjunction $(\sqcap)$, value restriction $(\forall r.C)$, and limited existential restriction $(\exists r.\top)$
$\mathcal{ALC}$	$\mathcal{AL}$ extended with full concept negation $(\mathcal{C})$
CQ	Critical Question
DL	Description Logic
DLs	Description Logics
$\mathcal{FL}_0$	$\mathcal{AL}$ disallowing the bottom concept $(\bot)$, atomic negation, and limited existential restriction $(\exists r.\top)$
$\mathcal{EL}$	$\mathcal{AL}$ disallowing the bottom concept $(\bot)$, atomic negation, and value restriction $(\forall r.C)$
$\mathcal{ELH}$	$\mathcal{EL}$ extended with role hierarchy $(\mathcal{H})$
CN	A set of concept names
PAF	Preference-based Argumentation Framework
RN	A set of role names
$\mathcal{T}$	A TBox

CHAPTER 1
INTRODUCTION

1.1 Argumentation by Analogy

The word 'Analogy' (or, 'according to ratio' in Greek) originally meant rational correspondence (Macagno, 2014). In the Posterior Analytics[1], Aristotle pointed out that this type of reasoning could be used for identifying a fundamental characteristic common to various entities, and for which no name exists. His statement is quoted as follows:

"Again, another way is excerpting in virtue of analogy; for you cannot get one identical thing which pounce and spine and bone should be called; but there will be things that follow them too, as though there were some single nature of this sort."

Intuitively, Aristotle noticed that there is not a specific generic class that subsumes the pounce (of a cuttlefish), the spine (of a fish), and the bone (of an animal). That is, no name representing this category existed. However, these three different concepts share substantial characteristics in common. Analogy, in this sense, can be used to reveal a genus (*i.e.* a generic, common, and relevant feature) that can be considered as an ontological and semantic property that does not have a conventional name (Glucksberg & Keysar, 1990; Hesse, 1965).

Argumentation by analogy (or analogical reasoning) are powerful cognitive tools, in a sense that enabling to deal with unfamiliar situations, and can be classified in various ways. For example, (Garssen, 2009, p. 134) classified that there are two variants of argumentation by analogy represented in the various argumentative patterns *viz.* descriptive analogy and normative analogy. In (Bermejo-Luque, 2014, p. 58), analogical reasoning is characterized based on two perspectives *viz.* qualitative analogy and quantitative analogy. Other proposed models can be found in (Copi, Cohen, & McMahon, 2016; Davies, 1988; Guarini, Butchart, Smith, & Moldovan, 2009; Walton, 2010; Walton, Reed, & Macagno, 2008).

---

[1] The Posterior Analytics is a text from Aristotle's Organon that deals with demonstration, definition, and scientific knowledge *i.e.* syllogisms of scientific knowledge and statements of things' nature.

Despite the diversity of existing models, analogical reasoning can be considered as a process based on a comparison between two pairs of concepts or states of affairs (*aka.* the *source* and the *target*) sharing some common features (Bartha, 2010). This comparison is the ground of this specific type of the reasoning, in which the conclusion of an argument is attributed to a specific feature characterized from one to another. The goal of this thesis is to study and investigate its computational aspect, particularly in light of the argumentation theory, rather than the psychological modeling.

Table 1.1 Argumentation by Analogy as An Extrapolation.

Description mentioned in the source	Description mentioned in the target
Relevant property 1	Relevant property 1
Relevant property 2	Relevant property 2
Relevant property 3	Extrapolated relevant property 3

In the following subsections, each variation of argumentation by analogy is exemplified and discussed w.r.t its usage pattern in an argumentative discourse.

1.1.1 Descriptive Analogy

A typical characteristic of *descriptive* analogical reasoning is that the conclusion that is defended and the reason that is advanced in its support are descriptive. In other words, a comparison in this kind of analogy is made between the actual characteristics of one concept or state of affairs and the actual characteristics of another concept or state of affairs. For instance,

"camera surveillance in the centre of Amsterdam will be effective because camera surveillance proved to be effective in London" in (Van Eemeren & Garssen, 2014, p. 48)

In this argumentation, London and Amsterdam are compared and it is claimed that 'camera surveillance' will be the case in Amsterdam because such thing has already happened in London and Amsterdam is comparable to London.

It is worth observing that similarity is often mentioned implicitly and is usually subjective to certain 'relevance' in argumentation by analogy. The above example may assume that there are a number of similarities relevant to 'safety' between Amsterdam and London. The fact that they are both capitals may be not relevant here since this condition is not directly related to safety in the streets. Instead, the fact that they are big cities is a relevant one to conclude that the property mentioned in London *i.e.* camera surveillance is effective is shared to Amsterdam.

Intuitively, descriptive analogy is often employed to extrapolate a property from commonalities between two concepts or states of affairs. In other words, when the source and the target are comparable and have properties shared in common, they are assumed to share another property mentioned in the target. Table 1.1 shows a general characterization for this kind of analogical reasoning. According to this characterization, this kind of analogical argument is alternatively called *case-based reasoning*.

To evaluate the acceptability of an extrapolated property, the first step is to ask whether the two concepts or states of affairs are comparable. If they are comparable, they are obliged to show that they are indeed belonged to the same class. This can be asked by the antagonist to the protagonist for mentioning the relevant properties which enable the conclusion that they are belonged to the same class. Mentioning the relevant properties can again enable the antagonist to criticize the perception of similarity since it may not be able to recognized as similarity or the mentioned properties may be not relevant to the issue at hand.

Not merely criticizing the mentioned relevant properties, the antagonist can also point out the differences between them. This leads the protagonist to show either these differences are not relevant or the mentioned properties for similarity outweigh the differences. Regarding this testing procedure, the acceptability of a claim can be decided if an extrapolation is successful nor not.

1.1.2 Normative Analogy

In *normative* analogy, the principle of consistency plays a central role. This can be seen in the *rule of justice i.e.* people and institutions which belong to the same category should be treated similarly. Like the descriptive analogy, this kind of argumentation by analogy is used to claim that what is mentioned in the target case is comparable to what is mentioned in the source case. For instance, *"the employees in the administration department should get a salary raise because the sale persons in our firm also get a salary raise"* in (Van Eemeren & Garssen, 2014, p. 48)

It is worth observing that normative analogy differs from descriptive analogy in a sense that the use of the principle of consistency does not involve an extrapolation of characteristics. Its central issue is whether two persons or groups of people are really belonged to the same class or not.

Another difference is the fact that a claim in normative analogical argumentation is 'normative' in nature *i.e.* it claims that people in the same category should be treated in the same direction.

Similar to the usage of descriptive analogical argumentation, a set of critical questions can be employed to evaluate a claim. Again, the first step is to ask whether a person or a group of people mentioned in the target case and another person or group of people mentioned in the source case are comparable. After this question has been answered by the protagonist, the antagonist can further ask the protagonist to justify that the two persons or groups of people being compared are really belonged to the same category. To that effect, the protagonist is forced to show additional argumentation. If the antagonist points at differences *i.e.* showing that the two persons or groups of people are not belonged to the same class, the protagonist's response has to show that the differences are not relevant or outweigh the similarities.

Normative analogical argumentation shares some characteristics with a specific pattern of reasoning introduced by Govier in 1987 called a *priori* reasoning (Gover, 2018). In a priori reasoning, some person, group or institution must act consistently: "You should do X because, in a similar situation, you would also do X. However, a priori reasoning differs from normative analogical reasoning in a sense that an example used in the comparison does not necessarily exist. Its basic idea is that if it is admitted in the answer that a person – imaginary or not – would be treated in a certain way, it has to be accepted that the (real) person of the same category must be treated in the same way.

1.1.3 Figurative Analogy

In figurative analogy, two different concepts or states of affairs are situated on different levels of experience (or are belonged to completely different kinds). An example was analyzed in (Hastings, 1963), in which President Truman puts forward at the beginning of the Korea conflict in defense of his claim that the United States should strike immediately as follows:

"The best time to meet the threat is in the beginning. It is easier to put out a fire in the beginning when it is small than after it has become a roaring blaze" in (Hastings, 1963, p. 114)

Fire and war are belonged to different classes, which make them impossible to compare directly between each other. By viewing literally, President Truman does not make a direct comparison between war and fire. To deal with this kind of comparison, the abstract relationship between the two classes is taken into account. Like the above example, the president's intention was to convey that the war in Korea would become unmanageable if we did not act now. This relationship was later characterized by (Perelman, 1969) as the 'resemblance of structures' as follows:

A and *B* together, the terms to which the conclusion relates are called *theme*.

C and *D* together, the terms that advance it are called *phoros*.

We schematize the above example according to Perelman as follows:

Theme (A - B): meeting the threat -- in the beginning

Phoros (C - D): putting out a fire -- when it is small

Intuitively, this schematization speaks out that only one similarity is shared between two situations. Its objective is to establish the general rule governing the situations *e.g.* the rule "it is suggested to approach a problem when it is small" is applied in the above example. After this general rule is constructed, it does not make sense for figurative analogy to search for additional similarities. This makes it differs from both descriptive analogical argumentation and normative analogical argumentation, which often imply that there are a number of similarities.

It is worth mentioning that figurative analogical argumentation is based on 'metaphorical' relation that serves as an indirect means of expressing a general rule for advancing a claim. Ones may also observe that no real comparison has been made in an idiomatic expression. Since an idiomatic expression is always used to convey implicitly intended meaning, it is ludicrous to ask critical questions in order to evaluate the acceptability of a claim advanced by a figurative analogy.

Using an idiomatic expression to advance a claim does not mean that this kind of analogical argumentation is logically weak. It might mean so if it is compared with other kinds of analogical argumentation. Nonetheless, when it is properly constructed, what is presented in a figurative analogy can be compelling argumentation in a different way. This also means that, when studying about analogical argumentation, figurative analogy should be treated in a different way from descriptive analogy and normative analogy. In this thesis, we 'only' concentrate on argumentative patterns containing descriptive analogy and normative analogy.

1.2 Argumentation Scheme for Argument from Analogy

Though many characterizations of argument from analogy have been proposed, they can be represented by a generic structure called argumentation scheme for argument from analogy introduced by (Walton et al., 2008) as follows:

Similarity Premise: Generally, case C_1 is similar to case C_2

Base Premise: A is true (false) in case C_1

Conclusion: A is true (false) in case C_2

This generic structure can be explained as follows. The similarity is regarded to hold between two cases. These cases could be two different 'concepts' or 'states of affairs'. Consequently, a property (*e.g.* a feature A) attributes from one to another. Intuitively, this kind of structure can be represented as a logic program where A and C_i are appeared as the head and the body of an inference rule, respectively. Several attempts similar to this approach were developed in (Racharak, Tojo, Hung, & Boonkwan, 2017a, 2017b; Raha, Hossain, & Ghosh, 2008; Sun, 1995a).

A fundamental problem for this kind of reasoning is how to evaluate an analogical argument, *i.e.* its acceptability. Basically, this problem amounts to investigations of the structure of analogical arguments and its defeasibility characteristics. At the abstract level, critical questions (CQ) (Walton et al., 2008) associated to the argument scheme outlines several conditions of defeasibility:

CQ1: Is A true (false) in C_1?

CQ2: Are C_1 and C_2 similar in the respects cited?

CQ3: Are there important differences (dissimilarities) between C_1 and C_2

CQ4: Is there some other case C_3 that is also similar to C_1 except that A is false (true) in C_3?

1.2.1 Three Issues in Walton's Scheme

Though the critical questions can be used to understand which analogical arguments should not be accepted. However, they do not address the following three basic problems:

1. How similarity/dissimilarity should be determined (which amounts to understand the notion of similarity)?;
2. How an analogical argument is constructed (which amounts to understand the structure of an analogical argument)?; and
3. How a conclusion drawn from the similarity premise and the base premise is warranted (which amounts to understand the evaluation of an analogical argument).

The argumentation scheme and its critical questions do not involve these aspects concretely.

To address the first problem, we first take a look into the literature of similarity models. The most basic (but useful) one was developed by (Tversky, 1977). In Tversky's model, an object is considered as a set of features. Then, the similarity of two objects is measured by the relationship between a number of common features and a number of different features. Nevertheless, not every feature need to be cited in analogical arguments, the studies in (Hesse, 1965; Waller, 2001; Weinreb, 2016) reported that features used by the comparison should be 'relevant' to the attribution of the property. This leads to our study on characteristics of similarity models for analogical arguments in this work.

Addressing the second and the third problems involve in computing arguments in terms of argumentation with structure (or *structured argumentation*). It should be noted that argumentation (Dung, 1995) is proven to be a promising platform to understand a non-monotonic and defeasible reasoning. With this viewpoint, these problems are indeed the problems of determining 'acceptable' analogical arguments w.r.t. argumentation semantics. That is, analogical arguments can attack (and be attacked by) other arguments. We show the correspondence between this attack-counterattack relationship and the defeasibility conditions of the argumentation scheme in this thesis.

1.3 Objectives

The primary objective of the thesis is to provide well understanding on the computational aspect of analogical reasoning in argumentation, rather than the psychological modeling, for dealing with the aforementioned problems. This goal is further developed into the following objectives:

1. To propose well-defined notions of similarity measure for concepts, particularly description logic concepts. These well-defined notions can be divided into two parts *viz.* the basic notion of concept similarity and its extension for similarity of concepts under preference context;
2. To investigate a general framework for analogical argumentation. Here, our development is restricted on assumption-based argumentation framework (ABA) (Dung, Kowalski, & Toni, 2009), which is a less abstract framework than the abstract argumentation (AA) (Dung, 1995);
3. To demonstrate potential applications of our proposed methods.

To fulfill these objectives, this thesis makes the following main contributions:

1. The well-defined notion of concept similarity measure in description logics, which is defined as a function mapping from a concept pair to a unit interval $(0 \leq x \leq 1)$ for any real number x, also, a group of identified preferential aspects (called preference context), which can together be used to define the notion *concept similarity under preferences* (*cf.* Chapter 3);
2. The well-defined concrete measure sim^{π} for the description logic $\mathcal{ELH}$ and mathematical proofs of their inherited properties, as well as, two algorithmic procedures for implementing simπ and their practical evaluation w.r.t. a medical ontology SNOMED CT (*cf.* Chapter 4); and
3. The general framework called *assumption-based argumentation with predicate similarity* $\text{ABA}^{(p)}$ framework, which can be seen as an extension of ABA *i.e.* not only assumptions can be used but also similarity of predicates are used to support a claim. In $\text{ABA}^{(p)}$, an argument is represented

by a tree labeled with a unit interval and its acceptability is evaluated w.r.t. the semantics of AA (*cf.* Chapter 5).

1.4 Reader's Guide

The remainder of the thesis is organized as follows:

Chapter 2 briefly summarizes the background in argumentation theory and description logics (DLs). First, the argumentation part basically introduces essential elements of Dung's abstract argumentation (AA) and its semantics. This chapter also introduces less abstract formalisms, dealing in particular with the construction of arguments and the conditions for an argument to attack another *e.g.* assumption-based argumentation (ABA). Second, the description logic part introduces its syntax, semantics, and basic reasoning algorithms, which are widely implemented by conventional reasoners.

Chapter 3 defines the problem of *concept similarity* and gives formal definitions of this problem in DLs. Intuitively, a problem of concept similarity can be seen as a generalization of concept equivalence; hence, it can be also seen as a function which maps two equivalent concepts to 1 and totally dissimilar concepts to 0. We also investigate how a concrete measure in sub-Boolean logics can be developed. Inherited properties are also proven. In contrast to expressive DLs, sub-Boolean DLs are inherently tractable by nature. This fact has motivated us to take a look into them closely in the thesis, especially how ones can generalize the notion of concept equivalence for developing concrete measures (also see (Racharak, 2018, Chapter 4) for additional detailed analysis). This chapter is mainly summarized from our published work (Racharak, Suntisrivaraporn, & Tojo, 2016b, 2016a).

As aforementioned, similarity measures may be subjective to relevant contexts. Chapter 4 investigates the notion of concept similarity under preferences in DLs and existing approaches dealing with this notion. This chapter re-visits and re-defines the development of similarity measures in a more formal way in DLs. After its redefinition, measure sim^{π} is also introduced for measuring the degree of similarity in DL $\mathcal{ELH}$. This chapter also provides mathematical proofs of its properties, studies algorithmic procedures for sim^{π} (*viz*. the top-down approach and the bottom-up approach), and performs empirical evaluation on the medical ontology SNOMED CT. This chapter is mainly summarized from our published work (Racharak, Suntisrivaraporn, et al., 2016a).

Concept similarity under preferences previously introduced is equipped with ABA to define a general framework called *assumption-based argumentation with predicate similarity* $\text{ABA}^{(p)}$ in Chapter 5. As a general framework, it is discussed that $\text{ABA}^{(p)}$ can be defined for any logical language specified by means of 'inference rules' and 'terminological formalism', by identifying 'sentences' in the underlying languages that can be treated as *assumptions* and *concept descriptions*. Like ABA, all semantic notions for determining the acceptability of arguments in AA also apply to arguments in $\text{ABA}^{(p)}$. Hence, it has been investigated in this chapter a constructive proof procedure for determining a grounded set of assumptions. Lastly, it is demonstrated that the proposed framework captures the argumentation scheme for argument from analogy and provides an explanation when it is used for persuasion. This chapter is mainly summarized from our published work (Racharak, Tojo, Hung, & Boonkwan, 2019).

Chapter 6 summarizes the significance of the thesis's results, discusses about the potential applications of the thesis w.r.t. the service science area, and sketches the future research directions. We note that service science is a new discipline emerging from the rapid development of services across the industrial world. Its root is an interdisciplinary study of computer science, operations research, industrial engineering, mathematics, business strategy, management science, decision theory, and social and cognitive science, and legal science. Its goal is to improve essential nature of service, *i.e.* the joint co-creation between service providers and service consumers.

Some results in this thesis have been previously published. Indeed, preference profile, the developed concept similarity measure under preferences profile, and concrete measure sim^{π} for the logic $\mathcal{ELH}$ (discussed in Chapter 3 and Chapter 4) appears in (Racharak, Suntisrivaraporn, & Tojo, 2018; Racharak, Suntisrivaraporn, et al., 2016b, 2016a). The proposed framework in argumentation and a constructive proof procedure (discussed in Chapter 5) appears in (Racharak et al., 2019). Preliminary studies, whose contents are not included in this thesis, can be found in (Racharak & Suntisrivaraporn, 2015; Racharak & Tojo, 2017b, 2017a, 2018; Racharak et al., 2017a, 2017b).

CHAPTER 2
PRELIMINARIES

This thesis exploits benefits of two different reasoning paradigms *viz.* rule reasoning and schemata reasoning. They exhibit certain shortcomings that can be compensated for by advantages of the other. In particular, an argumentation framework is used as underlying mechanisms of rule reasoning and is explained in Section 2.1. Furthermore, description logics are used as underlying mechanisms about conceptual schemata and is described in Section 2.2.

2.1 Argumentation Framework and Its Structure

2.1.1 Abstract Argumentation

An abstract argumentation framework (AA) (Dung, 1995) is a pair (A, R) where A is a (possibly infinite) set of arguments and $R \subseteq A \times A$ is called an *attack relation*. An AA has an obvious representation as a directed graph, in which each node is an argument and each edge connects an attacking argument to an attacked argument. A simple argumentation framework $\mathrm{AA}_{2.1} := \langle \{a, b\}, \{(a, b)\} \rangle$ is shown in Figure 2.1.

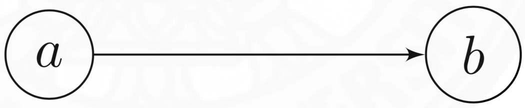

Figure 2.1 $\mathrm{AA}_{2.1}$: A Simple Argumentation Framework.

In AA, an argument is not assumed to have any specific structure; thus, it can represent different situations. For instance, in a context of reasoning about weather, argument *b* may be associated with the inference rule "Tomorrow will rain because the national forecast says so" whereas *a* may be associated with "Tomorrow will not rain because the regional weather forecast says so". In a legal dispute, argument *b* may be associated with the prosecutor's statement "The suspect is guilty because an eyewitness, Mr. Smith, says so" whereas *a* may be associated with "Mr. Smith is an alcohol-addicted and it is proved that he was completely drunk; hence, his testimony should not be considered". In the context of analogical reasoning, argument *b* may represent "I think a goose can quack since it is like a duck" whereas a represents "Though it is similar to a duck, but to say that it can quack, we have to look into their vocal cords; and since they are built differently, it cannot quack".

We may observe that arguments may attack each other. This spells out that arguments may not stand together and their statuses are 'subject to' an evaluation. It is worth mentioning that this evaluation only concerns about the acceptability of an argument – not about the conclusions. Figure 2.2 illustrates this distinction by continuing from the legal dispute example, where argument *c* represents "The suspect is guilty because his fingerprints have been found on the crime scene". Observe that this new argument has no attack relationship with others. Hence, apart from a, argument *c* is also accepted.

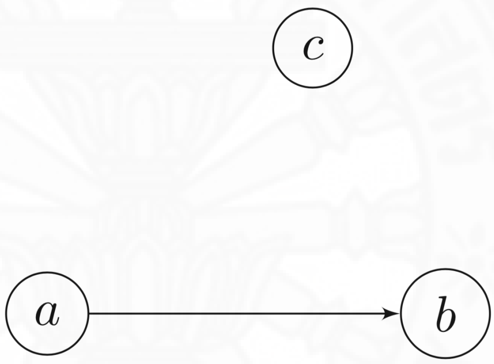

Figure 2.2 $\mathrm{AA}_{2.2}$: An Argumentation Framework Containing An Isolated Argument.

In fact, the notion of 'acceptability' can be defined in many ways and such formal definitions are called *argumentation semantics*. Basically, semantics for AA return sets of arguments called *extensions*, which are *conflict-free* and *defend* themselves against attacks. We formally give their definitions in the following.

Definition 2.1. Given an argumentation framework AA $:= \langle A, R \rangle$, a set $S \subseteq A$ is *conflict-free* iff $\nexists a, b \in S$ such that $(a, b) \in R$.

Given an argumentation framework $\langle A, R \rangle$, we call that a set $S \subseteq A$ of arguments 'attacks' an argument $b \in A$ (denoted by SRb) if $\exists a \in S: (a, b) \in R$, for convenience.

Definition 2.2. Given an argumentation framework AA $:= \langle A, R \rangle$, an argument $a \in A$ is *acceptable w.r.t.* (or *defended by*) a set $S \subseteq A$ iff $\forall b \in A$: $bRa \implies SRb$[2].

Intuitively, 'conflict-free' corresponds to the idea that a set of arguments must be able to stand together and 'acceptability' expresses the idea that an extension is a set of arguments that can withstand its attacks by responding with other attacks. These two properties are used to define the property of *admissibility*, which lies at the heart of all semantics introduced in (Dung, 1995).

Definition 2.3. Given an argumentation framework AA $:= \langle A, R \rangle$, a set $S \subseteq A$ is *admissible* iff S is conflict-free and $\forall a \in S$: a is acceptable w.r.t. S.

Based on the admissibility, Dung has defined four 'traditional' semantics in his landmark paper *viz. complete*, *grounded*, *stable*, and *preferred* semantics. We review each of them in the following.

2.1.1.1 Complete Semantics

The notion of complete extension lies at the heart of all traditional Dung's semantics. It is based on the admissibility defined earlier and a property that "a complete extension must be able defend itself and includes all arguments it defends". The term 'complete semantics' has subsequently gained acceptance in the literature and is used to refer to the formal definition ruling arguments of complete extensions.

Definition 2.4. Given an argumentation framework AA $:= \langle A, R \rangle$, a set $E \subseteq A$ is a complete extension iff E is admissible and $x \in E \iff x$ is acceptable w.r.t. E.

It is worth observing that the empty set is always admissible and that arguments not receiving attacks in an argumentation framework (called *initial arguments*) are acceptable w.r.t. the empty set. According to these observations, it can be shown that the following properties satisfy for any complete extension:

- It cannot be empty;
- It is the empty set iff its initial arguments are empty;

[2] For convenience, we also write aRb for $(a, b) \in R$.

- It subsumes initial arguments.

We illustrate the identification of complete extensions from Figure 2.3 as follows. We observe that its initial arguments are empty *i.e.* $\emptyset$. We also observe that all singletons except $\{b\}$ are admissible. Nonetheless, only $\{a\}$ and $\{d\}$ are complete extensions. Set $\{c\}$ is not complete since it defends a. Now, we consider larger admissible sets. Ones can observe that $\{a, c\}$ and $\{b, d\}$ are complete. Hence, we conclude that $\emptyset, \{a\}, \{d\}, \{a, c\}, \{b, d\}$ are complete extensions.

Figure 2.3 $\text{AA}_{2.3}$: Two Mutual Attacks.

2.1.1.2 Grounded Semantics

The grounded extension includes arguments whose defense is 'rooted' in initial arguments. To put it another way, this set of arguments represents 'strong defense'. This semantics has a correspondence with Pollock's approach (Pollock, 1992) and the well-founded semantics of logic programs (Van Gelder, Ross, & Schlipf, 1991). We give its formal definition in the following.

Definition 2.5. Given an argumentation framework $\text{AA} := \langle A, R \rangle$, a set $E \subseteq A$ is a *grounded extension* iff E is the set inclusion minimal complete extension.

The grounded extension is unique and can be built incrementally as follows. First, the procedure begins with initial arguments and the argumentation framework is modified by suppressing on the arguments attacked by the initial arguments. Next, the initial arguments are re-identified. We note that this set could now become larger since the arguments attacked by the 'new' initial arguments can be suppressed. The process stops when no new initial arguments can be found after a suppression step.

We illustrate the above procedure by considering on Figure 2.3. We observe that the initial arguments of the argumentation framework are empty. Since there are no initial arguments, the process has stopped.

2.1.1.3 Stable Semantics

Stable semantics relies on a very simple intuition *i.e.* an extension should be able to attack all arguments which are not included in it. This leads to the following definition.

Definition 2.6. Given an argumentation framework $\mathrm{AA} := \langle A, R \rangle$, a set $E \subseteq A$ is a *stable* iff E is *conflict-free* and $\forall x \in A: (x \notin E \Longrightarrow ERx)$.

By definition, any stable extension is also a complete extension and a maximal conflict-free set of an argumentation framework. Stable semantics has significant counterparts in several contexts *i.e.* (Dung, 1995) showed its correspondence with solutions of cooperative n-person games, solutions of stable marriage problem, extensions of Reiter's default logic (Reiter, 1980) , and stable models of logic programs (Gelfond & Lifschitz, 1988). Unfortunately, stable semantics has also a significant drawback i.e. there are argumentation frameworks in which no stable extensions exist. A simple example is shown in Figure 2.4.

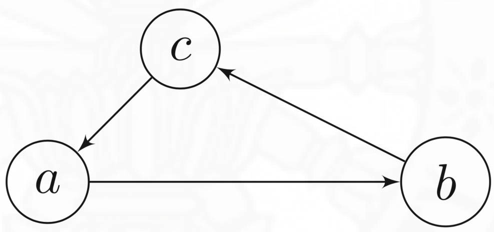

Figure 2.4 $\mathrm{AA}_{2.4}$: A Cyclic Argumentation Framework.

2.1.1.4 Preferred Semantics

The requirement that an extension must attack anything outside it may be too 'aggressive'. This requirement can be relaxed by considering an extension that is as large as possible and is able to defend itself from attacks. This intuition is captured by preferred (or credulous) semantics, which is defined as follows.

Definition 2.7. Given an argumentation framework $\mathrm{AA} := \langle A, R \rangle$, a set $E \subseteq A$ is a *preferred extension* iff E is the set inclusion maximal complete extension.

By definition, we can observe that any stable extension is also a preferred extension, but not vice versa. We illustrate the identification of preferred extensions by considering Figure 2.3. Here, it shows that there are two preferred extensions *viz.* $\{a, c\}$

and $\{b, d\}$. Furthermore, considering Figure 2.4, there is only one preferred extension *viz.* $\emptyset$.

2.1.2 Structured Argumentation

In AA, the structure and meaning of arguments and attacks are abstract. On the one hand, these characteristics enable the study of properties which are independent of any specific aspects (Baroni & Giacomin, 2009). On the other hand, this generality features a limited expressivity and can be hardly adopted to model practical target situations. To fill out this gap, less abstract formalisms were considered, dealing in particular with the construction of arguments and the conditions for an argument to attack another *e.g.* ASPIC$^+$ (Modgil & Prakken, 2014), DeLP (García & Simari, 2004), and assumption-based argumentation (ABA) (Dung et al., 2009). This work extends ABA and we include its basis here for self-containment.

Definition 2.8. An ABA framework is a quadruple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where

- $(\mathcal{L}, \mathcal{R})$ is a deductive system, in which $\mathcal{L}$ is a language and $\mathcal{R}$ is a set of inference rules,
- $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set, referred to as the set of *assumptions*,
- $\overline{}$ is a total mapping from $\mathcal{A}$ to $\mathcal{L}$, where $\bar{\alpha}$ is the contrary of α.

We assume that the inference rules in $\mathcal{R}$ have the syntax $l_0 \leftarrow l_1, \ldots, l_n$ (for $n \geq 0$) where $l_i \in \mathcal{L}$. We refer to l_0 and $l_1, \ldots, l_n$ as the head and the body of the rule, respectively. We also represent the rule $l \leftarrow$ simply as l and restrict our attention to flat ABA framework (Bondarenko, Dung, Kowalski, & Toni, 1997), *i.e.* if $l \in \mathcal{A}$, then there exists no inference rules of the form $l \leftarrow l_1, \ldots, l_n \in \mathcal{R}$ for any $n \geq 0$.

Now, we exemplify how ABA can be used to represent a human being's reasoning. An example is given in terms of dialogue between two fictitious agents called **Agent$_1$** and **Agent$_2$** as follows:

Agent$_1$: I think a goose can quack since it is like a duck.

Agent$_2$: No. Though it is like a duck, but to say that it can quack, we have to look into their vocal cords. Since they are built differently, it cannot quack.

The above example can be considered as analogical reasoning because **Agent$_1$** and **Agent$_2$** employ the perception of similarity as a means to justify their reasoning mechanism. The argumentation scheme for argument from analogy (*cf.* Section 1.2 for the description of its schemata) can be represented in ABA as follows[3]:

$hold(A, C_2) \leftarrow hold(A, C_1), sim(C_1, C_2), arguably(A, C_2)$

where C_i represents different concepts or states of affairs, the conclusion $hold(A, C_2)$ may read "A holds in C_2"; also, the assumption premises $hold(A, C_1), sim(C_1, C_2), arguably(A, C_2)$ may read "A holds in C_1", "C_1 and C_2 are similar to each other", and "the defeasible rule should not apply to the conclusion between A and C_2", respectively.

The above (domain-independent) inference rule is exemplified to the agent reasoning described in our running example. According to the biological family of birds, we know that ducks and geese are belonged to the same family *i.e.* 'Anatidae'. These birds are adapted for swimming, floating on the water surface, *etc.*. Though they are under the same family, ducks and geese are different. This information supports us to conclude that ducks and geese are similar. We represent the assumptions as follows.

$hold(quack, duck)$; $sim(duck, goose)$

where the assumptions $hold(quack, duck)$ and $sim(duck, goose)$ states that "ducks can quack" and "ducks and geese are similar to each other", respectively.

Given an ABA framework, an argument in favor of a sentence $c \in \mathcal{L}$ supported by a set S of assumptions, denoted by $S \vdash c$, is a backward deduction from c to S obtained by applying backward the rules in $\mathcal{R}$, *e.g.* $\{hold(quack, duck), sim(duck, goose), arguably(quack, goose)\} \vdash hold(quack, goose)$.

In ABA, the notion of attack between arguments is defined in terms of the contrary of assumptions, *i.e.* an argument $S_1 \vdash c_1$ attacks another (or the same) argument $S_2 \vdash c_2$ iff c_1 is the contrary of an assumption in S_2.

[3] We use inference rule schemata, with variables starting with capital letters, to stand for the set of all instances obtained by instantiating the variables so that the resulting premises and conclusions are sentences of the underlying language. For simplicity, we omit the formal definition of the language underlying our examples.

In general, the contrary of an assumption is a sentence representing a challenge against the assumption and can be suggested by critical questions (CQ) of an argumentation scheme (*cf.* Section 1.2 for its description). For instance, the assumption $hold(A, C_1)$ can be challenged by providing a negative answer to CQ1 *i.e.* $\neg hold(A, C_1)$, where symbol $\neg$ denotes the classical negation. Supplying a negative answer to CQ2 and CQ3 can also be understood as proving the contrary $\neg sim(C_1, C_2)$ (*i.e.* C_1 and C_2 are dissimilar to each other) of the assumption $sim(C_1, C_2)$. A negative answer to CQ4 can be understood as showing the contrary $\neg hold(A, C_2)$ of the assumption $arguably(A, C_2)$. This contrary $\neg hold(A, C_2)$ may be defined by an additional (domain-independent) inference rule: $\neg hold(A, C_2) \leftarrow sim(C_1, C_2), sim(C_1, C_3), hold(A, C_1), \neg hold(A, C_3)$. Contraries may also be derived via a chain of rules, *e.g.* $\neg hold(quack, A) \leftarrow cord(A, C), \neg built(quack, A); cord(cord_g, goose); \neg built(quack, cord_g)$, representing an abnormality condition that their vocal cords are built differently. The overall ABA framework is summarized in Figure 2.5.

$$\begin{array}{ll}
\mathcal{R}: & hold(A, C_2) \leftarrow hold(A, C_1), sim(C_1, C_2), arguably(A, C_2); \\
& \neg hold(A, C_2) \leftarrow sim(C_1, C_2), sim(C_1, C_3), hold(A, C_1), \neg hold(A, C_3); \\
& \neg hold(quack, C) \leftarrow cord(A, C), \neg built(quack, A); \\
& cord(cord_g, goose);\ \neg built(quack, cord_g) \\
\mathcal{A}: & hold(quack, duck);\ sim(duck, goose);\ arguably(quack, goose) \\
\overline{}: & \overline{hold(quack, duck)} = \neg hold(quack, duck); \\
& \overline{sim(duck, goose)} = \neg sim(duck, goose); \\
& \overline{arguably(quack,\ goose)} = \neg hold(quack,\ goose)
\end{array}$$

Figure 2.5 ABA Framework for The Running Example.

2.1.2.1 Acceptability of Arguments in ABA

ABA is an instance of AA. Hence, all semantic notions for determining the ‘acceptability’ of arguments in AA also apply to arguments in ABA. Moreover, as ones may see, like AA, ABA is also a general purpose argumentation framework that can be used to support various applications or formalize as a specialized framework *e.g.* most default reasoning framework (Bondarenko et al., 1997; Bondarenko, Toni, & Kowalski, 1993; Kakas & Toni, 1999), problems in legal reasoning (Dung & Thang, 2008), problems in practical reasoning and problems in decision theory (Matt, Toni, Stournaras, & Dimitrelos, 2008; Toni, 2007).

The framework proposed in Chapter 5 is extended from ABA and mainly focus on the 'grounded' semantics. Hence, we give informal definition of 'acceptability' for ABA toward the grounded semantics. A claim in ABA could be a potential belief to be justified, which is represented as a sentence in $\mathcal{L}$. To determine the 'acceptability' of a claim, the agent needs to find an argument for it that can be defended against attacks from other arguments. To defend an argument, other arguments must be found and may need to be defended in turn (Dung et al., 2009) . We formally define these characteristics as follows:

- A set of arguments Arg_1 *attacks* a set of arguments Arg_2 if an argument in Arg_1 attacks an argument in Arg_2;
- A set of arguments *Arg defends* an argument *arg* if *Arg* attacks all arguments that attack $\{arg\}$.

Now, we are ready to give informal definitions toward the grounded semantics as follows:

- A set of arguments is *admissible* iff it does not attack itself and it attacks every argument that attacks it;
- An admissible set of arguments is *complete* if it contains all arguments that it defends;
- The least (w.r.t. set inclusion) complete set of arguments is *grounded*.

Given that an argument in ABA attacks another if the former supports the contrary of an assumption in the support of the latter, the correspondence between the assumption view and the argument view (Dung, Mancarella, & Toni, 2007) in ABA can be summarized as follows:

- If a set of assumptions *S* is admissible/grounded, then the union of all arguments supported by any subset of *S* is admissible/grounded;
- If a set of arguments *S* is admissible/grounded, then the union of all sets of assumptions supporting the arguments in *S* is admissible/grounded.

The above notion of acceptable sets of arguments provides a non-constructive specification. Now, we show how to turn the specification into a constructive proof procedure. The method we focus here is defined for a 'grounded' set of arguments introduced in (Dung et al., 2007).

Informally, this constructive proof procedure is known as a *dispute derivation* which is defined as a sequence of transition steps from one state of a dispute to another. For each state, we maintain these following information. Component $\mathcal{P}$ maintains a set of (both standard and analogical) assumptions, which are used to support potential arguments of the proponent. Component $\mathcal{O}$ maintains multiple sets of assumptions, which are used to support all attacking arguments of the opponent. Component D holds a set of assumptions, which have already been used by the proponent. Component C holds a set of assumptions, which have already been used by the opponent and have been attacked by the proponent. In the following, we formally define the dispute derivation for a 'grounded' set of arguments.

Definition 2.9. Let an ABA is a quadruple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$. Given a selection function, a 'grounded belief' dispute derivation of a defence set Δ for a sentence δ is a finite:

$\langle \mathcal{P}_0, \mathcal{O}_0, D_0, C_0 \rangle, \dots, \langle \mathcal{P}_i, \mathcal{O}_i, D_i, C_i \rangle, \dots, \langle \mathcal{P}_n, \mathcal{O}_n, D_n, C_n \rangle$

where $\mathcal{P}_0 := \{\delta\}$, $D_0 := \mathcal{A} \cap \{\delta\}$, $\mathcal{O}_0 := \emptyset$, $C_0 := \emptyset$, $\mathcal{P}_n := \emptyset$, $\mathcal{O}_n := \emptyset$, $\Delta := D_n$, and for every $0 \leq i < n$, only one σ in $\mathcal{P}_i$ or one S in $\mathcal{O}_i$ is selected, and:

1. If $\sigma \in \mathcal{P}_i$ is selected, then
    a. If σ is an assumption, then
       $\mathcal{P}_{i+1} := \mathcal{P}_i \setminus \{\sigma\}$ and $\mathcal{O}_{i+1} := \mathcal{O}_i \cup \{\{\overline{\sigma}\}\}$
    b. Else if there exists an inference rule $\sigma \leftarrow R \in \mathcal{R}$ such that $C_i \cap R = \emptyset$, then
       $\mathcal{P}_{i+1:=}(\mathcal{P}_i \setminus \{\sigma\}) \cup R$ and $D_{i+1} := (\mathcal{A} \cap R)$
2. If S is selected in $\mathcal{O}_i$ and σ is selected in S, then
    a. If σ is an assumption, then
        i. Either σ is ignored, *i.e.*
           $\mathcal{O}_{i+1} := (\mathcal{O}_i \setminus \{S\}) \cup \{S \setminus \{\sigma\}\}$
        ii. Or $\sigma \notin D_i$ and
           $\mathcal{O}_{i+1} := \mathcal{O}_i \setminus \{S\}$, $\mathcal{P}_{i+1} := \mathcal{P}_i \cup \{\bar{\sigma}\}$, $D_{i+1} := D_i \cup (\{\bar{\sigma}\} \cap \mathcal{A})$, and $C_{i+1} := C_i \cup \{\sigma\}$

b. Otherwise, then

$$\mathcal{O}_{i+1} := (\mathcal{O}_i \setminus \{S\}) \cup \{ S \setminus \{\sigma\} \cup R \mid \sigma \leftarrow R \in \mathcal{R}\}$$

A dispute derivation can be seen as a way of representing a 'potential' winning strategy for a proponent to win a dispute against an opponent. The proponent starts by putting forward a claim whose acceptability is under dispute. After that, there are many possibilities as follows, The opponent can try to attack the proponent's claim by arguing for its contrary (*cf.* Case 1.a) or argues for a non-assumption by using an inference rule (*cf.* Case 1.b). Moreover, the proponent can select an assumption in one of the opponent's attacks and either ignores it because it is not selected as a culprit (*cf.* Case 2.a.i) or decides to counter-attack it by showing its contrary (*cf.* Case 2.a.ii). Otherwise, the opponent can argue for a non-assumption by using an inference rule (*cf.* Case 2.b). We give an informal dispute derivation for the working example as follows.

Example 2.1. Consider an ABA given in Figure 2.5. Table 2.1 shows that there does not exist a grounded belief dispute derivation for $hold(quack, goose)$[4].

At step 1, the proponent ($\mathcal{P}$) has completed the construction of an argument for $h(q, g)$ supported by '$h(q, d)$, $s(d, g)$, and $a(q, g)$', saying that "*geese quack because ducks also quack and ducks are similar to geese*". At step 2, the opponent ($\mathcal{O}$) has decided to attack on assumption $h(q, d)$ by showing its contrary $\neg h(q, d)$. However, this attack fails to build on step 3. Again, the opponent ($\mathcal{O}$) has decided to attack on assumption $s(d, g)$ by showing its contrary $\neg s(d, g)$ at step 4 and fails to build its support at 5. At step 6, the opponent ($\mathcal{O}$) has decided to attack on assumption $a(q, g)$ by showing its contrary $\neg h(q, g)$. This argument is fully constructed at step 9, in which no assumptions have been used. Thus, this dispute derivation fails.

With an analogous manner, Table 2.2 shows a successful dispute derivation for $\neg hold(quack, goose)$ with three transition steps.

Table 2.1 A Failed Grounded Belief Dispute Derivation for $hold(quack, goose)$.

Step	$\mathcal{P}$	$\mathcal{O}$	D	C
0	$\{h(q, g)\}$	$\emptyset$	$\emptyset$	$\emptyset$

[4] Obvious abbreviations are used here for the sake of succinctness.

1	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{s(d,g), a(q,g)\}$	$\{\{\neg h(q,d)\}\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
3	$\{s(d,g), a(q,g)\}$	$\emptyset$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
4	$\{a(q,g)\}$	$\{\{\neg s(d,g)\}\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
5	$\{a(q,g)\}$	$\emptyset$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
6	$\emptyset$	$\{\{\neg h(q,g)\}\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
7	$\emptyset$	$\{\{c(c_g,g), \neg b(q,c_g)\}\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
8	$\emptyset$	$\{\{\neg b(q,c_g)\}\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$
9	$\emptyset$	$\{\emptyset\}$	$\{h(q,d), s(d,g), a(q,g)$	$\emptyset$

Table 2.2 A Successful Grounded Belief Dispute Derivation for $\neg hold(quack, goose)$.

Step	$\mathcal{P}$	$\mathcal{O}$	D	C
0	$\{\neg h(q,g)\}$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{c(c_g,g), \neg b(q,c_g)\}$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{\neg b(q,c_g)\}$	$\emptyset$	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

2.2 Description Logics

Description logics (DLs) (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2007; Baader & Sattler, 2001; Calvanese, De Giacomo, Lenzerini, & Nardi, 2001) are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well understood way. The name *description logics* is coined based on the fact that the application domain is described by concept 'descriptions', *i.e.* expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL.

DL Knowledge base is captured by two different formalisms *viz.* a terminology (TBox) representing general knowledge about the problem domain and an assertion (ABox) representing knowledge about a specific situation. Furthermore, the ABox part may be ignored. In this thesis, our attempt mainly focuses on argumentative patterns containing ‘descriptive’ analogy and ‘normative’ analogy; and, the ABox part is not used. Therefore, we only review the basics of TBox in this section.

2.2.1 Description Languages

We assume two disjoint sets of concept names CN and role names RN. Description languages are distinguished by a set of *concept constructors* they provide. These constructors are used to inductively define concept descriptions (simply *concepts*). It is obvious that the more concept constructors a particular DL provides, the more expressive concepts can be constructed. In abstract notations, we use A and B to denote *atomic concepts*, C and D to denote *concept descriptions*[5], and r to denote an *atomic role*. Table 2.3 lists common concept constructors that are widely considered in the literature. The second and the third columns show the syntax and semantics elements, respectively. The attributive language $\mathcal{AL}$ was introduced in (Smolka, 1991) as a minimal language that is of practical interest. $\mathcal{AL}$ provides exactly the constructors as in the table except existential quantification ($\exists r.C$).

Table 2.3 Syntax and Semantics of Concept Constructors.

Constructor Name	Syntax	Semantics
Top Concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom Concept	$\bot$	$\emptyset$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{D}}$
Atomic Negation	$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Nominal	$\{a_1, \dots, a_n\}$	$\{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$
Limited Existential Quantification	$\exists r.\top$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e : (d,e) \in r^{\mathcal{I}}\}$
Existential Quantification	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e : (d,e) \in r^{\mathcal{I}} \land e \in C^{\mathcal{I}}\}$
Universal Restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e : (d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$

[5] The precise definition of concept description is given later.

The first naming scheme for DLs was also proposed in (Smolka, 1991): starting from the DL $\mathcal{AL}$, additional constructors are indicated by appending corresponding letters; *e.g.* $\mathcal{ALC}$ (which stands for *Attributive Language with Complement*) is obtained from $\mathcal{AL}$ by featuring the complement operator ($\neg$) and $\mathcal{ALE}$ is obtained from $\mathcal{AL}$ by adding existential quantification ($\exists r.C$). DL $\mathcal{ALC}$ is considered as the smallest Boolean-closed logic[6].

There are also a number of interesting sub-Boolean DLs[7], most of which disallow disjunction and (full) negation such as $\mathcal{FL}_0$ and $\mathcal{EL}$. For historical naming reasons, $\mathcal{FL}_0$ is obtained by disallowing atomic negation and limited existential quantification from $\mathcal{AL}$; and also, $\mathcal{EL}$ is obtained by disallowing atomic negation and universal restriction from $\mathcal{AL}$. Both are sub-languages of $\mathcal{AL}$ that are practical interest due to their practical efficiency and sufficient expressivity.

Let $\mathcal{L}$ be a specific DL. We denote the set of concept descriptions for DL $\mathcal{L}$ by $\mathsf{Con}(\mathcal{L})$. In this thesis, we merely focus on the logic $\mathcal{EL}$. Hence, formal definitions for the syntax and semantics of $\mathsf{Con}(\mathcal{EL})$ are given in the following.

Definition 2.10 ($\mathcal{EL}$ Concept Description). Let CN be a set of concept names, RN be a set of role names, and $\top$ be the top concept. A set of $\mathcal{EL}$ concept descriptions (denoted by $\mathsf{Con}(\mathcal{EL})$) is the smallest set such that:

1. If $A \in \mathrm{CN} \cup \{\top\}$, then $A \in \mathsf{Con}(\mathcal{EL})$;
2. If $C, D \in \mathsf{Con}(\mathcal{EL})$ and $r \in \mathrm{RN}$, then $C \sqcap D, \exists r.C \in \mathsf{Con}(\mathcal{EL})$.

The following example illustrates how ones can construct $\mathcal{EL}$ concept descriptions based on sets of concept names and role names defined in SNOMED CT.

Example 2.2. The concept of Endocarditis, whose members are an inflammation which has location on an endocardium tissue, may be expressed using concept names and role names in SNOMED CT: $\mathsf{Inflammation} \sqcap \exists \mathsf{hasLocation}.\mathsf{Endocardium}$.

[6] Strictly speaking, a DL must provide at least one quantifier, *i.e.* either existential or universal. Thus, the logic with the first five constructors in Table 2.3 is not a DL as it is equivalent to the propositional logic.

[7] Sub-Boolean DLs are DLs that are not equipped with all Boolean operators

We can agree that Endocarditis is an $\mathcal{EL}$ concept because, following Definition 2.10, it is obvious that:

1. Inflammation is an $\mathcal{EL}$ concept;
2. $\exists$hasLocation.Endocardium is also an $\mathcal{EL}$ concept;
3. Finally, Inflammation $\sqcap$ $\exists$hasLocation.Endocardium is an $\mathcal{EL}$ concept.

Though sub-Boolean DLs are not very expressive, they are also of theoretical interest due to their tractability. Table 2.4 shows the worst-case complexity of concept satisfiability problem[8] in $\mathcal{ALC}$ and the subsumption problem in $\mathcal{FL}_0$ and $\mathcal{EL}$. It is worth noting that the satisfiability problem is trivial in $\mathcal{FL}_0$ and $\mathcal{EL}$ since any concept expressed in these languages is satisfiable. The table also shows that $\mathcal{EL}$ exhibits the most robust behavior w.r.t. every type of terminology.

Table 2.4 Comparing The Description Logics.

Terminology	$\mathcal{ALC}$	$\mathcal{FL}_0$	$\mathcal{EL}$
The empty TBox	PSpace-complete (Smolka, 1991)	Polynomial (Brachman & Levesque, 1984)	Polynomial (Baader, Küsters, & Molitor, 1999)
Acyclic TBoxes	PSpace-complete (Smolka, 1991)	coNP-complete (Nebel, 1990)	Polynomial (Baader, 2003)
General TBoxes	ExpTime-complete (Schild, 1991)	ExpTime-complete (Baader, Brandt, & Lutz, 2005)	Polynomial (Brandt, 2004)

It should be also noted that these results are not merely theoretical interest. In fact, they also provide sufficiently expressivity. For instance, SNOMED CT[9] (Spackman, 2005; Stearns, Price, Spackman, & Wang, 2001) and Gene Ontology (Ashburner et al., 2000) employ $\mathcal{EL}$. It is also worth noting that $\mathcal{FL}_0$ and $\mathcal{EL}$ are the minimal candidate DLs to pursue a polynomial complexity since they would not inherit NP-hardness from the propositional logic (Van Harmelen, Lifschitz, & Porter, 2008).

[8] Subsection 2.2.3 gives precise definitions of most widely used reasoning services in DLs.
[9] http://bioportal.bioontology.org/ontologies/SNOMEDCT

Like any DLs, the semantics of $\mathcal{EL}$ concepts is defined through interpretations as shown in the following.

Definition 2.11 (Semantics of $\mathcal{EL}$ Concept). An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$ of interpretation domain and an interpretation function $\cdot^{\mathcal{I}}$, which assigns to each concept name $A \in \mathrm{CN}$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role name $r \in \mathrm{RN}$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to a concept descriptions by inductive definitions given in the right column of Table 2.3.

An interpretation $\mathcal{I}$ is said to be a *model* of a concept C, or $\mathcal{I}$ models C, the interpretation of C in $\mathcal{I}$, *i.e.* $C^{\mathcal{I}}$, is not empty, *i.e.* $C^{\mathcal{I}} \neq \emptyset$.

Example 2.3. Given a concept Inflammation ⊓ ∃hasLocation.Endocardium, we can find an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $\mathcal{I}$ is a model of the concept as follows:

1. Suppose an interpretation domain $\Delta^{\mathcal{I}} = \{a, b, c, d\}$;
2. Suppose $\mathrm{Inflammation}^{\mathcal{I}} = \{a, b, c\}$, $\mathrm{Endocardium}^{\mathcal{I}} = \{d\}$, and $\mathrm{hasLocation}^{\mathcal{I}} = \{(a, d), (b, c)\}$;
3. From Table 2.3, we know $(\exists \mathrm{hasLocation.Endocardium})^{\mathcal{I}} = \{a\}$;
4. From Table 2.3, we know $(\mathrm{Inflammation} \sqcap \exists \mathrm{hasLocation.Endocardium})^{\mathcal{I}} = \{a\}$.

Since $(\mathrm{Inflammation} \sqcap \exists \mathrm{hasLocation.Endocardium})^{\mathcal{I}} \neq \emptyset$, then the defined interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a model of the concept Inflammation ⊓ ∃hasLocation.Endocardium.

Table 2.5 Syntax and Semantics of Ontological Constructors.

Constructor Name	Syntax	Semantics
Concept Definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
Concept Inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Concept Disjointness	$C \sqcap D \sqsubseteq \bot$	$C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$
Domain Restriction	$\mathrm{domain}(r) \sqsubseteq C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e: (d, e) \in r^{\mathcal{I}}\} \subseteq C^{\mathcal{I}}$

Range Restriction	$\mathrm{range}(r) \sqsubseteq C$	$\{e \in \Delta^{\mathcal{I}} \mid \exists d: (d,e) \in r^{\mathcal{I}}\} \subseteq C^{\mathcal{I}}$
Functionality	$\mathrm{functional}(r)$	$\forall d \in \Delta^{\mathcal{I}}: \#\{e \in \Delta^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\} \leq 1$
Reflexivity	$\mathrm{reflexive}(r)$	$\forall d \in \Delta^{\mathcal{I}}: (d,d) \subseteq r^{\mathcal{I}}$
Transitivity	$\mathrm{transitive}(r)$	$\forall d,e,f \in \Delta^{\mathcal{I}}: (d,e),(e,f) \in r^{\mathcal{I}} \rightarrow (d,f) \in r^{\mathcal{I}}$
Role Hierarchy	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
Role Inclusion	$r_1 \circ \cdots \circ r_k \sqsubseteq s$	$r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq s^{\mathcal{I}}$

2.2.2 DL Terminological Formalism

We have seen how concept descriptions are built through the use of 'concept constructors'. Now, we want to form statements representing the general knowledge about the problem domain *i.e.* terminologies. For example,

$\mathrm{Endocarditis} \equiv \mathrm{Inflammation} \sqcap \exists \mathrm{hasLocation.Endocardium}$

saying that "endocarditis is an inflammation that has location on endocardium tissue". This kind of statements is seen as a terminological formalism (or a TBox statement) and can be characterized by a set of *ontological constructors* in DLs. Table 2.5 lists most commonly used constructors in the literature where the middle and the right column show their syntax and semantics. Formally, a TBox is defined as follows.

Definition 2.12 (TBox). Let $\mathcal{L}$ be a specific DL, $A \in \mathrm{CN}$, and $C \in \mathrm{Con}(\mathcal{L})$. Then, $\mathrm{A} \equiv \mathrm{C}$ and $\mathrm{A} \sqsubseteq \mathrm{C}$ are called a *concept definition* and a *primitive concept definition*, respectively. Let $\rhd$ denote either $\equiv$ or $\sqsubseteq$. Then, TBox $\mathcal{T}$ is a finite set of (possibly primitive) concept definitions. A concept definition $A \rhd C$ is unique if, for each $A \in \mathrm{CN}$, there is at most one concept definition $A \rhd C$ for some $C \in \mathrm{Con}(\mathcal{L})$.

We call *A directly uses B* in $\mathcal{T}$ if $A \rhd B$ occurs in $\mathcal{T}$ and we define *uses* to be the transitive closure of the relation *directly uses*. Then, a concept definition $A \rhd C$ is cyclic if A *uses* itself. Otherwise, we call such definition an *acyclic* concept definition. TBox $\mathcal{T}$ is called *unfoldable* if all concept definitions are unique and acyclic definitions.

A concept name P in $\mathcal{T}$ is said to be *undefined* if it is neither fully defined nor primitively defined in $\mathcal{T}$.

An interpretation $\mathcal{I}$ is a model of a concept definition $A \equiv C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$ and is a model of a primitive concept definition $A \sqsubseteq C$ iff $A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. $\mathcal{I}$ is a model of $\mathcal{T}$ iff it is a model of every definition $A \rhd C$ in $\mathcal{T}$.

Given an unfoldable TBox $\mathcal{T}$, concept names occurring on the left-hand side of a concept definition are called *defined concept names* (denoted by $\mathsf{CN}^{\mathrm{def}}$) whereas the others are called primitive concept names (denoted by $\mathsf{CN}^{\mathrm{pri}}$). The name *unfoldable* is motivated by the fact that, in such a TBox $\mathcal{T}$, $\mathcal{T}$ can be transformed into an equivalent one $\mathcal{T}'$ by substituting all the defined concept names in concept descriptions with their definitions until only primitive concept names remain. In particular, for a concept definition defined in $\mathcal{T}$ by an axiom $A \equiv D$, the procedure is simply to replace A with D whenever if occurs in C, and then to recursively unfold D. For a "primitive" concept definition defined in $\mathcal{T}$ by an axiom $A \sqsubseteq D$, the procedure is slightly more complex. Whenever A occurs in C, it is replaced with the concept $X \sqcap D$ where X is a new concept name not occurring in $\mathcal{T}$ or C. After that, D is recursively unfolded. We note that X represents the unspecified characteristics that differentiate it from D. Such unfolded concepts which remain only primitive concept names are called *fully expanded* concepts. This transformation is called unfolding and we use $\mathrm{Unfold}(C, \mathcal{T})$ to denote that the concept C is unfolded w.r.t. $\mathcal{T}$.

When $\mathcal{T}$ is unfolded to $\mathcal{T}'$, each defined concept name in $\mathcal{T}'$ is an independent concept description in a sense that the TBox itself can be disregarded. From a computational point of view, unfoldable TBoxes are interesting since they may allow for the use of simplified reasoning techniques (*cf.* Table 2.4) and reasoning in the presence of a TBox is often harder than that without a TBox (or an empty TBox).

A much more expressive formalism of TBox is called a *general TBox* where each statement is called a *general concept inclusion*. Informally, a general concept inclusion is a statement like this form: $\exists\mathsf{married}.\mathsf{Human} \sqsubseteq \mathsf{Human}$ saying that "a human is only married to a human". This general formalism is supported by most state-of-the-art DL reasoners. In the following, we gives a formal definition for a general TBox.

Definition 2.13 (General TBox). Let $\mathcal{L}$ be a specific DL and $C, D \in \mathsf{Con}(\mathcal{L})$. Then, a general concept inclusion (GCI) is of the form $\mathsf{C} \sqsubseteq \mathsf{D}$. Then, *general TBox* is a finite set of GCIs.

An interpretation $\mathcal{I}$ is called a model of a GCI $C \sqsubseteq D$ iff $\mathsf{C}^{\mathcal{I}} \subseteq \mathsf{D}^{\mathcal{I}}$. $\mathcal{I}$ is a model of a general TBox $\mathcal{T}$ iff it is a model of every GCI in $\mathcal{T}$.

It should be also noted that general TBoxes are more general than unfoldable TBoxes since GCIs can be used to express (primitive) concept definitions. In particular, a primitive concept definition is a special form of GCI whereas a concept definition can be expressed by means of two GCIs, *i.e.* $\mathsf{A} \equiv \mathsf{C}$ with $\mathsf{A} \sqsubseteq \mathsf{C}$ and $\mathsf{C} \sqsubseteq \mathsf{A}$.

It is also worth noting that, following (Nebel, 1991), the semantics we have studied so far is called *descriptive semantics*. This semantics can produce counter-intuitive results when a TBox contains cyclic dependency. In such a case, the so-called *fixpoint semantics* (De Giacomo & Lenzerini, 1996; Nebel, 1991) is recommended to use. However, the descriptive semantics is adopted in this thesis because of its wide acceptance as the most appropriate one (Donini, lenzerini, Nardi, & Schaerf, 1996; Kohlas, 2003).

Apart from concept definitions and inclusions, there are also interesting and important ontological constructors. Some of them are listed in the upper part of Table 2.5. In some cases, one constructor can be simulated by another. For instance, a domain restriction $\mathsf{domain}(r) \sqsubseteq C$ can be expressed by the GCI $\exists \mathsf{r}.\top \sqsubseteq \mathsf{C}$. Also, reflexivity, transitivity, and role hierarchy are special forms of role inclusion[10], *i.e.*, $\epsilon \sqsubseteq \mathsf{r}$, $\mathsf{r} \circ \mathsf{r} \sqsubseteq \mathsf{r}$, $\mathsf{r} \sqsubseteq \mathsf{s}$, respectively. Table 2.6 presents various DLs with their supported constructors, where ° denotes optional features that may or may not be supported.

Table 2.6 Logical Constructors in Various DLs.

DL Dialects	$\mathcal{L}_0$	$\mathcal{EL}$	$\mathcal{FL}_0$	$\mathcal{ELH}$	$\mathcal{ALC}$	$\mathcal{SHIF}$	$\mathcal{SROIQ}$
Top Concept	●	●	●	●	●	●	●
Bottom Concept					●	●	●
Conjunction	●	●	●	●	●	●	●
Disjunction					●	●	●
Negation					●	●	●
Nominal							●

[10] Sometimes such statements are regarded as another component called RBox.

Existential Restrictions		●		●	●	●	●
Value Restrictions			●		●	●	●
Concept Definition	○	○	○	○	○	○	○
Concept Inclusion	○	○	○	○	○	○	○
Domain Restriction		○	○	○	○	○	○
Concept Disjointness					○	○	○
Range Restriction					○	○	○
Functionality						●	●
Reflexivity							●
Transitivity						●	●
Role Hierarchy				●		●	●
Role Inclusion							●
Concept Assertion	○	○	○	○	○	○	●
Role Assertion		○	○	○	○	○	●

2.2.3 Reasoning Services

Reasoning services are processes of discovering valid statements in DL knowledge base and can be basically defined by means of logical inferences. We introduce some prominent ones for TBox in the following.

Definition 2.14 (Concept Satisfiability). Let $\mathcal{L}$ be a specific DL, $\mathcal{T}$ be a TBox, and $\mathrm{C} \in \mathsf{Con}(\mathcal{L})$. Then, a concept C is called *satisfiable* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{J}$ of $\mathcal{T}$ with $\mathrm{C}^{\mathcal{J}} \neq \emptyset$.

Definition 2.15 (Concept Subsumption). Let $\mathcal{L}$ be a specific DL, $\mathcal{T}$ be a TBox, and $C, D \in \mathsf{Con}(\mathcal{L})$. Then, a concept D *subsumes* a concept C w.r.t. $\mathcal{T}$ (denoted by $\mathcal{T} \models \mathrm{C} \sqsubseteq \mathrm{D}$ or $\mathrm{C} \sqsubseteq_{\tau} \mathrm{D}$) if $\mathrm{C}^{\mathcal{J}} \subseteq \mathrm{D}^{\mathcal{J}}$ holds for all models $\mathcal{J}$ of $\mathcal{T}$.

Definition 2.16 (Concept Equivalence). Let $\mathcal{L}$ be a specific DL, $\mathcal{T}$ be a TBox, and $C, D \in \mathsf{Con}(\mathcal{L})$. Then, two concepts C, D are *equivalent* w.r.t. $\mathcal{T}$ (denoted by $\mathcal{T} \models \mathrm{C} \equiv \mathrm{D}$ or $\mathrm{C} \equiv_{\tau} \mathrm{D}$) if $\mathcal{T} \models \mathrm{C} \sqsubseteq \mathrm{D}$ and $\mathcal{T} \models \mathrm{D} \sqsubseteq \mathrm{C}$.

The reasoning services introduced in Definition 2.14 - 2.16 are called *basic reasoning services* and should be supported by most DL systems. There are also additional services which could be implemented by a finite number of calls to the basic services. These are formally defined as follows.

Definition 2.17 (Ontology Classification). Let $\mathcal{T}$ be a TBox and $\mathrm{CN}(\mathcal{T})$ be a set of concept names occurring in $\mathcal{T}$. Then, an ontology classification of $\mathcal{T}$ is the identification of subsumption between all pairs of concept names in $\mathcal{T}$, *i.e.* for all $\mathrm{A}, \mathrm{B} \in \mathrm{CN}(\mathcal{T})$, determines whether or not $\mathcal{T} \vDash \mathrm{A} \sqsubseteq \mathrm{B}$.

Reasoning may become conceptually easier by abstracting away from the TBox or assuming that it is empty. The following theorem formally presents the use of unfolding for TBox elimination. When the TBox is eliminated, we omit to denote it.

Theorem 2.1. Let $\mathcal{L}$ be a specific DL and $\mathcal{T}$ be a TBox. Then, for every pair $\mathrm{C}, \mathrm{D} \in \mathrm{Con}(\mathcal{L})$, we have:

$$\mathcal{T} \vDash \mathrm{C} \sqsubseteq \mathrm{D} \Leftrightarrow \vDash \mathrm{Unfold}(\mathrm{C}, \mathcal{T}) \sqsubseteq \mathrm{Unfold}(\mathrm{D}, \mathcal{T})$$

The procedure of unfolding is only restricted to an unfoldable TBox $\mathcal{T}$ (Baader et al., 2007). For example, if $\mathcal{T}$ is not unique, e.g. $\{(\mathrm{A} \equiv \mathrm{C}), (\mathrm{A} \equiv \mathrm{D})\} \subseteq \mathcal{T}$, then it is not possible to make precisely the substitution for A. If $\mathcal{T}$ contains the cyclic dependency, it could lead to a non-termination problem. If T contains GCIs, *e.g.* $\exists \mathrm{r}.\mathrm{C} \sqsubseteq \mathrm{D}$, then it could not be guaranteed that an interpretation satisfying an unfolded concepts would also satisfy these logical statements.

2.2.4 Reasoning Algorithms

A variety of reasoning algorithms were introduced for the services discussed earlier. Two widely used algorithmic approaches are tableau-based approaches and structural ones for sub-Boolean DLs. Our proposed computational method in Chapter 3 is developed based on the structural approach for $\mathcal{EL}$. Hence, we review its basis here for self-containment.

Before looking at an algorithmic procedure, let us state the general requirements on the 'behaviors' of such procedures (Baader, Horrocks, & Sattler, 2004) as follows:

1) The procedure should be a *decision procedure*[11], meaning that it should be:
    a) Sound *i.e.* the positive answers should be correct,
    b) Complete *i.e.* the negative answers should be correct, and

[11] This can be seen as a metaphorical meaning of soundness and completeness in logic.

c) Terminating *i.e.* it should always give an answer in finite time;

2) The procedure should be as 'efficient' as possible. That is, it should be 'optimal' w.r.t. the worst-case complexity of the problem;

3) The procedure should be 'practical', *i.e.* it should be easy to be implemented, be easy to be optimized, and behave well in applications.

2.2.4.1 Structural Approach for $\mathcal{EL}$

When trying to find a DL with a polynomial subsumption algorithm, it is clear that such a particular DL should not provide all Boolean operators since it will inherit NP-hardness from propositional logic (Baader et al., 2004). When ones have to decide to drop an operator, conjunction seems to be indispensable since it is used to state for different properties of a defining concept. Finally, if ones want to call that logic a DL, a constructor using roles is needed. This leads to the consideration of two minimal candidate sub-Boolean DLs, *viz.* $\mathcal{FL}_0$ and $\mathcal{EL}$. As aforementioned in Table 2.4, these DLs exhibit robust behaviors. The following discusses the characterization for DL $\mathcal{EL}$ in detail.

Suppose that TBox $\mathcal{T}$ is unfoldable and $\mathcal{EL}$ concepts are fully expanded. Let an $\mathcal{EL}$ concept C is of the following form:

$$\mathrm{P}_1 \sqcap \ldots \sqcap \mathrm{P}_\mathrm{m} \sqcap \exists \mathrm{r}_1.\mathrm{C}_1 \sqcap \ldots \sqcap \exists \mathrm{r}_\mathrm{n}.\mathrm{C}_\mathrm{n} \quad (2.1)$$

That concept C can be structurally transformed into the corresponding $\mathcal{EL}$ description tree. The root v_0 of the $\mathcal{EL}$ description tree $\mathcal{T}_C$ has $\{\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{m}\}$ as its label and has n outgoing edges, each labeled r_j to a vertex v_j for $1 \leq \mathrm{j} \leq \mathrm{n}$. Then, a subtree with the root v_j is defined recursively relative to the concept C_j. In (Baader, 2003; Baader, Brandt, & Küsters, 2001), a characterization of subsumption for the DL $\mathcal{EL}$ w.r.t. an unfoldable TBox was proposed. Instead of considering concept descriptions, the so-called $\mathcal{EL}$ description trees corresponding to those concept descriptions are used. The subsumption is then characterized by an existence of a homomorphism in the reverse direction (*cf.* Theorem 2.2).

Definition 2.18 (Homomorphism (Baader, 2003; Baader et al., 2001)**).** An $\mathcal{EL}$ description tree $\mathcal{T}$ is a quintuple $(\mathrm{V}, \mathrm{E}, rt, \mathrm{l}, \rho)$ where V is a set of vertices, $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ is a set of edges, rt is the root, $l: V \rightarrow 2^{CN^{pri}}$ is a vertex labeling function, and $\rho: E \rightarrow RN$ is an edge labeling function. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two $\mathcal{EL}$ description trees, $v_1 \in V_1$, and $v_2 \in V_2$. Then, the mapping $h: V_1 \rightarrow V_2$ is a *homomorphism* from $\mathcal{T}_1$ to $\mathcal{T}_2$ iff the following conditions are satisfied:

- For all $v_1 \in V_1$, $l_1(v_1) \subseteq l_2(h(v_1))$; and
- For each successor w_1 of v_1 in $\mathcal{T}_1$, $h(w_1)$ is a successor of $h(v_1)$ with $\rho_1(v_1, w_1) = \rho_2(h(v_1), h(w_1))$

Theorem 2.2 ((Baader, 2003; Baader et al., 2001)**).** Let $C, D \in Con(\mathcal{EL})$ and $\mathcal{T}_C$ and $\mathcal{T}_D$ be the corresponding description trees. Then, $C \sqsubseteq D$ iff there exists a homomorphism (denoted by $h: \mathcal{T}_D \rightarrow \mathcal{T}_C$) which maps the root v of $\mathcal{T}_D$ to the root w of $\mathcal{T}_C$.

We illustrate how the subsumption relation between $\mathcal{EL}$ concepts in Example 2.4.

Example 2.4. Let a family TBox is given as follows: $\text{GrandFather} \equiv \text{Man} \sqcap \exists\text{hasChild.Parent}$, $\text{Man} \equiv \text{Male} \sqcap \text{Person}$, and $\text{Parent} \equiv \text{Person} \sqcap \exists\text{hasChild.Person}$. By unfolding, it yields a semantically equivalent TBox $\mathcal{T}'$ as follows:

$\text{GrandFather} \equiv \text{Male} \sqcap \text{Person} \sqcap \exists\text{hasChild.}(\text{Person} \sqcap \exists\text{hasChild.Person})$

$\text{Man} \equiv \text{Male} \sqcap \text{Person}$

$\text{Parent} \equiv \text{Person} \sqcap \exists\text{hasChild.Person}$

To show that $\text{GrandFather} \sqsubseteq \text{Parent}$, we construct the description tree $\mathcal{T}'_{\text{GrandFather}}$ for the concept GrandFather (*cf.* Figure 2.6a) and the description tree $\mathcal{T}'_{\text{Parent}}$ for the concept Parent (*cf.* Figure 2.6b). Following Definition 2.18, it is not difficult to identify a homomorphism from $\mathcal{T}'_{\text{Parent}}$ to $\mathcal{T}'_{\text{GrandFather}}$. Thus, $\text{GrandFather} \sqsubseteq \text{Parent}$.

As shown in (Baader, 2003; Baader et al., 2001), this form of characterization can be decided in polynomial time (*cf.* Table 2.4). This result is not only of theoretical interest. In fact, well-known medical ontologies such as Gene Ontology (Ashburner et al., 2000) and SNOMED CT (Spackman, 2005; Stearns et al., 2001) are expressible with the logic $\mathcal{EL}$. As we shall also see soon, our concrete development for concept similarity under subjective factors in $\mathcal{ELH}$[12] is driven by this form of structural subsumption.

(a) The description tree $\mathcal{T}'_{\mathsf{GrandFather}}$

(b) The description tree $\mathcal{T}'_{\mathsf{Parent}}$

Figure 2.6 The Corresponding Description Trees of Concepts GrandFather and Parent.

[12] Strictly speaking, $\mathcal{ELH}$ extends $\mathcal{EL}$ with the role hierarchy.

CHAPTER 3
CONCEPT SIMILARITY IN DESCRIPTION LOGICS

Considering examples in Chapter 1, ones may observe that the perception of 'concept similarity' used by both descriptive analogy and normative analogy can be explicitly reconstructed by considering the description or taxonomy of concepts. For instance, "London is the capital city and one of the big cities of England" and "Amsterdam is the capital city and one of the big cities of Netherlands". Regarding this observation, any frameworks developed for analogical argumentation must provide mechanisms to formalize the description of concepts. In a very simple way, we may formalize the description of concepts in terms of inference rules. For instance, the descriptions of the above city example can be represented by

$\mathrm{london}(X) \leftarrow \mathrm{capital\_city}(X,Y),\mathrm{big\_city}(X,Y),\mathrm{england}(Y)$ and
$\mathrm{amsterdam}(X) \leftarrow \mathrm{capital\_city}(X,Y),\mathrm{big\_city}(X,Y),\mathrm{netherlands}(Y)$

respectively. When inference rules are grounded, ones can employ the model theory to derive the similarity between predicates as in (Goebel, 1989b; Haraguchi & Arikawa, 1987).

Though using inference rules can encode our example, other knowledge representation formalisms which provide more expressivity may be also used to encode concepts *e.g.* description logics (cf. Subsection 2.2) or (other fragments of) first-order logic. For instance, the same description can be formalized based on TBox formalism as:

$\mathrm{London} \sqsubseteq \mathrm{CapitalCity} \sqcap \mathrm{BigCity} \sqcap \exists \mathrm{isCityOf.England}$ and
$\mathrm{Amsterdam} \sqsubseteq \mathrm{CapitalCity} \sqcap \mathrm{BigCity} \sqcap \exists \mathrm{isCityOf.Netherlands}$

respectively. Successful examples of DL knowledge bases are ontologies in medicine and bioinformatics *e.g.* SNOMED CT (www.snomed.org) or GO (www.geneontology.org).

This chapter addresses the perception of concept similarity and defines it as a human judgment of a degree to which a pair of concepts in question is similar. *Concept* *similarity measures* are computational techniques attempting to imitate the human judgments of concept similarity. Formally, they aim at identifying a degree of commonality of two given concepts and can be regarded as a generalization of the classical reasoning problem of equivalence *i.e.* two concepts (or states of affairs) are identical if and only if their degree of 'similarity' is equal to 1. It is worth noting that similarity of concepts is oftentimes context-sensitive and can be recognized from the comparison of features shared between them. Nevertheless, (Hesse, 1965; Waller, 2001; Weinreb, 2016) reported that features used in comparisons should be 'relevant' to the attribution of the property. This means that there must be ways of expressing aspects of a context in consideration.

3.1 Preference Context for Having Relevance

This section introduces preference context (denoted by $\mathfrak{p}$) as a collection of 'abstract' preferential elements in which the development of similarity measure of concepts for a particular cognitive agent should consider. Its first intuition is to represent different forms of preferences (of an agent) based on concept names and role names. Similarity measure which adopts this notion is flexible to be tuned by an agent and can determine the similarity conformable to that agent's perception.

The syntax and semantics of each form are given in terms of 'partial' functions because agents may not have preferences over all concept names and role names. We recommend to devise similarity measure with consideration on an appropriate 'instance' of preference context if we aim at developing concept similarity measure for general purposes *i.e.* a measure based on both subjective and objective factors.

Let I, S, D be non-empty sets equipped with total orders $\leq_I$, $\leq_S$, and $\leq_D$, respectively. As aforementioned, our intention is to define preference context for representing a broad notion of the user's preferences *i.e.* the 'abstract' notion level. At its 'implementation' level, there could be many ways to instantiate these values. For instance, our development in Section 4.1 use [0, 2], [0, 1], [0,1] for I, S, and D, respectively. Different representations may result in different desirable properties of similarity measures. We investigate and discuss their differences with our work in Section 4.6. Mathematical definitions for each aspect of preference context are formally defined as follows.

Definition 3.1 (Importance of Concept Name). Let CN be a set of concept names. Then, the *importance of concept name* is a 'partial' function ic: $\mathsf{CN} \rightarrow I$.

For any concepts $A, B \in \mathsf{CN}$, the agent perceives that A has equal or more (less) importance to B if $\mathrm{ic(A)} = i_2$, $\mathrm{ic(B)} = i_1$, and $i_1 \leq_I i_2$ (or $i_2 \leq_I i_1$, respectively) is defined in I.

Example 3.1. Suppose that an agent A is using a similarity measure to compare the degree of equality between concepts London and $\mathsf{Amsterdam}$ w.r.t. safety in the street. In that moment, being 'capital city' may be not relevant to the consideration. Thus, to capture this subjective feeling of proximity, we may suppose I $:= \{i_1, i_2\}$ such that $i_1 \leq_I i_2$. Then, the agent may just assign as: $\mathrm{ic(BigCity)} = i_2$ and $\mathrm{ic(CapitalCity)} = i_1$.

Definition 3.2 (Importance of Role Name). Let RN be a set of role names. Then, the *importance of role name* is a 'partial' function ir: RN $\rightarrow I$.

For any roles $\mathrm{r, s} \in$ RN, the agent perceives that r has equal or more (less) importance to s if $\mathrm{ir(r)} = i_2$, $\mathrm{ir(s)} = i_1$, and $i_1 \leq_I i_2$ (or $i_2 \leq_\mathrm{I} i_1$, respectively) is defined in I.

Example 3.2 (Continuation of Example 3.1). Suppose that an agent A is using a similarity measure to compare the degree of equality between concepts London and $\mathsf{Amsterdam}$ w.r.t. safety in the street. In that moment, the 'isCityOf' relation between a city and a country may be not relevant to the consideration. Thus, to capture this subjective feeling of proximity, we may suppose I $:= \{i_1, i_2\}$ such that $i_1 \leq_\mathrm{I} i_2$. Then, the agent may just assign as: $\mathrm{ir(isCityOf)} = i_1$ and assign i_2 to concepts $\mathsf{CapitalCity}$ and $\mathsf{BigCity}$.

Definition 3.3 (Similarity of Concept Names). Let CN be a set of concept names. Then, the *similarity of concept names* is a 'partial' function sc: $\mathsf{CN} \times \mathsf{CN} \rightarrow S$.

For any concepts $A, B \in \mathrm{CN}$, the agent perceives that A and B present subjective feeling of equality in degree x if $\mathrm{sc}(A, B) = x$ and x is defined in S.

Example 3.3 (Continuation of Example 3.1). Suppose that an agent A feels that England and Netherlands are similar because they are countries in Europe even though they are different countries. To capture this subjective feeling of equality, we may suppose $\mathrm{S} := \{s_1, s_2\}$ such that $s_1 \leq_S s_2$ and assign: $\mathrm{sc}(\mathrm{England,Netherlands}) = s_2$. Other pairs of different concept names are assigned to s_1 by default.

Definition 3.4 (Similarity of Role Names). Let RN be a set of role names. Then, the *similarity of role names* is a 'partial' function sr: $\mathrm{RN} \times \mathrm{RN} \to S$.

For any roles $r, s \in \mathrm{RN}$, the agent perceives that r and s present subjective feeling of equality in degree x if $\mathrm{sr}(r, s) = x$ and x is defined in S.

Example 3.4 (Continuation of Example 3.1). Suppose that concept London is redefined as: $\mathrm{London} \sqsubseteq \mathrm{CapitalCity} \sqcap \mathrm{BigCity} \sqcap \exists \mathrm{isPartOf.England}$. According to this terminology, an agent A feels that the relations isPartOf and isCityOf invoke similar feeling. To capture this subjective feeling of equality, we may suppose $\mathrm{S} := \{s_1, s_2\}$ such that $s_1 \leq_S s_2$ and assign: $\mathrm{sr}(\mathrm{isPartOf,isCityOf}) = s_2$. Other pairs of different role names are assigned to s_1 by default.

Basically, our motivation of both functions sc and sr are the same, *i.e.* we aim at attaching subjective feeling of proximity (about concept names and role names) into a similarity measure. In description logics (or most of other formalisms), different (concept and role) names may refer to different (classes of) instances even though they could be recognized as being similar in real-world domains.

Definition 3.5 (Importance Factor of Quantified Role). Let RN be a set of role names. Then, the *importance factor of quantified role* is a 'partial' function d: $\mathrm{RN} \to D$.

For any role $\mathrm{r} \in \mathrm{RN}$, the agent perceives that, given a quantified concept $\boxdot\ \mathrm{r.C}$ where $\boxdot \in \{\forall, \exists\}$, there is an importance factor x of a quantified role r in relation to the corresponding concept C if $\mathrm{d(r)} = \mathrm{x}$ and x is defined in D.

Intuitively, an agent can employ this notion to distinguish the weighted factor be- tween a quantified role and its corresponding concept description. Indeed, this notion is used when an agent wants to compare two quantified concepts *i.e.* $\boxdot\ \mathrm{r_1.C_1}$ and $\boxdot\ \mathrm{r_2.C_2}$ where $\boxdot \in \{\forall, \exists\}$. We exemplify its usage in the following.

Example 3.5 (Continuation of Example 3.1). Suppose that an agent A feels that the relation isCityOf is less influential than a corresponding concept description, compared to other relations. To capture this subjective feeling of biased weights, we may suppose $D := \{d_1, d_2\}$ such that $d_1 \leq_D d_2$ and assign: $\mathrm{d(isCityOf)} = d_1$. Other role names are assigned to d_2 by default.

The next section discusses our methodology to develop similarity measure w.r.t. preference context for indicating the degree of similarity w.r.t. certain contexts of consideration between two concepts (or states of affairs) in description logics.

3.2 Formal Notion of Context Similarity under Preference

To understand the notion of similarity under subjective factors, we first take a look into the definition of concept equivalence (*cf.* Definition 2.16). That is, let $\mathcal{T}$ be a TBox and $C, D \in \mathrm{Con}(\mathcal{L})$ for a particular DL $\mathcal{L}$, then

$$C \equiv_\tau D \Leftrightarrow C \sqsubseteq_\tau D \text{ and } D \sqsubseteq_\tau C \tag{3.1}$$

It is worth observing that concept equivalence can be seen as an operation for comparing two concepts. For instance, if two concepts are equivalent (*i.e.* $C \equiv_\tau D \Leftrightarrow C^{\mathcal{I}} = D^{\mathcal{I}}$), then the concept equivalence relation yields 1 (true); or yields 0 (false) otherwise. We adopt this viewpoint with preference context and introduce the following.

Definition 3.6. Let $\mathfrak{P}$ be an infinite set of preference contexts where $\mathfrak{p} \in \mathfrak{P}$, $\mathrm{Con}(\mathcal{L})$ be a set of concept descriptions for a particular DL $\mathcal{L}$ where $C, D \in \mathrm{Con}(\mathcal{L})$, and $\mathcal{T}$ be a TBox. Then, a *concept similarity under preferences* is a family of functions $\sim_{\tau}^{\mathfrak{p}}$: Con(CN, RN) $\times$ Con(CN, RN) $\rightarrow$ [0,1] such that

$$\forall \mathfrak{p}' \in \mathfrak{P}: C \sim_{\tau}^{\mathfrak{p}'} D = 1 \iff C \equiv_{\tau} D$$

(called *preference invariant w.r.t. concept equivalence*) holds; and

- $C \sim_{\tau}^{\mathfrak{p}} D = 1$ indicates *maximal similarity* (or concept equivalence) under preference context $\mathfrak{p}$ w.r.t. $\mathcal{T}$ between concept descriptions *C* and *D*,
- $C \sim_{\tau}^{\mathfrak{p}} D = 0$ indicates *having no relation* under preference context $\mathfrak{p}$ w.r.t. $\mathcal{T}$ between concept descriptions *C* and *D.*

The reason we require preference invariance w.r.t. concept equivalence because we do not want to allow the usage of any preference context to effect on the perception of semantically identical concept descriptions.

An interesting question to the above definition is that "how concrete measures for particular DLs should be developed?". We address this question by generalizing from the concept equivalence relation w.r.t. preference context as follows:

$$\mathrm{C} \sim_{\tau}^{\mathfrak{p}} \mathrm{D} = 1 \Leftrightarrow \mathrm{C} \backsim_{\tau}^{\mathfrak{p}} \mathrm{D} = 1 \text{ and } \mathrm{D} \backsim_{\tau}^{\mathfrak{p}} \mathrm{C} = 1 \quad (3.2)$$

That is, two concepts are similar w.r.t. preference context $\mathfrak{p}$ to each other iff the degree of *directional subsumption* w.r.t. preference context p from a concept to another one is 1 and vice versa. In Equation 3.2, the notion of directional subsumption degree w.r.t. preference context is denoted by $\backsim_{\tau}^{\mathfrak{p}}$. It should be noted that the logical conjunction 'and' in the equation should also be generalized in such a way that two numerical values are aggregated and result in a unit interval [0, 1]. Hence, we outline our methodology to develop a 'concrete' similarity measure as follows:

1. Generalize the notion of concept subsumption ($\sqsubseteq_{\tau}$) to the notion of *subsumption degree* ($\backsim_{\tau}$);
2. Generalize the notion of subsumption degree ($\backsim_{\tau}$) to the notion of *subsumption degree under preference* ($\backsim_{\tau}^{\mathfrak{p}}$); and
3. Generalize the logical conjunction (*i.e.* 'and') for aggregating two numerical values to result in a unit interval.

We address the first step in Chapter 3 and the remaining steps in Chapter 4. Basically, the classical reasoning technique is investigated under scrunity in this thesis with the main focus on DL $\mathcal{ELH}$. Our goal is to find out a similarity measure under preferences, which can be computed efficiently *i.e.* in polynomial time. Intuitively, the computational approach introduced in this chapter is derived from the scrunity of structural subsumption approach in $\mathcal{EL}$ (*cf.* Subsubsection 2.2.4).

3.3 From Concept Subsumption to Subsumption Degree

We first discuss a computational approach for identifying the subsumption degree between $\mathcal{ELH}$ concepts. Since $\mathcal{ELH}$ is a superlogic of $\mathcal{EL}$, its structural subsumption procedure can be slightly modified from $\mathcal{EL}$ as follows:

1. Concepts are fully expanded to the form $\mathrm{P}_1 \sqcap \ldots \sqcap \mathrm{P}_\mathrm{m} \sqcap \exists \mathrm{r}_1.\mathrm{C}_1 \sqcap \ldots \sqcap \exists \mathrm{r}_\mathrm{n}.\mathrm{C}_\mathrm{n}$;
2. Fully expanded concepts are structurally transformed into the corresponding description trees, where its root has $\{\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{m}\}$ as its label, has n outgoing edges, each labeled by the set $\mathcal{R}_{r_j}$ of all $\mathrm{r_j}$'s super roles to a vertex $\mathrm{v_j}$ for $1 \leq \mathrm{j} \leq \mathrm{n}$. Formally, $\mathcal{R}_r = \{\mathrm{s} \mid \mathrm{r} \sqsubseteq^* \mathrm{s}\}$ and $\mathrm{r} \sqsubseteq^* \mathrm{s}$ if $\mathrm{r} = \mathrm{s}$ or $\mathrm{r_i} \sqsubseteq \mathrm{r_{i+1}} \in \mathcal{T}$ where $1 \leq \mathrm{i} \leq \mathrm{n}, \mathrm{r}_1 = \mathrm{r}, \mathrm{r_n} = \mathrm{s}$. That is, $\sqsubseteq^*$ denotes a transitive closure of $\sqsubseteq$ between roles. Then, a subtree with the root $\mathrm{v_j}$ is defined recursively relative to the concept $\mathrm{C_j}$; and
3. Given two description trees $\mathcal{T}_C, \mathcal{T}_D$, we conclude that $\mathrm{C} \sqsubseteq \mathrm{D}$ holds iff there exists a homomorphism from $\mathcal{T}_D$ to $\mathcal{T}_C$ according to the following definition and theorem.

Definition 3.7 (Homomorphism (Baader, 2003; Baader et al., 2001)**).** An $\mathcal{ELH}$ description tree $\mathcal{T}$ is a quintuple $(\mathrm{V}, \mathrm{E}, rt, \mathrm{l}, \rho)$ where V is a set of vertices, $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ is a set of edges, rt is the root, $\mathrm{l}: \mathrm{V} \to 2^{\mathrm{CN^{pri}}}$ is a vertex labeling function, and $\rho: \mathrm{E} \to 2^{\mathrm{RN}}$ is an edge labeling function. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two $\mathcal{ELH}$ description trees, $v_1 \in \mathrm{V}_1$ and $v_2 \in \mathrm{V}_2$. Then, the mapping $\mathrm{h}: \mathrm{V}_1 \to \mathrm{V}_2$ is a *homomorphism* from $\mathcal{T}_1$ to $\mathcal{T}_2$ iff the following conditions are satisfied:

- For all $v_1 \in \mathrm{V}_1, \mathrm{l}_1(v_1) \subseteq \mathrm{l}_2(\mathrm{h}(v_1))$; and
- For each successor w_1 of v_1 in $\mathcal{T}_1$, $h(w_1)$ is a successor of $\mathrm{h}(v_1)$ with $\rho_1(v_1, w_1) \subseteq \rho_2(\mathrm{h}(v_1), \mathrm{h}(w_1))$.

Theorem 3.1. Let $\mathrm{C}, \mathrm{D} \in \mathrm{Con}(\mathcal{ELH})$ and $\mathcal{T}_C$ and $\mathcal{T}_D$ be the corresponding description trees. Then, $\mathrm{C} \sqsubseteq \mathrm{D}$ iff there exists a homomorphism (denoted by $\mathrm{h}: \mathcal{T}_D \rightarrow \mathcal{T}_C$) which maps the root v of $\mathcal{T}_D$ to the root w of $\mathcal{T}_C$.

Let us demonstrate how ones can employ the approach to check if subsumption relation holds between $\mathcal{ELH}$ concepts with the following example and shade some light of the possibility to compute the subsumption degree.

Example 3.6. An agent A wants to visit a place for doing some physical activities (*i.e.* ActivePlace). Suppose that a place ontology is modeled as follows. The classical reasoning of subsumption may be used to find out a concept subsumed by ActivePlace.

ActivePlace ⊑ Place ⊓ ∃canWalk.Trekking ⊓ ∃canSail.Kayaking
Mangrove ⊑ Place ⊓ ∃canWalk.Trekking
Beach ⊑ Place ⊓ ∃canSail.Kayaking
canWalk ⊑ canMoveWithLegs
canSail ⊑ canTravelWithSails

Following the above steps, each primitive definition is transformed to a corresponding equivalent full definition and the corresponding description tree is constructed accordingly.

ActivePlace ≡ X ⊓ Place ⊓ ∃canWalk.Trekking ⊓ ∃canSail.Kayaking
Mangrove ≡ Y ⊓ Place ⊓ ∃canWalk.Trekking
Beach ≡ Z ⊓ Place ⊓ ∃canSail.Kayaking

where X, Y, and Z are fresh primitive concept names. canWalk ≡ t ⊓ canMoveWithLegs and canSail ≡ u ⊓ canTravelWithSails, where t and u are fresh primitive role names. In other words, $\mathcal{R}_{\text{canWalk}} = \{\text{t}, \text{canMoveWithLegs}\}$ and $\mathcal{R}_{\text{canSail}} = \{\text{u}, \text{canTravelWithSails}\}$. Figure 3.1a - 3.1c depict $\mathcal{T}_{\text{ActivePlace}}$, $\mathcal{T}_{\text{Mangrove}}$, and $\mathcal{T}_{\text{Beach}}$, respectively.

It is not difficult to find a failed attempt of identifying a homomorphism mapping the root of $\mathcal{T}_{\text{ActivePlace}}$ to the root of $\mathcal{T}_{\text{Mangrove}}$, *i.e.* $h: \mathcal{T}_{\text{ActivePlace}} \nrightarrow \mathcal{T}_{\text{Mangrove}}$. Hence, this infers Mangrove $\not\sqsubseteq$ ActivePlace. Similarly, we can conclude that Beach $\not\sqsubseteq$ ActivePlace.

Though we conclude that subsumption relations from ActivePlace to Mangrove and from ActivePlace to Beach do not hold, we can notice that they have some commonalities among their structures. For instance, considering the roots of $\mathcal{T}_{\text{ActivePlace}}$ and $\mathcal{T}_{\text{Beach}}$,, it appears that Place is belonged to both {X, Place} and {Y, Place}. This observation leads us to develop approaches for computing the subsumption degree between $\mathcal{ELH}$ concepts. In the next subsection, a homomorphism-based structural subsumption degree function is discussed. And, its properties are investigated accordingly.

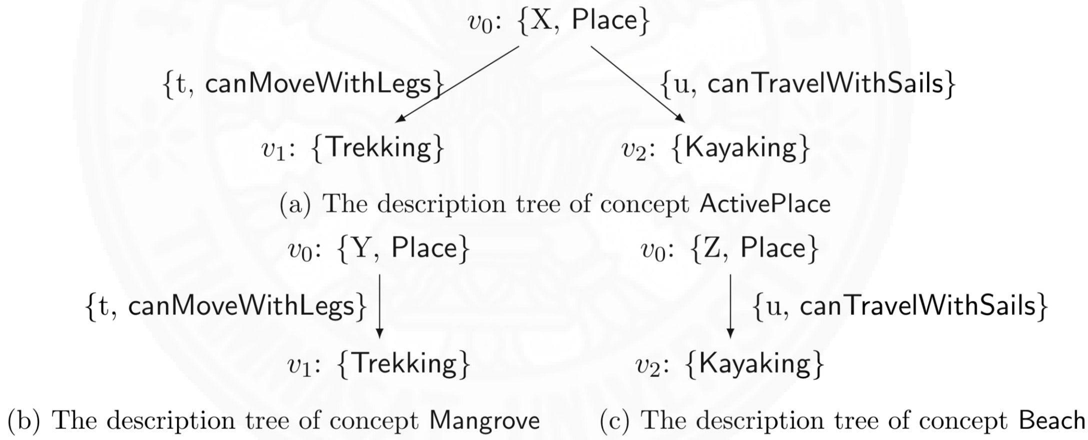

Figure 3.1 The Description Trees of Concepts $\mathcal{T}_{\text{ActivePlace}}$, $\mathcal{T}_{\text{Mangrove}}$, and $\mathcal{T}_{\text{Beach}}$.

3.3.1 Homomorphism Degree

Let us reconsider Example 3.6. It is obvious that $h: \mathcal{T}_{\text{ActivePlace}} \nrightarrow \mathcal{T}_{\text{Mangrove}}$ holds due to {X,Place} $\not\subseteq$ {Y,Place}. However, Place appears to be in common on both sets. Ones may regard this as *partial mapping* from $\mathcal{T}_{\text{ActivePlace}}$ to $\mathcal{T}_{\text{Mangrove}}$. Intuitively, the homomorphism degree function adopts this viewpoint to develop the computational procedure.

Let $\mathrm{C}, \mathrm{D} \in \mathrm{Con}(\mathcal{ELH})$ be fully expanded concept of the form: $\mathrm{P}_1 \sqcap \ldots \sqcap \mathrm{P}_\mathrm{m} \sqcap \exists \mathrm{r}_1.\mathrm{C}_1 \sqcap \ldots \sqcap \exists \mathrm{r}_\mathrm{n}.\mathrm{C}_\mathrm{n}$. We denote the set $\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{m}$ of the concepts C, D and the set

$\exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n$ of the concepts C, D by $\mathcal{P}_C, \mathcal{P}_D$ and $\mathcal{E}_C, \mathcal{E}_D$, respectively. The super roles $\mathcal{R}_r, \mathcal{R}_s$ are as defined on Section 3.3. The following definition extends Theorem 3.1 to the case where no such homomorphism exists but there is some commonality.

Definition 3.8 (Homomorphism Degree (Tongphu & Suntisrivaraporn, 2015)**).** Let $\mathbb{T}^{\mathcal{ELH}}$ be a set of all $\mathcal{ELH}$ description trees and $\mathcal{T}_C, \mathcal{T}_D \in \mathbb{T}^{\mathcal{ELH}}$ correspond to two $\mathcal{ELH}$ concept names C and D, respectively. The *homomorphism degree* function $\text{hd}: \mathbb{T}^{\mathcal{ELH}} \times \mathbb{T}^{\mathcal{ELH}} \rightarrow [0,1]$ is inductively defined as follows:

$$\text{hd}(\mathcal{T}_D, \mathcal{T}_C) = \mu \cdot \text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) + (1-\mu) \cdot \text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C), \tag{3.3}$$

where $\mu = |\mathcal{P}_D| / (|\mathcal{P}_D \cup \mathcal{E}_D|)$ and $|\quad|$ represents the set cardinality;

$$\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = \begin{cases} 1, & \mathcal{P}_D = \emptyset \\ \frac{|\mathcal{P}_D \cap \mathcal{P}_C|}{|\mathcal{P}_D|}, & \text{otherwise} \end{cases} \tag{3.4}$$

$$\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C) = \begin{cases} 1, & \mathcal{P}_D = \emptyset \\ 0, & \mathcal{E}_D \neq \emptyset \text{ and } \mathcal{E}_C = \emptyset \\ \sum_{\epsilon_i \in \mathcal{E}_D} \frac{\max_{\epsilon_j \in \mathcal{E}_C} \{\text{e-hd}(\epsilon_i, \epsilon_j)\}}{|\mathcal{E}_D|}, & \text{otherwise} \end{cases} \tag{3.5}$$

with ϵ_i, ϵ_j existential restrictions; and

$$\text{e-hd}(\exists r.X, \exists s.Y) = \gamma\big(\nu + (1-\nu) \cdot \text{hd}(\mathcal{T}_X, \mathcal{T}_Y)\big) \tag{3.6}$$

where $\gamma = (|\mathcal{R}_r \cap \mathcal{R}_s|) / |\mathcal{R}_r|$ and $0 \leq \nu < 1$.

The value of ν determines how important the roles are to be considered for similarity between two existential restriction information. For instance, ∃canWalk.Trekking and ∃canWalk.Parading for dissimilar nested concepts Trekking and Parading should not be regarded as entirely dissimilar themselves. If ν is assigned the values 0.3, 0.4, and 0.5, then e-hd(∃canWalk.Trekking, ∃canWalk.Parading) is

0.3, 0.4, and 0.5, respectively. This value may vary among applications. In this work, ν is set to 0.4 (if it is not explicitly defined) for exemplifying the calculation of hd.

Example 3.7. (Continuation of Example 3.6) For brevity, let ActivePlace, Mangrove, Beach, Place, Trekking, Kayaking, canWalk, and canSail be abbreviated as AP, M, B, P, T, K, cW, and cS, respectively. Using Definition 3.8, the homomorphism degree from $\mathcal{T}_{\mathrm{AP}}$ to $\mathcal{T}_{\mathrm{M}}$, or

$$\begin{aligned}\mathrm{hd}(\mathcal{T}_{\mathrm{AP}},\mathcal{T}_{\mathrm{M}}) &= \left(\frac{2}{4}\right)\left(\frac{1}{2}\right)+\left(\frac{2}{4}\right)\left(\frac{\max\{\text{e-hd}(\exists \mathrm{cW.T},\exists \mathrm{cW.T})\}}{2}+\frac{\max\{\text{e-hd}(\exists \mathrm{cS.K},\exists \mathrm{cW.T})\}}{2}\right)\\ &= \left(\frac{2}{4}\right)\left(\frac{1}{2}\right)+\left(\frac{2}{4}\right)\left(\frac{1+0}{2}\right)=0.5\end{aligned}$$

Similarly, $\mathrm{hd}(\mathcal{T}_{\mathrm{M}},\mathcal{T}_{\mathrm{AP}}) = 0.67$, $\mathrm{hd}(\mathcal{T}_{\mathrm{AP}},\mathcal{T}_{\mathrm{B}}) = 0.5$, and $\mathrm{hd}(\mathcal{T}_{\mathrm{B}},\mathcal{T}_{\mathrm{AP}}) = 0.67$.

The example shows that the homomorphism degree from $\mathcal{T}_{\mathrm{AP}}$ to $\mathcal{T}_{\mathrm{M}}$ is 0.5 even though M is not subsumed by AP. Similar interpretations can be applied for the other results.

3.3.2 Properties underlying Homomorphism Degree

Theorem 3.2. Let $\mathrm{C}, \mathrm{D} \in \mathrm{Con}(\mathcal{ELH})$ and $\mathcal{T}_C, \mathcal{T}_D$ be their corresponding description tree, respectively. Then, the following are equivalent:

1. $\mathrm{C} \sqsubseteq \mathrm{D}$; and
2. $\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) = 1$

Proof. ($\Longrightarrow$) Assume $\mathrm{C} \sqsubseteq \mathrm{D}$ *i.e.* there exists a homomorphism h which maps the root of $\mathcal{T}_D$ to the root of $\mathcal{T}_C \Leftrightarrow \mathrm{l}_D(v_D) \subseteq \mathrm{l}_C(\mathrm{h}(v_D))$ for each $v_\mathrm{D} \in \mathrm{V}_D$ and $\rho_D(v_D, w_D) \subseteq \rho_C(\mathrm{h}(v_C), \mathrm{h}(w_C))$ for each successor w_D of v_D. We show $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$ by cases.

- When $|\mathrm{V}_D| = 1$ *i.e.* $\mathcal{T}_D$ contains only one node, then we show $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1 \Leftrightarrow$ $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = 1$ (by Definition 3.8). This is obvious since $\mathcal{P}_D \subseteq \mathcal{P}_C$.
- When $|\mathrm{V}_\mathrm{D}| > 1$, then we need to show $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1 \Leftrightarrow \text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = 1$ and $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C) = 1$ (by Definition 3.8). Since $\mathrm{l}_\mathrm{D}(rt_\mathrm{D}) \subseteq \mathrm{l}_\mathrm{C}(\mathrm{h}(rt_\mathrm{D}))$ (by assumption), then $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = |\mathcal{P}_D\ | \ / \ |\mathcal{P}_D| = 1$. To show $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C) = 1$, we need to show that $\rho_D(rt_D, w) \subseteq \rho_C(rt_C, \mathrm{h}(w))$ for

each successor w of rt_D (in order to have $\gamma = 1$) and there exists a homomorphism h' which maps w of its subtree $\mathcal{T}_{D_i}$ to $\mathrm{h}'(\mathrm{h}(w))$ of another subtree $\mathcal{T}_{C_j}$ (in order to have $\mathrm{hd}\left(\mathcal{T}_{D_i}, \mathcal{T}_{C_j}\right) = 1$). The former is obvious by assumption. Since $\mathcal{T}_{D_i}$ is part of $\mathcal{T}_D$ and $\mathcal{T}_{C_j}$ is also part of $\mathcal{T}_C$, then such h' also exists by assumption. Thus, we have $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$.

($\Longleftarrow$) Assume $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$ *i.e.* $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = 1$ and $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C) = 1$ (by Definition 3.8). Then, we need to show $\mathrm{C} \sqsubseteq \mathrm{D}$ *i.e.* there exists a homomorphism which maps the root of $\mathcal{T}_D$ to the root of $\mathcal{T}_C$. By Definition 3.8, $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C) = 1$ implies that $\mathcal{P}_D \subseteq \mathcal{P}_C$. Also, $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C) = 1 \Leftrightarrow \gamma = 1$ and $\mathrm{hd}\left(\mathcal{T}_{D_i}, \mathcal{T}_{C_j}\right) = 1$ for each depth of the tree $\mathcal{T}_D \Leftrightarrow \mathrm{l}_D(v_D) \subseteq \mathrm{l}_C(\mathrm{h}(v_D))$ for each $v_D \in \mathrm{V}_D$ and $\rho_D(v_D, w_D) \subseteq \rho_C(h(v_C), h(w_C))$ for each successor w_D of v_D. Therefore, we conclude that $\mathrm{C} \sqsubseteq \mathrm{D}$.

Theorem 3.2 describes a property of concept subsumption, *i.e.* C is a subconcept of D if the homomorphism degree of the corresponding description tree $\mathcal{T}_D$ to $\mathcal{T}_C$ is equal to 1, and vice versa. In other words, the more value of $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C)$ is closer to 1, the more likely the subsumption of C and D may hold.

In the following, we show that hd can be computed in polynomial time.

Theorem 3.3. Let $\mathrm{V}_1, \mathrm{V}_2$ be sets of vertices corresponding to $\mathcal{T}_1, \mathcal{T}_2$, respectively. The computational complexity of hd is $\mathcal{O}(|\mathrm{V}_1| \cdot |\mathrm{V}_2|)$.

Proof. Let $\mathrm{C} := \mathrm{P}_1 \sqcap \ldots \sqcap \mathrm{P}_\mathrm{m} \sqcap \exists \mathrm{r}_1.\mathrm{C}_1 \sqcap \ldots \sqcap \exists \mathrm{r}_\mathrm{n}.\mathrm{C}_\mathrm{n}$, $\mathrm{D} := \mathrm{Q}_1 \sqcap \ldots \sqcap \mathrm{Q}_\mathrm{l} \sqcap \exists \mathrm{s}_1.\mathrm{D}_1 \sqcap \ldots \sqcap \exists \mathrm{r}_\mathrm{o}.\mathrm{D}_\mathrm{o}$, and $\mathcal{T}_C$, $\mathcal{T}_D$ be the corresponding description trees. We need to show μ, γ, $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C)$, and $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C)$ are bounded by $\mathcal{O}(|V_1| \cdot |V_2|)$.

Since the set union, the intersection, and the set cardinality $|\quad|$ can be computed in polynomial time in the worst case, then μ, $\text{p-hd}(\mathcal{P}_D, \mathcal{P}_C)$, and γ are bounded by $\mathcal{O}(|V_1| \cdot |V_2|)$.

Computing $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C)$ requires to call e-hd for $|\mathcal{E}_D||\mathcal{E}_C|$ times. Each call of e-hd will make a recursive call to hd and its number of calls is bounded by the height of $\mathcal{T}_D$ and $\mathcal{T}_C$. Hence, $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C)$ are bounded by $\mathcal{O}(|V_1| \cdot |V_2|)$.

3.3.3 Concept Similarity Degree

The degree of concept similarity can be determined from the two directional subsumption degree of each corresponding direction. This is an intermediate result of developing a concrete notion of subsumption degree such as the homomorphism degree. Mathematically, the degree of concept similarity can be defined as any binary operators accepting the unit interval *e.g.* the average, the multiplication, and the root mean square.

In the following, a measure for description logic $\mathcal{ELH}$ is defined based on the average of two homomorphism degrees. This measure is named sim and is introduced (Suntisrivaraporn, 2013; Tongphu & Suntisrivaraporn, 2015). We give its formal definition as follows:

Definition 3.9 ($\mathcal{ELH}$ Similarity Degree). Let $\mathrm{C}, \mathrm{D} \in \mathrm{Con}(\mathcal{ELH})$ and $\mathcal{T}_C, \mathcal{T}_D$ be the corresponding description trees. Then, the $\mathcal{ELH}$ similarity degree between C and D (denoted by $\mathrm{sim}(C, D)$) is defined as follows:

$$\mathrm{sim}(C, D) = \frac{\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) + \mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C)}{2} \tag{3.7}$$

Example 3.8. (Continuation of Example 3.7) The $\mathcal{ELH}$ similarity degree between AP and M can be calculated as follows:

$$\mathrm{sim}(\mathrm{AP,M}) = \frac{hd(\mathcal{T}_{\mathrm{AP}}, \mathcal{T}_{\mathrm{M}}) + hd(\mathcal{T}_{\mathrm{M}}, \mathcal{T}_{\mathrm{AP}})}{2} = \frac{0.5 + 0.67}{2} = 0.585$$

Similarly, $\mathrm{sim}(\mathrm{AP,B}) = 0.585$.

Definition 3.10 (Ordering of Functions). Let α and β be different functions. Then, α is more *skeptical than or equal to* β (denoted by $\alpha \preceq \beta$) if $(C\ \alpha\ D) \leq (C\ \beta\ D)$ for all concepts $C, D \in \mathrm{Con}(\mathcal{L})$.

The following proposition discusses about some inherited properties of the above measure for $\mathcal{ELH}$ concepts *i.e.* it is symmetric and is less skeptical than the concept equivalence (*cf.* Definition 3.10).

Proposition 3.1. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$. Then, the following properties hold:

1. $\mathrm{sim}(C, D) = \mathrm{sim}(D, C)$; and
2. $\equiv \preceq \mathrm{sim}$.

Proof. (1) This is obvious by the average.

(2) This is immediately followed from Theorem 3.2 and the average.

Theorem 3.4. Let $\mathrm{V}_1, \mathrm{V}_2$ be sets of vertices corresponding to $\mathcal{T}_1, \mathcal{T}_2$, respectively. The computational complexity of sim is $\mathcal{O}(|\mathrm{V}_1| \cdot |\mathrm{V}_2|)$.

Proof. This is immediately followed from Theorem 3.3 and the average.

We can show that sim is also a procedure which ensures termination and can be used as an indicator for the degree of commonalities between $\mathcal{ELH}$ concepts. Intuitively, we ensure that the correct results are correct (*cf.* Lemma 3.1) and the negative results are also correct (*cf.* Lemma 3.2). Termination guarantees to provide an answer in finite time.

Lemma 3.1. Let C, D be any $\mathcal{ELH}$ concepts and $\nu \in (0,1]$. Then, $\mathrm{sim}(C, D) \in (0,1]$ implies that C and D share commonalities among each other.

Proof. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$, $\mathcal{T}_C, \mathcal{T}_D$ be their corresponding trees, and $\nu \in (0,1)$. With Theorem 3.2 and the average, it suffices to show that $\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) \in (0,1]$ implies $\text{p-hd}(\mathcal{P}_C, \mathcal{P}_D) > 0$ or $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_D) > 0$. We show these cases as follows:

• If there exists $v \in \mathrm{V_C}$ such that $\mathrm{l_C}(v) \cap \mathrm{l_D}(\mathrm{h}(v)) \neq \emptyset$, then we show $\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) \in (0,1]$. Since $|\mathrm{l_C}(v) \cap \mathrm{l_D}(\mathrm{h}(v))| > 0$, then we know $\mu > 0$ and $\text{p-hd}(\mathcal{P}_v, \mathcal{P}_{\mathrm{h}(v)}) > 0$. That is, $\text{p-hd}(\mathcal{P}_C, \mathcal{P}_D) > 0$.

• If there exist $v, w \in \mathrm{V_C}$ such that $\rho_C(v, w) \cap \rho_D(\mathrm{h}(v), \mathrm{h}(w)) \neq 0$, then we show $\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) \in (0,1]$. Since $|\rho_C(v, w) \cap \rho_D(\mathrm{h}(v), \mathrm{h}(w))| > 0$, then we know $\gamma > 0$. Since hd cannot be decreased, we know $\text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_D) > 0$.

Lemma 3.2. Let C,D be $\mathcal{ELH}$ concepts and $\nu \in (0,1]$. Then, if C and D share commonalities among each other, then $\mathrm{sim}(C, D) \in (0,1]$.

Proof. Let $C, D \in \mathsf{Con}(\mathcal{ELH})$, $\mathcal{T}_C, \mathcal{T}_D$ be their corresponding trees, and $\nu \in (0,1)$. We show their contraposition *i.e.* $\mathrm{sim}(C, D) = 0$ implies that C and D do not share commonalities to each other.

By the average, we know that $\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) = 0$ and $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 0$. This means that both C and D do not share any commonalities to each other.

Theorem 3.5. The measure sim is guaranteed for termination and fulfills the condition:

$\mathrm{sim}(C, D) \in (0,1]$ iff both C and D share commonalities among each other.

Proof. This is obvious by Lemma 3.1, Lemma 3.2, and Theorem 3.4.

Discussing about the choice of an aggregating operator, ones may argue to base the definitions on other methods. However, those may create unsatisfactory results for the extreme cases. To illustrate this, we define the functions $\mathrm{sim}_\times$ and $\mathrm{sim}_{\mathrm{rms}}$ as follows.

$$\mathrm{sim}_\times(C, D) = \mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D) \times \mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) \tag{3.8}$$

$$\mathrm{sim}_{\mathrm{rms}}(\mathrm{C}, \mathrm{D}) = \sqrt{\frac{\left(\mathrm{hd}(\mathcal{T}_C, \mathcal{T}_D)\right)^2 + \left(\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C)\right)^2}{2}} \tag{3.9}$$

Then, for any primitive concept A, we have $\mathrm{sim}_\times(\mathrm{A}, \top) = 0 \times 1 = 0$ and $\mathrm{sim}_{\mathrm{rms}}(\mathrm{A}, \top) = \sqrt{(0^2 + 1^2)/2} = 0.707$, whereas $\mathrm{sim}(\mathrm{A}, \top) = (0 + 1)/2 = 0.5$. This implies that $\mathrm{sim}_\times(C, D) \leq \mathrm{sim} \leq \mathrm{sim}_{\mathrm{rms}}(C, D)$ for any concept descriptions C and D. Hence, we agree with (Racharak et al., 2018; Suntisrivaraporn, 2013) that the average-based definition as given above is the most appropriate method[13].

[13] Though we recommend to use the average, its choice of operators may be changed and it may produce a different behavior as discussed.

CHAPTER 4
PERSONALIZING CONCEPT SIMILARITY IN DESCRIPTION LOGICS

The previous chapter proposes an approach for identifying subsumption degree and concept similarity degree in description logics. This produces a number which indicates the commonalities and discrepancies of the subset relation and equivalence relation, respectively, between concepts.

However, not every feature need to be cited in analogical reasoning, the studies in (Hesse, 1965; Waller, 2001; Weinreb, 2016) reported that features used by the comparison should be 'relevant' to the attribution of the property. This leads us to investigate and develop an approach for computing the degree of concept similarity under subjective factors. We illustrate an example (slightly modified from Example 3.6) in which subjective factors play a decisive role in similarity perception.

Example 4.1. An agent A wants to visit a place for doing some physical activities (*i.e.* $\mathsf{ActivePlace}$). At that moment, he would like to enjoy walking. Suppose that a place ontology has been modeled as follows:

$\mathsf{ActivePlace} \sqsubseteq \mathsf{Place} \sqcap \exists \mathsf{canWalk.Trekking} \sqcap \exists \mathsf{canSail.Kayaking}$

$\mathsf{Mangrove} \sqsubseteq \mathsf{Place} \sqcap \exists \mathsf{canWalk.Trekking}$

$\mathsf{Beach} \sqsubseteq \mathsf{Place} \sqcap \exists \mathsf{canSail.Kayaking}$

$\mathsf{canWalk} \sqsubseteq \mathsf{canMoveWithLegs}$

$\mathsf{canSail} \sqsubseteq \mathsf{canTravelWithSails}$

Since the above ontology is expressed in $\mathcal{ELH}$, we may use the measure sim to query the similarity degree between ActivePlace and Beach *i.e.* $\mathsf{sim}(\mathsf{AP},\mathsf{B})$, and also, between ActivePlace and Mangrove *i.e.* $\mathsf{sim}(\mathsf{AP},\mathsf{M})$. As shown in Example 3.8, $\mathsf{sim}(\mathsf{AP},\mathsf{B}) = \mathsf{sim}(\mathsf{AP},\mathsf{M}) = 0.585$. These information shows that both Mangrove and Beach are equally similar to ActivePlace. We note that sim was developed based on the structural subsumption algorithm; thus, it merely considers the objective aspects. Taking into account also the agent's preferences, Mangrove may appear to be more suitable for his perception of ActivePlace at that moment. In other words, he will not be happy if an intelligent system happens to recommend him to go for a Beach.

To address this issue, how preferential aspects of a context in consideration should be properly formalized. As introduced in Chapter 3, preference context is an 'abstract' guideline for devising similarity measure under subjective factors. Our intention for leaving the internal structure of preference context is related to an intuition that methods of developing specific similarity measure can be so many. Indeed, certain concrete notion of preference context may depend on an algorithmic definition of similarity measure. Later, we will exemplify other concrete notion of preference context that we have found in the literature of description logics in this chapter (*cf.* Subsection 4.6.2) and compare them to our well-investigated concrete notion of preference context called *preference profile*. This chapter also generalizes the approach for computing subsumption degree w.r.t. those preferential aspects called concept *similarity under preference profile*.

4.1 Preference Profile

We first introduced preference profile (denoted by π) in (Racharak, Suntisrivaraporn, et al., 2016b) as a collection of preferential elements in which the development of similarity measure of concepts for a particular cognitive agent should consider. Its first intuition is to model different forms of preferences (of an agent) based on concept names and role names. Similarity measure which adopts this notion is flexible to be tuned by an agent and can determine the similarity conformable to that agent's perception.

The syntax and semantics of each form are given in term of 'partial' functions because agents may not have preferences over all concept names and role names. We recommend to devise similarity measures with considerations on preference profile if we aim at developing concept similarity measure for general purposes – a measure based on both subjective and objective factors. Mathematical definitions for each form of preferences are formally defined as follows.

Definition 4.1 (Primitive Concept Importance). Let $\mathsf{CN}^{\mathsf{pri}}(\mathcal{T})$ be a set of primitive concept names occurring in a TBox $\mathcal{T}$. Then, a *primitive concept importance* is a 'partial' function $\mathfrak{i}^{\mathrm{c}}: \mathsf{CN}^{\mathsf{pri}}(\mathcal{T}) \rightarrow [0,2]$[14].

For any A $\in \mathsf{CN}^{\mathsf{pri}}(\mathcal{T})$, $\mathfrak{i}^{\mathrm{c}}(A) = 1$ captures an expression of normal importance for A, $\mathfrak{i}^{\mathrm{c}}(A) > 1$ (and $\mathfrak{i}^{\mathrm{c}}(A) < 1$) indicates that A has higher (and lower, respectively) importance, and $\mathfrak{i}^{\mathrm{c}}(A) = 0$ indicates that A is of no importance to the agent.

Example 4.2. (Continuation of Example 4.1) Suppose that an agent A is using a similarity measure for querying some names similar to ActivePlace. He concerns that those names will be similar to ActivePlace if they are 'places'. Thus, the agent can express this preference as $\mathfrak{i}^{\mathrm{c}}(\text{Place}) = 2$, *i.e.* values should be higher than 1.

On the other hand, suppose he 'does not care' if those are places or not, he may express this preference as $\mathfrak{i}^{\mathrm{c}}(\text{Place}) = 0$, *i.e.* values must be equal to 0.

Definition 4.2 (Role Importance). Let RN$(\mathcal{T})$ be a set of role names occurring in $\mathcal{T}$. Then, a *role importance* is a 'partial' function $\mathfrak{i}^{\mathrm{r}}$:RN$(\mathcal{T}) \rightarrow [0,2]$.

For any $r \in$ RN$(\mathcal{T})$, $\mathfrak{i}^{\mathrm{r}}(r) = 1$ captures an expression of normal importance for r, $\mathfrak{i}^{\mathrm{r}}(r) > 1$ (and $\mathfrak{i}^{\mathrm{r}}(r) < 1$) indicates that r has higher (and lower, respectively) importance, and $\mathfrak{i}^{\mathrm{r}}(r) = 0$ indicates that r is of no importance to the agent.

Example 4.3. (Continuation of Example 4.1) Suppose that the agent A wants to enjoy 'walking'. He may express this preference as $\mathfrak{i}^{\mathrm{r}}(\mathsf{canWalk}) = 2$, *i.e.* values should be higher than 1.

It is worth noticing that, at the concrete level, preference profile makes use of 1 as the special value to represent the normal importance and call other values above 1 as the higher importance (and below 1 as the lower importance).

Definition 4.3 (Primitive Concepts Similarity). Let $\mathsf{CN}^{\mathsf{pri}}(\mathcal{T})$ be a set of primitive concept names occurring in $\mathcal{T}$. For $A, B \in \mathsf{CN}^{\mathsf{pri}}(\mathcal{T})$, a *primitive concepts*

[14] In the original definition of preference profile (Racharak, Suntisrivaraporn, et al., 2016b), elements in the domains of both $\mathfrak{i}^{\mathrm{c}}$ and $\mathfrak{i}^{\mathrm{r}}$ are mapped to $\mathrm{R}_{\geq 0}$, which is a minor error.

similarity is a 'partial' function $\mathfrak{s}^{\mathrm{c}}: \mathrm{CN}^{\mathrm{pri}}(\mathcal{T}) \times \mathrm{CN}^{\mathrm{pri}}(\mathcal{T}) \rightarrow [0,1]$ such that $\mathfrak{s}^{\mathrm{c}}(A, B) = \mathfrak{s}^{\mathrm{c}}(B, A)$ and $\mathfrak{s}^{\mathrm{c}}(A, A) = 1$.

For $A, B \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$, $\mathfrak{s}^{\mathrm{c}}(A, B) = 1$ captures an expression of total similarity between A and B and $\mathfrak{s}^{\mathrm{c}}(A, B) = 0$ captures an expression of their total dissimilarity.

Example 4.4. (Continuation of Example 4.1) Suppose that the agent *A* believes that 'trekking' and 'kayaking' invoke similar feeling. Thus, he can express $\mathfrak{s}^{\mathrm{c}}(\mathrm{Trekking,Kayaking}) = 0.1$, *i.e.* values should be higher than 0.

Another example is the similarity of concepts Pet_1 and Pet_2, in which both are defined as follows: $\mathrm{Pet}_1 \sqsubseteq \mathrm{Dog} \sqcap \exists \mathrm{hasOwned.Human}$; $\mathrm{Pet}_2 \sqsubseteq \mathrm{Cat} \sqcap \exists \mathrm{hasOwned.Human}$. Here, Dog and Cat are both primitive concept names. Intuitively, Dog and Cat are similar, then we may attach this knowledge in form of $\mathfrak{s}^{\mathrm{c}}$ in order to yield more accuracy on the measure.

Definition 4.4 (Primitive Roles Similarity). Let $\mathrm{RN}^{\mathrm{pri}}(\mathcal{T})$ be a set of primitive role names occurring in $\mathcal{T}$. For $\mathrm{r}, \mathrm{s} \in \mathrm{RN}^{\mathrm{pri}}(\mathcal{T})$, a *primitive roles similarity* is a 'partial' function $\mathfrak{s}^{\mathrm{r}}: \mathrm{RN}^{\mathrm{pri}}(\mathcal{T}) \times \mathrm{RN}^{\mathrm{pri}}(\mathcal{T}) \rightarrow [0,1]$ such that $\mathfrak{s}^{\mathrm{r}}(r, s) = \mathfrak{s}^{\mathrm{r}}(s, r)$ and $\mathfrak{s}^{\mathrm{r}}(r, r) = 1$.

For $r, \mathrm{s} \in \mathrm{RN}(\mathcal{T})$, $\mathfrak{s}^{\mathrm{r}}(r, s) = 1$ captures an expression of total similarity between r and s and $\mathfrak{s}^{\mathrm{r}}(r, s) = 0$ captures an expression of their total dissimilarity.

Example 4.5. (Continuation of Example 4.1) Suppose that the agent *A* believes that 'moving with legs' and 'traveling with sails' invoke similar feeling. He may express $\mathfrak{s}^{\mathrm{r}}(\mathrm{canMoveWithLegs,canTravelWithSails}) = 0.1$, *i.e.* values should be higher than 0.

Basically, the intention of both functions $\mathfrak{s}^{\mathrm{c}}$ and $\mathfrak{s}^{\mathrm{r}}$ are the same, *i.e.* they are aimed at capturing subjective feeling of proximity (about primitive concept names and primitive role names) into a measure. In DLs, different primitive concept names (and also primitive role names) are considered to be total dissimilarity even though they may be recognized as being similar in real-world domains.

Definition 4.5 (Role Discount Factor). Let $\mathrm{RN}(\mathcal{T})$ be a set of role names occurring in $\mathcal{T}$. Then, a *role discount factor* is a 'partial' function $\mathfrak{d}$:$\mathrm{RN}(\mathcal{T}) \rightarrow [0,1]$.

Intuitively, role discount factor means a factor that discounts an important contribution of a role. This aspect plays a part when comparing two existential restrictions or two value restrictions, *i.e.* concepts of the form $\exists r. C$ or concepts of the form $\forall \mathrm{r}. \mathrm{C}$, respectively, are being compared. For example, comparing $\exists r_1.(\exists r_2. C_1)$ and $\exists r_3.(\exists r_4. C_2)$ involves checking the commonality of r_1, r_3 and the commonality of $\exists r_2. C_1, \exists r_4. C_2$. Depending on a context of consideration, the commonality appeared in r_1 may have more/less importance than the commonality appeared in its nested concept part *i.e.* $\exists r_2. C_1$.

More formally, for any $r \in \mathrm{RN}(\mathcal{T})$, $\mathfrak{d}(r) = 1$ captures an expression of total importance on the role (beyond a corresponding nested concept) and $\mathfrak{d}(r) = 0$ captures an expression of total importance on a nested concept (beyond the correspondent role r).

Example 4.6. (Continuation of Example 4.1) Suppose that the agent A does not concern much if places permit to either walk or to sail. He would rather consider on actual activities which he can perform. Thus, he may express $\mathfrak{d}(\mathrm{canWalk}) = 0.3$ and $\mathfrak{d}(\mathrm{canSail}) = 0.3$, *i.e.* values should be close to 0.

Definition 4.6 (Preference Profile). A *preference profile*, in symbol π, is a quintuple $\langle \mathfrak{i}^{\mathrm{c}}, \mathfrak{i}^{\mathrm{r}}, \mathfrak{s}^{\mathrm{c}}, \mathfrak{s}^{\mathrm{r}}, \mathfrak{d} \rangle$ where $\mathfrak{i}^{\mathrm{c}}$, $\mathfrak{i}^{\mathrm{r}}$, $\mathfrak{s}^{\mathrm{c}}$, $\mathfrak{s}^{\mathrm{r}}$, and $\mathfrak{d}$ are as defined above and the *default preference profile*, in symbol π_0, is the quintuple $\langle \mathfrak{i}_0^{\mathrm{c}}, \mathfrak{i}_0^{\mathrm{r}}, \mathfrak{s}_0^{\mathrm{c}}, \mathfrak{s}_0^{\mathrm{r}}, \mathfrak{d}_0 \rangle$ where

$\mathfrak{i}_0^{\mathrm{c}}(A) = 1$ for all $A \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$,
$\mathfrak{i}_0^{\mathrm{r}}(r) = 1$ for all $r \in \mathrm{RN}(\mathcal{T})$,
$\mathfrak{s}^{\mathrm{c}}(A, B) = 0$ for all $(A, B) \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T}) \times \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$,
$\mathfrak{s}_0^{\mathrm{r}}(r, s) = 0$ for all $(r, s) \in \mathrm{RN}^{\mathrm{pri}}(\mathcal{T}) \times \mathrm{RN}^{\mathrm{pri}}(\mathcal{T})$, and
$\mathfrak{d}_0(r) = 0.4$ for all $r \in \mathrm{RN}(\mathcal{T})$.

Intuitively, the default preference profile π_0 represents the agent's preference in the default manner, *i.e.* when preferences are not given. That is, every $A \in \mathrm{CN}^{\mathrm{pri}}$ has normal importance and so does every $r \in \mathrm{RN}$. Also, every $(A, B) \in \mathrm{CN}^{\mathrm{pri}} \times \mathrm{CN}^{\mathrm{pri}}$ is totally different and so does every $(r, s) \in \mathrm{RN}^{\mathrm{pri}} \times \mathrm{RN}^{\mathrm{pri}}$. Lastly, every $r \in \mathrm{RN}$ is considered 0.4 importance for the similarity of two existential restriction information (or two value restriction information). It is interesting to note that changes in the definition of the default preference profile yield different interpretations of the default preference and thereby may produce a different degree of similarity under the default manner. As for its exemplification, the value 0.4 is used by $\mathfrak{d}_0$ to conform with the value of ν used by sim in Chapter 3.

In this work, a preference profile of an agent is denoted by subscribing that agent below π, *e.g.* π_{A} represents a preference profile of the agent A.

4.2 From Subsumption Degree to Subsumption Degree under Preferences

Now, we are ready to exemplify how the notion of preference profile can be adopted toward the development of concept similarity under preference profile. Our next step is to generalize the function hd to expose preferential elements of preference profile. As a result, the new function hd^{π} is also driven by the structural subsumption characterization by means of tree homomorphism in $\mathcal{ELH}$.

We start by presenting each aspect of preference profile in term of 'total' functions in order to avoid computing on null values. A *total importance* function is firstly introduced as $\hat{\mathfrak{i}}: \mathrm{CN}^{\mathrm{pri}} \cup \mathrm{RN} \to [0,2]$ based on the primitive concept importance and the role importance.

$$\hat{\mathfrak{i}}(x) = \begin{cases} \mathfrak{i}^{c}(x), x \in \mathrm{CN}^{\mathrm{pri}} \text{ and } \mathfrak{i}^{c} \text{ is defined on } x \\ \mathfrak{i}^{r}(x), x \in \mathrm{RN} \text{ and } \mathfrak{i}^{r} \text{ is defined on } x \\ 1, \text{otherwise} \end{cases} \quad (4.1)$$

A *total similarity* function is also presented as $\hat{\mathfrak{s}}: (\mathrm{CN}^{\mathrm{pri}} \times \mathrm{CN}^{\mathrm{pri}}) \cup (\mathrm{RN}^{\mathrm{pri}} \times \mathrm{RN}^{\mathrm{pri}}) \to [0,1]$ using the primitive concepts similarity and the primitive roles similarity.

$$\hat{\mathfrak{s}}(x,y)=\begin{cases}1, & x=y\\ \mathfrak{s}^{c}(x,y),(x,y)\in\mathrm{CN}^{\mathrm{pri}}\times\mathrm{CN}^{\mathrm{pri}} & \text{and } \mathfrak{s}^{c} \text{ is defined on (x,y)}\\ \mathfrak{s}^{r}(x,y),(x,y)\in\mathrm{RN}^{\mathrm{pri}}\times\mathrm{RN}^{\mathrm{pri}} & \text{and } \mathfrak{s}^{r} \text{ is defined on (x,y)}\\ 0, & \text{otherwise}\end{cases} \quad (4.2)$$

Similarly, a *total role discount factor* function[15] is presented in the following in term of a function $\hat{\mathfrak{d}}$:RN → [0,1] based on the role discount factor.

$$\mathfrak{d}(x)\begin{cases}\mathfrak{d}(x), \text{ if } \mathfrak{d} \text{ is defined on } x\\ 0.4, \quad \text{otherwise}\end{cases} \quad (4.3)$$

The next step is to generalize the notion of homomorphism degree hd (*cf.* Definition 3.8). Let $C, D \in \mathrm{Con}(\mathcal{ELH})$ and $r, s \in$ RN. Also, let $\mathcal{T}_C$, $\mathcal{T}_D$, $\mathcal{P}_C$, $\mathcal{P}_D$, $\mathcal{E}_C$, $\mathcal{E}_D$, $\mathcal{R}_r$, and $\mathcal{R}_s$ be as defined in Subsection 3.3.1. The homomorphism degree under preference profile π from $\mathcal{T}_D$ to $\mathcal{T}_C$ can be formally defined in Definition 4.7.

Definition 4.7. Let $\mathbb{T}^{\mathcal{ELH}}$ be a set of all $\mathcal{ELH}$ description trees, and $\pi = \langle \mathfrak{i}^c, \mathfrak{i}^r, \mathfrak{s}^c, \mathfrak{s}^r, \mathfrak{d} \rangle$ be a preference profile. The *homomorphism degree under preference profile* π is a function $\mathrm{hd}^{\pi}: \mathbb{T}^{\mathcal{ELH}} \times \mathbb{T}^{\mathcal{ELH}} \to [0,1]$ defined inductively as follows:

$$\mathrm{hd}^{\pi}(\mathcal{T}_D,\mathcal{T}_C)=\mu^{\pi}(\mathcal{P}_D,\mathcal{E}_D)\cdot \text{p-hd}^{\pi}(\mathcal{P}_D,\mathcal{P}_C)+\left(1-\mu^{\pi}(\mathcal{P}_D,\mathcal{E}_D)\right)\cdot \text{e-set-hd}^{\pi}(\mathcal{E}_D,\mathcal{E}_C), \quad (4.4)$$

$$\text{where } \mu^{\pi}(\mathcal{P}_D,\mathcal{E}_D)=\begin{cases}1, \sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)+\sum_{\exists r.X\in\mathcal{E}_D}\hat{\mathfrak{i}}(r)=0\\ \dfrac{\sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)}{\sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)+\sum_{\exists r.X\in\mathcal{E}_D}\hat{\mathfrak{i}}(r)}, \text{otherwise}\end{cases} \quad (4.5)$$

$$\text{p-hd}^{\pi}(\mathcal{P}_D,\mathcal{P}_C)=\begin{cases}1, \sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)=0\\ 0, \sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)\neq 0 \text{ and} \sum_{B\in\mathcal{P}_C}\hat{\mathfrak{i}}(B)=0\\ \dfrac{\sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)\cdot\max_{B\in\mathcal{P}_C}\{\hat{\mathfrak{s}}(A,B)\}}{\sum_{A\in\mathcal{P}_D}\hat{\mathfrak{i}}(A)}, \text{ otherwise}\end{cases} \quad (4.6)$$

---

[15] We set the default value to 0.4 to comply with the default value of π_0.

$$\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C) = \begin{cases} 1, \sum_{\exists r.X \in \mathcal{E}_D} \hat{\mathfrak{i}}(r) = 0 \\ 0, \sum_{\exists r.X \in \mathcal{E}_D} \hat{\mathfrak{i}}(r) \neq 0 \text{ and } \sum_{\exists s.Y \in \mathcal{E}_C} \hat{\mathfrak{i}}(s) = 0 \\ \frac{\sum_{\exists r.X \in \mathcal{E}_D} \hat{\mathfrak{i}}(r) \cdot \max_{\epsilon_j \in \mathcal{E}_C} \{\text{e-hd}^{\pi}(\exists r.X, \epsilon_j)\}}{\sum_{\exists r.X \in \mathcal{E}_D} \hat{\mathfrak{i}}(r)}, \text{otherwise} \end{cases} \tag{4.7}$$

where ϵ_j is an existential restriction; and

$$\text{e-hd}^{\pi}(\exists r.X, \exists s.Y) = \gamma^{\pi}(r,s) \cdot \Big(\mathfrak{d}(r) + \big(1 - \mathfrak{d}(r)\big) \cdot \text{hd}^{\pi}(\mathcal{T}_X, \mathcal{T}_Y)\Big) \tag{4.8}$$

$$\text{where } \gamma^{\pi}(\text{r}, \text{s}) = \begin{cases} 1, \sum_{r' \in \mathcal{R}_r} \hat{\mathfrak{i}}(r') = 0 \\ \frac{\sum_{r' \in \mathcal{R}_r} \hat{\mathfrak{i}}(r') \cdot \max_{s' \in \mathcal{R}_s} \{\hat{\mathfrak{s}}(r', s')\}}{\sum_{r' \in \mathcal{R}_r} \hat{\mathfrak{i}}(r')}, \text{otherwise} \end{cases} \tag{4.9}$$

Intuitively, the function hd^{π} (Equation 4.4) is defined as the weighted sum of the degree under preferences of the vertex set commonalities (p-hd^{π}) and the degree under preferences of edge condition matching (e-set-hd^{π}). Equation 4.6 calculates the average of the best matching under preferences of primitive concepts in $\mathcal{P}_D$. Equation 4.8 calculates the degree under preferences of a potential homomorphism of a matching edge. If edge labels share some commonalities under preferences (Equation 4.9), *i.e.* $0 < \gamma^{\pi} \leq 1$, then part of the edge matching is satisfied; but the successors labels and structures have yet to be checked. This is defined recursively as $\text{hd}^{\pi}(\mathcal{T}_X, \mathcal{T}_Y)$ in Equation 4.8. Equation 4.7 calculates the best possible edge matching under preferences of each edge in $\mathcal{E}_D$ and returns the average thereof.

The weight μ^{π} in Equation 4.4 determines how important the primitive concept names are to be considered for preference-based similarity. For the special case where $D = \top$, *i.e.* $\mathcal{P}_D = \mathcal{E}_D = \emptyset$, μ^{π} is irrelevant as $\mathcal{T}_\top$ is the smallest $\mathcal{ELH}$ description tree and $\text{hd}^{\pi}(\mathcal{T}_\top, \mathcal{T}_C) = 1$ for all concepts C.

It is to be mentioned that the function hd^{π} may look similar to $simi_{\text{d}}$ (Lehmann & Turhan, 2012) as both are recursive definitions for the same DL $\mathcal{ELH}$. However, they are obviously different caused by the distinction of their inspirations and their viewpoints of the development. While hd^{π} is inspired by the homomorphism-based structural subsumption characterization, $simi_d$ is inspired by the Jaccard Index

(Jaccard, 1901). Technically speaking, $simi_d$ employs t-conorm instead of fixing an operator. However, unlike $simi_d$, the use of μ^{π} for determining how primitive concepts are weighted and the use of γπ for determining the proportion of shared super roles are employed. Furthermore, $simi_d$, is originated from the viewpoint of ordinary concept similarity measure, thus some aspects of preference profile are missed; though some may exist. We continue the discussion in Section 4.6.

The function hd^{π} yields a numerical value that represents structural similarity w.r.t. a particular profile π of a concept against another concept. We present an example about the calculation of hd^{π} in the following.

Example 4.7. (Continuation of Example 4.1) Let enrich the example. Assume the agent A's preference profile is defined as follows: (i) $\mathfrak{i}^{\mathrm{c}}(\text{Place}) = 2$; (ii) $\mathfrak{i}^{\mathrm{r}}(\text{canWalk}) = 2$; (iii) $\mathfrak{s}^{\mathrm{c}}(\text{Trekking,Kayaking}) = 0.1$; (iv) $\mathfrak{s}^{\mathrm{r}}(\text{canMoveWithLegs,canTravelWithSails}) = 0.1$; (v) $\mathfrak{d}(\text{canWalk}) = 0.3$ and $\mathfrak{d}(\text{canSail}) = 0.3$. Let ActivePlace, Mangrove, Beach, Place, Trekking, Kayaking, canWalk, and canSail are rewritten shortly as AP, M, B, P, T, K, cW, and cS, respectively. Using Definition 4.7, $\mathrm{hd}^{\pi}(\mathcal{T}_{\mathrm{AP}}, \mathcal{T}_{\mathrm{M}})$

$$= \left(\frac{3}{6}\right) \cdot \text{p-hd}^{\pi}(\mathcal{P}_{\mathrm{AP}}, \mathcal{P}_{\mathrm{M}}) + \left(\frac{3}{6}\right) \cdot \text{e-set-hd}^{\pi}(\mathcal{E}_{\mathrm{AP}}, \mathcal{E}_{\mathrm{M}})$$

$$= \left(\frac{3}{6}\right) \cdot \left(\frac{\mathfrak{i}(\mathrm{X}) \cdot \max\{\mathfrak{s}\,(X, Y), \mathfrak{s}(X,\mathrm{P})\} + \mathfrak{i}(\mathrm{P}) \cdot \max\{\,\mathfrak{s}(\mathrm{P}, Y), \mathfrak{s}(\mathrm{P},\mathrm{P}\})\})}{\mathfrak{i}(X) + \mathfrak{i}(\mathrm{P})}\right)$$

$$+ \left(\frac{3}{6}\right) \cdot \text{e-set-hd}^{\pi}(\mathcal{E}_{\mathrm{AP}}, \mathcal{E}_{\mathrm{M}})$$

$$= \left(\frac{3}{6}\right)\left(\frac{1 \cdot \max\{\,0{,}0\} + 2 \cdot \max\{\,0{,}1\}}{1 + 2}\right) + \left(\frac{3}{6}\right) \cdot \text{e-set-hd}^{\pi}(\mathcal{E}_{\mathrm{AP}}, \mathcal{E}_{\mathrm{M}})$$

$$= \left(\frac{3}{6}\right)\left(\frac{2}{3}\right) + \left(\frac{3}{6}\right)\left[\frac{\mathfrak{i}(\mathrm{cW}) \cdot \max\{\,\text{e-hd}^{\pi}(\exists \mathrm{cW.T}, \exists \mathrm{cW.T})\} + 1 \cdot \max\{\,0.019\}}{\mathfrak{i}(\mathrm{cW}) + \mathfrak{i}(\mathrm{cS})}\right]$$

$$= \left(\frac{3}{6}\right)\left(\frac{2}{3}\right) + \left(\frac{3}{6}\right)\left[\frac{2 \cdot \max\{\,(1)\big(0.3 + 0.7(1)\big)\} + 1 \cdot \max\{\,0.019\}}{\mathfrak{i}(\mathrm{cW}) + \mathfrak{i}(\mathrm{cS})}\right]$$

$$= \left(\frac{3}{6}\right)\left(\frac{2}{3}\right) + \left(\frac{3}{6}\right)\left[\frac{(2)(1) + (1)(0.019)}{2 + 1}\right] \approx 0.67$$

Similarly, we obtain $\mathrm{hd}^{\pi}(\mathcal{T}_{\mathrm{M}}, \mathcal{T}_{\mathrm{AP}}) = 0.80$. Furthermore, using Definition 4.7, $\mathrm{hd}^{\pi}(\mathcal{T}_{\mathrm{AP}}, \mathcal{T}_{\mathrm{B}}) \approx 0.51$ and $\mathrm{hd}^{\pi}(\mathcal{T}_{\mathrm{B}}, \mathcal{T}_{\mathrm{AP}}) = 0.75$.

The function hd^{π} can be used when preferences of the agent are not given. That is, we tune the function according to the default preference profile *i.e.* hd^{π_0} . We state this property in the following proposition.

Proposition 4.1. For $\mathcal{T}_D, \mathcal{T}_C \in \mathbb{T}^{\mathcal{ELH}}$, $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$.

Proof. Recall by Definition 4.6 that the default preference profile π_0 is the quintuple $\langle \mathfrak{i}_0^{\mathfrak{c}}, \mathfrak{i}_0^{\mathfrak{r}}, \mathfrak{s}_0^{\mathfrak{c}}, \mathfrak{s}_0^{\mathfrak{r}}, \mathfrak{d}_0 \rangle$. Also, suppose a concept name D is of the form: $P_1 \sqcap \ldots \sqcap P_{\mathrm{m}} \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_{\mathrm{n}}.D_{\mathrm{n}}$ where $P_i \in \mathrm{CN}^{\mathrm{pri}}$, $r_j \in \mathrm{CN}$, $D_j \in \mathrm{Con}(\mathcal{ELH})$, $1 \le i \le m$, $1 \le j \le n$, $P_1 \sqcap \ldots \sqcap P_{\mathrm{m}}$ is denoted by $\mathcal{P}_D$, and $\exists r_1.D_1 \sqcap \ldots \sqcap \exists r_{\mathrm{n}}.D_{\mathrm{n}}$ is denoted by $\mathcal{E}_D$. Let d be the depth of $\mathcal{T}_D$. We prove that, for any $d \in \mathrm{N}$, $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$ by induction on d.

When $d = 0$, we know that $D = P_1 \sqcap \ldots \sqcap P_m$. To show that $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$, we need to show that $\mu^{\pi_0} = \mu$ and $\text{p-hd}^{\pi_0}(\mathcal{P}_D, \mathcal{P}_C) = \text{p-hd}(\mathcal{P}_D, \mathcal{P}_C)$. Let us derive as follows:

$$\mu^{\pi_0} = \frac{\sum_{A \in \mathcal{P}_D} \hat{\mathfrak{i}}(A)}{\sum_{A \in \mathcal{P}_D} \hat{\mathfrak{i}}(A) + \sum_{\exists r.X \in \mathcal{E}_D} \hat{\mathfrak{i}}(r)} = \frac{\sum_{i=1}^{m} 1}{\sum_{i=1}^{m} 1 + 0} = \frac{m}{m+0} = \mu.$$

Furthermore, we only need to show $\sum_{A \in \mathcal{P}_D} \max\{\hat{\mathfrak{s}}(A, B) : B \in \mathcal{P}_C\} = |\mathcal{P}_D \cap \mathcal{P}_C|$ in order to show $\text{p-hd}^{\pi_0}(\mathcal{P}_D, \mathcal{P}_C) = \text{p-hd}(\mathcal{P}_D, \mathcal{P}_C)$. We know that $\mathfrak{s}_0^{\mathfrak{c}}$ maps name identity to 1 and otherwise to 0. Thus, $\sum_{A \in \mathcal{P}_D} \max\{\hat{\mathfrak{s}}(A, B) : B \in \mathcal{P}_C\} = |\{x : x \in \mathcal{P}_D \text{ and } x \in \mathcal{P}_C\}| = |\mathcal{P}_D \cap \mathcal{P}_C|$.

We must now prove that if $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$ holds for $d = h - 1$ where $h > 1$ and $D = P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_n.D_n$ then $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$ also holds for $d = h$. To do that, we have to show $\text{e-set-hd}^{\pi_0}(\mathcal{E}_D, \mathcal{E}_C) = \text{e-set-hd}(\mathcal{E}_D, \mathcal{E}_C)$. This can be done by showing in the similar manner that $\gamma^{\pi_0} = \gamma$ and $\mathrm{hd}^{\pi_0}(\mathcal{T}_X, \mathcal{T}_Y) = \mathrm{hd}^{\pi}(\mathcal{T}_X, \mathcal{T}_Y)$ from $\text{e-hd}^{\pi_0}(\exists r.X, \exists s.Y) = \text{e-hd}(\exists r.X, \exists s.Y)$, where $\exists r.X \in \mathcal{E}_D$ and $\exists s.Y \in \mathcal{E}_C$. Consequently, it follows by induction that, for $\mathcal{T}_D, \mathcal{T}_C \in \mathbb{T}^{\mathcal{ELH}}$, $\mathrm{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$.

4.3 Concept Similarity under Preference Profile

In this section, we present a 'general' notion of *concept similarity measure under the agent's preferences* (Racharak et al., 2018; Racharak, Suntisrivaraporn, et al., 2016a) and its desirable properties. This notion can be seen as a function of the family $\sim_{\mathcal{T}}^{\mathrm{p}}$ given in Definition 3.6. As we shall see, the previous developments on subsumption degree under preference profile can be utilized to develop concrete measures of this abstract notion. Hence, the measures sim^{π} is introduced by utilizing the function hd^{π}. Our first intuition is to exemplify the applicability of preference profile onto an arbitrary existing measure of concept similarity. This shows that our proposed notion of preference profile can be considered as a collection of noteworthy aspects for the development of concept similarity measure under the agent's preferences. Furthermore, it is obvious that preference profile can be seen as a concrete notion of preference context.

Definition 4.8. Given a preference profile π, two concepts $C, D \in \mathrm{Con}(\mathcal{L})$, and a TBox $\mathcal{T}$, a concept similarity measure under preference profile w.r.t. a TBox $\mathcal{T}$ is a function $\sim_{\mathcal{T}}^{\pi}$:$\mathrm{Con}(\mathcal{L}) \times \mathrm{Con}(\mathcal{L}) \to [0,1]$.

When a TBox $\mathcal{T}$ is clear from the context, we simply write $\sim^{\pi}$. Furthermore, to avoid confusion on the symbols, $\sim_{\mathcal{T}}^{\pi}$ is used when referring to arbitrary measures.

The notion $\sim^{\pi}$ may be informally read as "the computation of ~ is influenced by π". That informal interpretation shapes our intuition to consider this kind as a more generalized concept similarity *i.e.* not only objective factors but also subjective factors are considered in the identification of the degree of similarity. With adopting of this viewpoint of the interpretation, we can agree that sim^{π} is informally interpreted as "we compute sim under an existence of a given preference profile π".

Basically, the notion $\sim^{\pi}$ is a function mapping a pair of two concept descriptions w.r.t. a particular π to a unit interval. We have identified a property called *preference invariance w.r.t. equivalence* in our preliminary study (Racharak, Suntisrivaraporn, et al., 2016a). To identify more important properties of $\sim^{\pi}$, we started by investigating important properties of concept similarity measure existing in the literature (*e.g.* (D'Amato, Staab, & Fanizzi, 2008; Lehmann & Turhan, 2012)). Our primary motivation is to identify the properties of concept similarity measure which are also reasonable for $\sim^{\pi}$. The following collects fundamental properties for the introduced concept similarity measure under preference profile. They can be used to answer the question "What could be good preference-based similarity measures?". In other words, any preference-based measures satisfying the fundamental properties are considered to be good ones.

Definition 4.9. Let $C, D, E \in \mathsf{Con}(\mathcal{L})$ and Π be a countably infinite set of preference profile. Then, we call a concept similarity measure under preference profile $\sim^{\pi}$ is:

1. *Symmetric* iff $\forall \pi' \in \Pi: \left(C \sim^{\pi'} D = D \sim^{\pi'} C\right)$;
2. *Equivalence invariant* iff $C \equiv D \Longrightarrow \forall \pi' \in \Pi: \left(C \sim^{\pi'} E = D \sim^{\pi'} E\right)$;
3. *Structurally dependent* iff for any finite sets of concepts C_1 and C_2 with the following conditions:
   a. $\mathrm{C}_1 \subseteq \mathrm{C}_2$,
   b. Concepts $A, B \notin \mathrm{C}_2$,
   c. $\mathfrak{i}^{c}(\Phi) > 0$ if Φ is primitive and $\Phi \in \mathrm{C}_2$, and
   d. $\mathfrak{i}^{r}(\varphi) > 0$ if Φ is existential, *i.e.* $\Phi := \exists \varphi. \Psi$, and $\Phi \in \mathrm{C}_2$,

   the concepts $C := \sqcap(\mathrm{C}_1 \cup \{A\})$, $D := \sqcap(\mathrm{C}_1 \cup \{B\})$, $E := \sqcap(\mathrm{C}_2 \cup \{A\})$ and $F := \sqcap(\mathrm{C}_2 \cup \{B\})$ fulfill the condition $\forall \pi' \in \Pi: \left(C \sim^{\pi'} D \leq E \sim^{\pi'} F\right)$; and
4. *Preference invariant w.r.t. equivalence* iff $C \equiv D \Leftrightarrow \forall \pi' \in \Pi: C \sim^{\pi'} D = 1$.

Next, we discuss the underlying intuitions of each property subsequently. We note that the properties 1 to 3 are adopted from (D'Amato et al., 2008; Lehmann & Turhan, 2012). However, to the best of our knowledge, the property 4 is first introduced for concept similarity measure under preference profile in this work (originally introduced in (Racharak, Suntisrivaraporn, et al., 2016a)).

Let Π be a countably infinite set of preference profile. In the following, we discuss the intuitive interpretation of each property. Firstly, *symmetry* states that an order of concepts in question does not influence the notion π' for any $\pi' \in \Pi$. For instance, $\mathsf{Mangrove} \sim^{\pi'} \mathsf{Beach} = \mathsf{Beach} \sim^{\pi'} \mathsf{Mangrove}$ w.r.t. any particular context

π'. This property is controversial since cognitive science believes that similarity is asymmetric. An example given in (Tversky, 1977) is as follows: People usually speak "the son resembles the father" rather than "the father resembles the son". Some work in DLs also prefer asymmetry such as (Janowicz, 2006; Janowicz & Wilkes, 2009). It is worth observing that such a statement is made w.r.t. some particular contexts. Thus, this work favors on symmetry as it appears more natural to use and gives more intuitive computational understanding. For example, rather than viewing like "the son resembles the father", we would view like "if certain contexts are fixed, then the son and the father are similar to each other" (*cf.* (Racharak & Suntisrivaraporn, 2015; Racharak et al., 2018; Racharak, Suntisrivaraporn, et al., 2016a)). Furthermore, we agree on the symmetry because axiomatic information in TBox is not dynamically changed; and also, the notion of preference profile studied in this work is static, *i.e.* it can be changed merely by tuning. Some work in DLs which favors on symmetry includes (Borgida & Walsh, n.d.; d'Amato, Fanizzi, & Esposito, 2009; D'Amato, Fanizzi, & Esposito, 2006; D'Amato et al., 2008; Fanizzi & D'Amato, n.d.; Lehmann & Turhan, 2012; Racharak & Suntisrivaraporn, 2015; Racharak, Suntisrivaraporn, et al., 2016a; Tongphu & Suntisrivaraporn, 2015).

Secondly, *equivalence invariance* (alternatively called *equivalence soundness* (D'Amato et al., 2008) in the context of dissimilarity measure) states that if two concepts C and D are logically equivalent, then measuring the similarity of each toward the third concept E w.r.t. any $\pi' \in \Pi$ must be the same. This property is inspired from a characteristics of synonym concepts, *i.e.* concepts that means exactly the same. For instance, let $C \equiv \exists \mathsf{canWalk.Trekking}$ and $D \equiv \exists \mathsf{canWalk.Trekking}$. It is clear that C and D are logically equivalent. Therefore, let $E \in \mathsf{Con}(\mathcal{L}), C \sim^{\pi'} E = D \sim^{\pi'} E$ for any $\pi' \in \Pi$.

Thirdly, the notion of *structural dependence* was originally introduced by Tversky in (Tversky, 1977). Later, the authors of (Lehmann & Turhan, 2012) has collected it as another important properties for concept similarity measure in their work. Basically, in Tversky's model, an object was considered as a set of features. Then, the similarity of two objects was measured by the relationship between a number of common features and a number of different features. Extending this idea to $\sim^{\pi}$ gives the meaning that the similarity of two concepts C, D increases if a more number of 'equivalent' concepts is shared and each is considered 'important'.

Lastly, *preference invariance w.r.t. equivalence* states that if two concepts are logically equivalent, then the similarity degree of two concepts under preference profile π is always 1 for every $\pi \in \Pi$, and vice versa. Taking the negation both sides, this means $C \not\equiv D \Leftrightarrow \exists \pi' \in \Pi: C \sim^{\pi'} D \neq 1$. For instance, let $C \equiv \exists \text{canWalk.Trekking}$ and $D \equiv \exists \text{canWalk.Parading}$. It is clear that C and D are not logically equivalent, then taking $\pi = \pi_0$ obtains $C \sim^{\pi_0} D \neq 1$; though, taking $\pi = \pi_1$ where $\mathfrak{s}^c(\text{Trekking,Parading}) = 1$ is defined in π_1 yields $C \sim^{\pi_1} D = 1$.

There are several properties which are not considered as fundamental properties of concept similarity measure under preference profile because the behaviors may not obey their properties when used under 'non-default' preference profiles, *e.g.* reverse subsumption preserving. According to (Lehmann & Turhan, 2012), a concrete measure $\sim$ satisfies the reverse subsumption preserving iff, for any concepts C, D, and E, $C \sqsubseteq D \sqsubseteq E \Longrightarrow C \sim E \leq D \sim E$. The property states that the similarity of D and E is higher than the one of C and E because E is closer to D than C. To refute it, we need only one preference profile π such that the implication does not hold (*cf.* Example 4.8), i.e. to show that $(C \sqsubseteq D \sqsubseteq E)$ and $\exists \pi' \in \Pi: (C \sim^{\pi'} E > D \sim^{\pi'} E$.

Example 4.8. Suppose concepts A_1, A_2, A_3, and A_4 are primitive. Query describes features of an item that an agent is searching for. Item_1 and Item_2 are items, which compose of features $\text{A}_1, \text{A}_2, \text{A}_3$ and $\text{A}_1, \text{A}_2, \text{A}_3, \text{A}_4$, respectively.

$\text{Query} \equiv \text{A}_1 \sqcap \text{A}_2$
$\text{Item}_1 \equiv \text{A}_1 \sqcap \text{A}_2 \sqcap \text{A}_3$
$\text{Item}_2 \equiv \text{A}_1 \sqcap \text{A}_2 \sqcap \text{A}_3 \sqcap \text{A}_4$

The ontology shows the hierarchy: $\text{Item}_2 \sqsubseteq \text{Item}_1 \sqsubseteq \text{Query}$. By taking $\mathfrak{s}^c(\text{A}_2, \text{A}_4) = 1$, it is reasonable to conclude that $\text{Item}_2 \sim^{\pi} \text{Query} > \text{Item}_1 \sim^{\pi} \text{Query}$ due to an increased number of totally similar concepts.

Our proceeding paper (Racharak & Suntisrivaraporn, 2015) studies CSM for the DL $\mathcal{FL}_0$. In this paper, we suggest two measures, *viz.* the skeptical measure $\sim^s$ and the credulous measure $\sim^c$, which are derived from the known structural characterization subsumption through inclusion of regular languages. This fact exhibits that there is no a unique CSM for similarity-based applications. Which CSMs should be used depends on concrete applications, especially the type of a rational agent. For example, when employing the notion $\sim$ to a query answering system, a credulous agent may want to see answers as much as possible; hence, the measure $\sim^c$ is employed. On the other hand, a skeptical agent would like to see sufficient relevant answers; hence, the measure $\sim^s$ is employed. This idea is generalized and is extended toward the notion $\sim^\pi$ to be used under different agent's profiles.

Definition 4.10. Let Π be a countably infinite set of preference profile and $\pi_1, \pi_2 \in \Pi$. For any fixed measure $\sim^\pi$, the concept similarity measure under π_1 is more skeptical than π_2 (denoted by $\sim^{\pi_1} \preceq \sim^{\pi_2}$) if $C \sim^{\pi_1} D \leq C \sim^{\pi_2} D$ for all $C, D \in \mathsf{Con}(\mathcal{L})$.

Intuitively, if an arbitrary concept similarity measure under preference profile $\sim^\pi$ is fixed, measuring the similarity of two concepts under different preference profiles may yield different values. A similar experiment was done in (Bernstein, Kaufmann, Bürki, & Klein, 2005) where different measures were used in target ontologies and obtained the better results than just using a single measure.

4.3.1 From Subsumption Degree under Preferences to Concept Similarity under Preferences

The idea of developing 'concrete' concept similarity measures under preference profile can be analogously brought from concept similarity. Indeed, we have pointed out this in Section 3.2, *i.e.* the second and the third step of our outlined methodology (cf. Section 3.2). We formally recast this in the following.

$$C \sim^\pi_\tau D = 1 \Leftrightarrow C \rightsquigarrow^\pi_\tau D = 1 \text{ and } D \rightsquigarrow^\pi_\tau C = 1 \qquad (4.10)$$

where the notion of directional subsumption degree under preference profile is denoted by $\rightsquigarrow^\pi_\tau$ and the binary operator 'and' should be generalized to aggregate two unit intervals. In the following, we show how ones can employ this idea to develop concrete concept similarity measure under preference profile for DL $\mathcal{ELH}$ in this subsection.

The function hd^{π} yields a numerical value that represents structural similarity w.r.t. a particular profile π of a concept against another concept. We can use this knowledge to develop a concrete measure of $\mathcal{ELH}$ concepts as follows.

Definition 4.11. Let $C, D \in \mathsf{Con}(\mathcal{ELH})$, $\mathcal{T}_C$ and $\mathcal{T}_D$ be the corresponding description trees, and $\pi = \langle \mathfrak{i}^c, \mathfrak{i}^r, \mathfrak{s}^c, \mathfrak{s}^r, \mathfrak{d} \rangle$ be a preference profile. Then, the $\mathcal{ELH}$ similarity measure under preference profile π between C and D (denoted by $\mathrm{sim}^{\pi}(C, D)$) is defined as follows:

$$\mathrm{sim}^{\pi}(C, D) = \frac{\mathrm{hd}^{\pi}(\mathcal{T}_C, \mathcal{T}_D) + \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)}{2} \tag{4.11}$$

Example 4.9. (Continuation of Example 4.7) Using Definition 4.11, it yields that

$$\mathrm{sim}^{\pi}(\mathrm{M,AP}) = \frac{0.67 + 0.80}{2} \approx 0.74$$

Similarly, $\mathrm{sim}^{\pi}(\mathrm{B,AP}) \approx 0.63$. The fact that $\mathrm{sim}^{\pi}(\mathrm{M,AP}) > \mathrm{sim}^{\pi}(\mathrm{B,AP})$ corresponds with the agent A's needs and preferences.

The above definition uses the average to aggregate two corresponding unit intervals. We may also argue to aggregate both values based on alternative operators accepting unit intervals *e.g.* the multiplication or the root mean square of both values. Un- fortunately, those give unsatisfactory values for the extreme cases. Similar arguments about this point has been discussed on Subsection 3.3.3. Hence, we believe that the average-based definition given above is the most appropriate method for aggregating two values of subsumption degree under preference profile. Based on this form, simπ is basically considered as a generalization of sim, which determines similarity under preference profile, *i.e.* behavioral expectation of the measure will conform to the agent's perception. We note that, though we recommend to use the average, its choice of operators may be changed and it may produce a different behavior. The following discusses some inherited properties of the measures sim^{π}.

First, sim^{π} can be used in the case that a preference profile is not defined by the agent. In such a case, we tune the profile setting to π_0. That is, computing sim^{π_0} yields the degree of concept similarity measure merely w.r.t. the structure of concept descriptions in question.

Theorem 4.1. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$, $\mathrm{sim}^{\pi_0}(C, D) = \mathrm{sim}(C, D)$.

Proof. It immediately follows from Lemma 4.1, Definition 3.9, and Definition 4.11.

The above theorem shows that $\mathrm{sim}^{\pi}$is also backward compatible in the sense that using sim^{π} with $\pi = \pi_0$, *i.e.* sim^{π_0}, coincides with sim.

In the following, we show that the measure sim^{π} can also be computed in polynomial time *i.e.* there exists an algorithmic procedure whose execution time is upper bounded by a polynomial expression in the size of the description trees

Theorem 4.2. Assume that a value from any preference functions is retrieved in $\mathcal{O}(1)$. Given $C, D \in \mathrm{Con}(\mathcal{ELH})$, $\mathrm{sim}^{\pi}(C, D) \in \mathcal{O}(|V_C| \cdot |V_D|)$ where V_C and V_D are set of vertices of the description trees $\mathcal{T}_C$ and $\mathcal{T}_D$, respectively.

Proof. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$, $C := P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n$, $D := Q_1 \sqcap \ldots \sqcap Q_l \sqcap \exists s_1.D_1 \sqcap \ldots \sqcap \exists r_o.D_o$, π be any preference profile, and $\mathcal{T}_C, \mathcal{T}_D$ be corresponding description trees. By Definition 4.11, we show $\mathrm{hd}^{\pi}(\mathcal{T}_C, \mathcal{T}_D) \in \mathcal{O}(|V_C| \cdot |V_D|)$ and $\mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C) \in \mathcal{O}(|V_D| \cdot |V_C|)$. Without loss of generality, it suffices to show merely $\mathrm{hd}^{\pi}(\mathcal{T}_C, \mathcal{T}_D) \in \mathcal{O}(|V_C| \cdot |V_D|)$. That is, we need to show μ^{π}, γ^{π}, $\text{p-hd}^{\pi}(\mathcal{P}_D, \mathcal{P}_C)$, and $\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C)$ are bounded by $\mathcal{O}(|V_C| \cdot |V_D|)$.

Since the summation, the maximal matching between $\mathcal{P}_D$ and $\mathcal{P}_C$, and the maximal matching between $\mathcal{R}_r$ and $\mathcal{R}_s$ can be computed in polynomial time in the worst case, the functions μ^{π}, γ^{π}, and $\text{p-hd}^{\pi}(\mathcal{P}_D, \mathcal{P}_C)$ are bounded by $\mathcal{O}(|V_C| \cdot |V_D|)$.

Computing $\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C)$ requires to call e-hd^{π} for $|\mathcal{E}_D| \cdot |\mathcal{E}_C|$ times. Each call of e-hd^{π} will make a recursive call to hd^{π} and its number of calls is bounded by the height of $\mathcal{T}_D$ and $\mathcal{T}_C$. Hence, $\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C)$ are bounded by $\mathcal{O}(|V_C| \cdot |V_D|)$.

We can also show sim^{π} is a procedure which ensures termination and can be used as an indicator for the degree of commonalities under preference profile π between

$\mathcal{ELH}$ concepts. That is, we ensure that the correct results are corrects (*cf.* Lemma 4.1) and the negative results are also correct (*cf.* Lemma 4.2). Termination ensures to provide an answer in finite time.

Lemma 4.1. Let C, D be $\mathcal{ELH}$ concepts and $\pi' = \langle \mathfrak{i}^{\mathrm{c}}, \mathfrak{i}^{\mathrm{r}}, \mathfrak{s}^{\mathrm{c}}, \mathfrak{s}^{\mathrm{r}}, \mathfrak{d} \rangle$ be any preference profile, where $\mathfrak{i}^{\mathrm{c}}(A) \in (0,2]$ for all $A \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$, $\mathfrak{i}^{\mathrm{r}}(r) \in (0,2]$ for all $r \in \mathrm{RN}(\mathcal{T})$, $\mathfrak{d}(r) \in (0,1]$ for all $r \in \mathrm{RN}(\mathcal{T})$. Then, $\mathrm{sim}^{\pi'}(C, D) \in (0,1]$ implies that both C and D share commonalities under π' among each other.

Proof. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$, $\mathcal{T}_C, \mathcal{T}_D$ be their corresponding trees, and π' be any preference profile where $\mathfrak{i}^{\mathrm{c}}(A) \in (0,2]$ for all $A \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$, $\mathfrak{i}^{\mathrm{r}}(r) \in (0,2]$ for all $r \in \mathrm{RN}(\mathcal{T})$, $\mathfrak{d}(r) \in (0,1]$ for all $r \in \mathrm{RN}(\mathcal{T})$. With Lemma 3.1, Theorem 4.1, and the average, it suffices to show that $\mathrm{hd}^{\pi'}(\mathcal{T}_C, \mathcal{T}_D) \in (0,1]$ implies the partial subsumption under π' from D to C based on the characterization of homomorphism structural subsumption $\Leftrightarrow \mathrm{hd}^{\pi'}(\mathcal{T}_C, \mathcal{T}_D) \in (0,1]$ implies $\text{p-hd}^{\pi'}(\mathcal{P}_C, \mathcal{P}_D) > 0$ or $\text{e-set-hd}^{\pi}(\mathcal{E}_C, \mathcal{E}_D) > 0$. We show these cases as follows:

- For any $v \in \mathrm{V}_C$, for any $\mathrm{h}(v) \in \mathrm{V}_D$, if there exists $A \in \mathrm{l}_C(v)$ and $B \in \mathrm{l}_D(\mathrm{h}(v))$ such that $\hat{\mathfrak{s}}(A, B) > 0$, then we show that $\text{p-hd}^{\pi'}(\mathcal{P}_C, \mathcal{P}_D) > 0$. To show this, we fix any $v' \in \mathrm{V}_C$, any $\mathrm{h}(v') \in \mathrm{V}_D$; and assume $\mathrm{A} \in \mathrm{l}_C(v')$, $B \in \mathrm{l}_D(\mathrm{h}(v'))$, and $\hat{\mathfrak{s}}(A, B) > 0$. By Definition 4.11, we know $\text{p-hd}^{\pi'}(\mathcal{P}_C, \mathcal{P}_D) > 0$.
- For any $v, w \in \mathrm{V}_C$, for any $\mathrm{h}(v), \mathrm{h}(w) \in \mathrm{V}_D$, if there exists $r \in \rho_C(v, w)$ and $s \in \rho_D(\mathrm{h}(v), \mathrm{h}(w))$ such that, then we show $\hat{\mathfrak{s}}(r, s) > 0$ that $\text{e-set-hd}^{\pi'}(\mathcal{E}_C, \mathcal{E}_D) > 0$. To show this, we fix any $v', w' \in \mathrm{V}_C$, any $\mathrm{h}(v'), \mathrm{h}(w') \in \mathrm{V}_D$; and assume $r \in \rho_C(v', w')$, $\mathrm{s} \in \rho_D(\mathrm{h}(v'), \mathrm{h}(w'))$, and $\hat{\mathfrak{s}}(r, s) > 0$. By assumptions, we know $\gamma^{\pi'}(r, s) > 0$. Since $\mathrm{hd}^{\pi'}$ cannot be decreased according to Definition 4.11, we conclude that $\text{e-set-hd}^{\pi'}(\mathcal{E}_C, \mathcal{E}_D) > 0$.

Lemma 4.2. Let C, D be any $\mathcal{ELH}$ concepts and $\pi' = \langle \mathfrak{i}^{\mathrm{c}}, \mathfrak{i}^{\mathrm{r}}, \mathfrak{s}^{\mathrm{c}}, \mathfrak{s}^{\mathrm{r}}, \mathfrak{d} \rangle$ be any preference profile where $\mathfrak{i}^{\mathrm{c}}(A) \in (0,2]$ for all $A \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$, $\mathfrak{i}^{\mathrm{r}}(r) \in (0,2]$ for all $r \in$

$\mathrm{RN}(\mathcal{T})$, $\mathfrak{d}(r) \in (0,1]$ for all $r \in \mathrm{RN}(\mathcal{T})$. Then, if both C and D share commonalities under π' among each other, then $\mathrm{sim}^{\pi'}(C, D) \in (0,1]$.

Proof. Let $C, D \in \mathrm{Con}(\mathcal{ELH})$, $\mathcal{T}_C, \mathcal{T}_D$ be their corresponding trees, and π' be any preference profile where $\mathfrak{i}^{\mathrm{c}}(A) \in (0,2]$ for all $A \in \mathrm{CN}^{\mathrm{pri}}(\mathcal{T})$, $\mathfrak{i}^{\mathrm{r}}(r) \in (0,2]$ for all $r \in \mathrm{RN}(\mathcal{T})$, $\mathfrak{d}(r) \in (0,1]$ for all $r \in \mathrm{RN}(\mathcal{T})$. We show its contraposition *i.e.* $\mathrm{sim}^{\pi'}(C, D) = 0$ implies that C and D do not share commonalities under π' to each other.

By the average, we know that $\mathrm{hd}^{\pi'}(\mathcal{T}_C, \mathcal{T}_D) = 0$ and $\mathrm{hd}^{\pi'}(\mathcal{T}_D, \mathcal{T}_C) = 0$. This means that both C and D do not share any commonalities under π' to each other.

Theorem 4.3. The measure sim^{π} is guaranteed for termination and fulfills the condition:

$\mathrm{sim}^{\pi'}(C, D) \in (0,1]$ iff both C and D share commonalities under π' among each other.

Proof. This is obvious by Lemma 4.1, Lemma 4.2, Theorem 4.2.

4.3.2 Desirable Properties of sim^{π}

Previously, we theorize a set of desirable properties that a concept similarity measure under preference profile should satisfy and systematically introduce the measure sim^{π}. In this section, we provide mathematical proofs for the desirable properties of sim^{π}. Understanding the properties gives many benefits to the users of sim^{π} since they can predict its expected behaviors.

Theorem 4.4. sim^{π} is symmetric.

Proof. Let Π be a countably infinite set of preference profile. Fix any $\pi \in \Pi$ and $C, D \in \mathrm{Con}(\mathcal{ELH})$, we have $\mathrm{sim}^{\pi}(C, D) = \mathrm{sim}^{\pi}(D, C)$ by Definition 4.11.

Theorem 4.5. sim^{π} is equivalence invariant.

Proof. Let Π be a countably infinite set of preference profile. Fix any $\pi \in \Pi$ and $C, D, E \in \mathrm{Con}(\mathcal{ELH})$, we show $C \equiv D \Longrightarrow \mathrm{sim}^{\pi}(C, E) = \mathrm{sim}^{\pi}(D, E)$.

Suppose $C \equiv D$, *i.e.* $C \sqsubseteq D$ and $D \sqsubseteq C$, then we know there exists a homomorphism $\mathrm{h}_1: \mathcal{T}_D \to \mathcal{T}_C$ which maps the root of $\mathcal{T}_D$ to the root of $\mathcal{T}_C$ and $\mathrm{h}_2: \mathcal{T}_C \to \mathcal{T}_D$ which maps the root of $\mathcal{T}_C$ to the root of $\mathcal{T}_D$, respectively, by Theorem 3.1. This means $\mathcal{T}_C = \mathcal{T}_D$. Thus, $\mathrm{sim}^{\pi}(C, E) = \mathrm{sim}^{\pi}(D, E)$.

Theorem 4.6. sim^{π} is structurally dependent.

Proof. Let Π be a countably infinite set of preference profile. Fix any $\pi \in \Pi$ and any finite sets of concepts C_1 and C_2 with the following conditions: (1) $\mathrm{C}_1 \subseteq \mathrm{C}_2$; (2) concepts $A, B \notin \mathrm{C}_2$; (3) $\mathfrak{i}^{\mathrm{c}}(\Phi) > 0$ if primitive $\Phi \in \mathrm{C}_2$; (4) $\mathfrak{i}^{\mathrm{r}}(\varphi) > 0$ if existential $\exists\varphi.\Psi \in \mathrm{C}_2$. Suppose $C := \sqcap(\mathrm{C}_1 \cup \{A\})$, $D := \sqcap(\mathrm{C}_1 \cup \{B\})$, $E := \sqcap(\mathrm{C}_2 \cup \{A\})$, and $F := \sqcap(\mathrm{C}_2 \cup \{B\})$ where $\mathrm{C}_1 = \{P_1, \dots, P_m, \exists r_1.P'_1, \dots, \exists r_n.P'_n\}$ and $\mathrm{C}_2 = \{P_1, \dots, P_i, \exists r_1.P'_1, \dots, \exists r_j.P'_j\}$, w.l.o.g. we show $\mathrm{sim}^{\pi}(C, D) \le \mathrm{sim}^{\pi}(E, F)$ by following two cases.

Suppose $m \le i$, $n = j$ and A, B be primitives, we have $\text{p-hd}^{\pi}(\mathcal{P}_C, \mathcal{P}_D) = \frac{\sum_{P \in \mathcal{P}_C} \mathfrak{i}^{\mathrm{c}}(P)}{\sum_{P \in \mathcal{P}_C} \mathfrak{i}^{\mathrm{c}}(P) + \mathfrak{i}^{\mathrm{c}}(A)}$, $\text{p-hd}^{\pi}(\mathcal{P}_D, \mathcal{P}_C) = \frac{\sum_{P \in \mathcal{P}_D} \mathfrak{i}^{\mathrm{c}}(P)}{\sum_{P \in \mathcal{P}_D} \mathfrak{i}^{\mathrm{c}}(P) + \mathfrak{i}^{\mathrm{c}}(B)}$, $\text{p-hd}^{\pi}(\mathcal{P}_E, \mathcal{P}_F) = \frac{\sum_{P \in \mathcal{P}_E} \mathfrak{i}^{\mathrm{c}}(P)}{\sum_{P \in \mathcal{P}_E} \mathfrak{i}^{\mathrm{c}}(P) + \mathfrak{i}^{\mathrm{c}}(A)}$, and $\text{p-hd}^{\pi}(\mathcal{P}_F, \mathcal{P}_E) = \frac{\sum_{P \in \mathcal{P}_F} \mathfrak{i}^{\mathrm{c}}(P)}{\sum_{P \in \mathcal{P}_F} \mathfrak{i}^{\mathrm{c}}(P) + \mathfrak{i}^{\mathrm{c}}(B)}$. Since $m \le i$, we know $\text{p-hd}^{\pi}(\mathcal{P}_C, \mathcal{P}_D) \le \text{p-hd}^{\pi}(\mathcal{P}_E, \mathcal{P}_F)$ and $\text{p-hd}^{\pi}(\mathcal{P}_D, \mathcal{P}_C) \le \text{p-hd}^{\pi}(\mathcal{P}_F, \mathcal{P}_E)$. This infers $\mathrm{sim}^{\pi}(C, D) \le \mathrm{sim}^{\pi}(E, F)$.

Suppose $m = i$, $n \le j$, and A, B be existentials, then with the similar manner, we can show $\text{e-set-hd}^{\pi}(\mathcal{E}_C, \mathcal{E}_D) \le \text{e-set-hd}^{\pi}(\mathcal{E}_E, \mathcal{E}_F)$ and $\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C) \le \text{e-set-hd}^{\pi}(\mathcal{E}_F, \mathcal{E}_E)$. This also infers $\mathrm{sim}^{\pi}(C, D) \le \mathrm{sim}^{\pi}(E, F)$.

Therefore, we have shown $\mathrm{sim}^{\pi}(C, D) \le \mathrm{sim}^{\pi}(E, F)$.

Lemma 4.3. Let $\mathcal{T}_D, \mathcal{T}_C \in \mathbb{T}^{\mathcal{ELH}}$ and Π be a countably infinite set of preference profile. Then, $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1 \Leftrightarrow \forall\pi \in \Pi: \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C) = 1$.

Proof. Let Π be a countably infinite set of preference profile and π_0 be the default preference profile. Fix any $\pi \in \Pi$, we show $\mathrm{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1 \Leftrightarrow \mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C) = 1$.

($\Longrightarrow$) $\text{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$ implies that there exists a homomorphism $\text{h}: \mathcal{T}_D \rightarrow \mathcal{T}_C$ which maps the root of $\mathcal{T}_D$ to the root of $\mathcal{T}_C$. Consequently, any setting on π does not influence the calculation on $\text{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C)$.

($\Longleftarrow$) In particular, it suffices to show $\text{hd}^{\pi_0}(\mathcal{T}_D, \mathcal{T}_C) = 1 \Longrightarrow \text{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$. By Lemma 4.1, it is the case that $\text{hd}(\mathcal{T}_D, \mathcal{T}_C) = 1$.

Theorem 4.7. sim^{π} is preference invariant w.r.t. equivalence.

Proof. Let $\mathsf{C}, \mathsf{D} \in \mathsf{Con}(\mathcal{ELH})$ and Π be a countably infinite set of preference profile. Fix any $\pi \in \Pi$, we show $C \equiv D \Leftrightarrow \text{sim}^{\pi}(C, D) = 1$.

($\Longrightarrow$) Assume $C \equiv D$, we need to show $\text{sim}^{\pi}(C, D) = 1$. By Theorem 3.2, we know $C \equiv D \Leftrightarrow \text{sim}(C, D) = 1$. With the usage of Lemma 4.3, Definition 3.9, and Definition 4.11, we can derive $\text{sim}^{\pi}(C, D) = 1$.

($\Longleftarrow$) This can be shown similarly as in the forward direction.

Theorem 4.4 to 4.7 spells out that sim^{π} satisfies all fundamental properties of concept similarity measure under preference profile.

Definition 4.10 suggests that different preference profile settings represent different types of a rational agent. An easy characterization is observed from the aspect of role discount factor ($\mathfrak{d}$). Intuitively, when the settings $\mathfrak{i}^{\mathfrak{c}}$, $\mathfrak{i}^{\mathfrak{r}}$, $\mathfrak{s}^{\mathfrak{c}}$, and $\mathfrak{s}^{\mathfrak{r}}$ defined by two rational agents A, B are the same, the agent which defines the lower $\mathfrak{d}$ on every $r \in \text{RN}$ is always more skeptical. For instance, if $\mathfrak{d}_{\text{A}}(\text{canWalk}) = 0.3$ and $\mathfrak{d}_{\text{B}}(\text{canWalk}) = 0.4$, then $\text{sim}^{\pi_{\text{A}}}(\exists\text{canWalk.Trekking}, \exists\text{canWalk.Parading}) = 0.3$ and $\text{sim}^{\pi_{\text{B}}}(\exists\text{canWalk.Trekking}, \exists\text{canWalk.Parading}) = 0.4$. This is clear that the agent A is more skeptical than the agent B.

Proposition 4.2. Let Π be a countably infinite set of preference profile and $\pi_1, \pi_2 \in \Pi$ such that $\pi_1 = \langle \mathfrak{i}_1^{\mathfrak{c}}, \mathfrak{i}_1^{\mathfrak{r}}, \mathfrak{s}_1^{\mathfrak{c}}, \mathfrak{s}_1^{\mathfrak{r}}, \mathfrak{d}_1 \rangle$, $\pi_2 = \langle \mathfrak{i}_2^{\mathfrak{c}}, \mathfrak{i}_2^{\mathfrak{r}}, \mathfrak{s}_2^{\mathfrak{c}}, \mathfrak{s}_2^{\mathfrak{r}}, \mathfrak{d}_2 \rangle$, and RN be a set of role names. The following holds[16]:

$$\forall r \in \text{RN}: \big(\mathfrak{d}_1(r) \leq \mathfrak{d}_2(r)\big) \Longrightarrow \equiv \preceq \ \text{sim}^{\pi_1} \preceq \ \text{sim}^{\pi_2}$$

---

[16] See Definition 4.10 for the meaning of $\preceq$.

for fixed functions $\mathfrak{i}_1^{\mathfrak{c}} = \mathfrak{i}_2^{\mathfrak{c}}$, $\mathfrak{i}_1^{\mathfrak{r}} = \mathfrak{i}_2^{\mathfrak{r}}$, $\mathfrak{s}_1^{\mathfrak{c}} = \mathfrak{s}_2^{\mathfrak{c}}$, and $\mathfrak{s}_1^{\mathfrak{r}} = \mathfrak{s}_2^{\mathfrak{r}}$.

4.4 Implementation Methods of sim^{π}

Theorem 4.2 tells us that sim^{π} can be computed in the polynomial time. This section exhibits two algorithmic procedures of sim^{π} belonging to that class.

4.4.1 Top-Down Implementation of sim^{π}

Algorithm 1 Pseudo code for hd^{π} using top-down fashion (Part 1)

```
function hd^π(T_D, T_C, π)
    return (μ^π(T_D, π) × p-hd^π(P_D, P_C, π)) + ((1 − μ^π(T_D, π)) × e-set-hd^π(E_D, E_C, π))
end function

function μ^π(T_D, π)
    if P_D.isEmpty() and E_D.isEmpty() then
        return 1
    end if
    return Σ i^c(P_D, π) / (Σ i^c(P_D, π) + Σ i^r(E_D, π))
end function

function p-hd^π(P_D, P_C, π)
    if Σ i^c(P_D, π) = 0 then
        return 1
    else if Σ i^c(P_C, π) = 0 then
        return 0
    else
        w ← 0
        for A ∈ P_D do
            m ← 0
            for B ∈ P_C do
                v ← ŝ(A, B)
                if v > m then
                    m ← v
                end if
            end for
            w ← w + (m × î(A))
        end for
        return w / Σ i^c(P_D, π)
    end if
end function
```

Figure 4.1 Pseudo Code for hd^{π} using Top-Down Fashion (Part 1).

In Definition 4.7, hd^{π} is established by an inductive procedure. Therefore, it is a very straightforward way to implement the procedure by recursion (see Figure 4.1).

Algorithm 1 Pseudo code for hd^{π} using top-down fashion (Part 2)

```
function e-set-hd^π(E_D, E_C, π)
    if ∑ i^r(E_D, π) = 0 then
        return 1
    else if ∑ i^r(E_C, π) = 0 then
        return 0
    else
        w ← 0
        for ∃r.X ∈ E_D do
            m ← 0
            for ∃s.Y ∈ E_C do
                e ← e-hd^π(∃r.X, ∃s.Y, π)
                if e > m then
                    m ← e
                end if
            end for
            w ← w + (m × î(r))
        end for
        return w/∑ i^r(P_D, π)
    end if
end function
```

Figure 4.2 Pseudo Code for hd^{π} using Top-Down Fashion (Part 2).

From Figure 4.1 and Figure 4.2, hd^{π} is directly followed from Equation 4.4 of Definition 4.7. That is, it receives three parameters as inputs, *viz.* a description tree $\mathcal{T}_D$, a description tree $\mathcal{T}_C$, and a preference profile π. Suppose $\mathcal{T}_D$ be defined as $\mathcal{P}_D \cup \mathcal{E}_D$, $\mathcal{T}_C$ be defined as $\mathcal{P}_C \cup \mathcal{E}_C$, and $\pi = \langle \mathfrak{i}^{\mathfrak{c}}, \mathfrak{i}^{\mathfrak{r}}, \mathfrak{s}^{\mathfrak{c}}, \mathfrak{s}^{\mathfrak{r}}, \mathfrak{d} \rangle$ is given. The function $\mathrm{hd}^{\pi}(\mathcal{T}_D, \mathcal{T}_C, \pi)$ computes the function value for a composition in a prescribed way from the function values of the composing parts, *i.e.* $\mu^{\pi}(\mathcal{T}_D, \pi)$, $\text{p-hd}^{\pi}(\mathcal{P}_D, \mathcal{P}_C, \pi)$, and $\text{e-set-hd}^{\pi}(\mathcal{E}_D, \mathcal{E}_C, \pi)$. μ^{π}, p-hd^{π}, e-set-hd^{π} are also followed from Equation 4.5, 4.6, and 4.7, respectively, of Definition 4.7. Each internally uses subfunctions $\sum \mathfrak{i}^{\mathfrak{c}}$ and $\sum \mathfrak{i}^{\mathfrak{r}}$ (see Figure 4.4) to calculate the total number of concept importance and the total number of role importance, respectively.

For Figure 4.3 and Figure 4.4, e-hd^{π} is directly followed from Equation 4.8 of Definition 4.7. To compute the function value e-hd^{π}, we recursively compute the function value hd^{π} on the children of certain nodes (denoted by X and Y) and π. γ^{π} is directly followed from Equation 4.9 of Definition 4.7. Also, γ^{π} internally invokes subfunction $\sum \mathfrak{i}^{\mathfrak{r}}$ to calculate the total number of role importance.

Algorithm 1 Pseudo code for hd^{π} using top-down fashion (Part 3)

```
function e-hd^π(∃r.X, ∃s.Y, π)
    return γ^π(r, s, π) × (𝔡̂(r) + ((1 − 𝔡̂(r)) × hd^π(𝒯_X, 𝒯_Y, π)))
end function

function γ^π(r, s, π)
    if ∑i^r(ℛ_r, π) then
        return 1
    else
        w ← 0
        for ζ ∈ ℛ_r do
            m ← 0
            for ϱ ∈ ℛ_s do
                v ← 𝔰̂(ζ, ϱ)
                if v > m then
                    m ← v
                end if
            end for
            w ← w + (m × 𝔦̂(ζ))
        end for
        return w/∑i^r(ℛ_r, π)
    end if
end function
```

Figure 4.3 Pseudo Code for hd^{π} using Top-Down Fashion (Part 3).

The reader may easily observe that the time efficiency of Algorithm 1 is quintic because the computation of p-hd^{π} is quadratic and e-set-hd^{π} contains double nested loops which indirectly make recursive calls to hd^{π}. It is also not difficult to observe that the number of recursive calls is upper bounded by the height of the description tree.

It is worth to mention that using hd^{π} requires concept descriptions to be transformed into $\mathcal{ELH}$ description trees. Taking this as an advantage, the next subsection introduces an alternative way to compute hd^{π} from bottom to up, which is approximately three times faster than the counterpart top-down approach in the worst case (*cf.* Subsection 4.5.1 for useful discussion).

Algorithm 1 Pseudo code for hd^π using top-down fashion (Part 4)

```
function ∑ i^c(𝒫_D, π)
    w ← 0
    for A ∈ 𝒫_D do
        w ← w + î(A)
    end for
    return w
end function

function ∑ i^r(ℰ_D, π)
    w ← 0
    for ∃r.X ∈ ℰ_D do
        w ← w + î(r)
    end for
    return w
end function

function ∑ i^r(ℛ_r, π)
    w ← 0
    for r ∈ ℛ_r do
        w ← w + î(r)
    end for
    return w
end function
```

Figure 4.4 Pseudo Code for hd^π using Top-Down Fashion (Part 4).

4.4.2 Bottom-Up Implementation of sim^π

Rather than computing (possibly duplicated) value of hd^π again and again, Figure 4.5 shows the classical bottom-up version of dynamic programming technique to compute hd^π of the smaller subtrees and records the results in a table (see the variable $\mathrm{result}[\cdot][\cdot]$ in the figure) from which a solution to the original computation of hd^π can be then obtained (*cf.* at line no. 20, the function returns value $result[0][0]$).

To compute hd^π from bottom to up, we need to know the height of the trees in advance. According to Figure 4.5, we employ 'breath-first search' algorithm (denoted by BFS) to determine the height of each description tree (*cf.* line no. 4 and 5 of the algorithm). The algorithm reuses the methods μ^π, p-hd^π, e-set-hd^π, γ^π, $\sum \mathrm{i}^\mathrm{c}$, and $\sum \mathrm{i}^\mathrm{r}$ from the top-down algorithm and provides pseudo code for e-hd^π since it is merely overridden.

Algorithm 2 Pseudo code for hd^π using bottom-up fashion

1: Initialize a global $result[\cdot][\cdot]$ to store the degree of similarity between 2 concepts.
2:
3: **function** $\mathsf{hd}^\pi(\mathcal{T}_D, \mathcal{T}_C, \pi)$
4: $\quad$ $\text{Map} < \mathbb{Z}, \text{List} < \mathcal{T} >> map_D \leftarrow \mathsf{BFS}(\mathcal{T}_D)$ $\quad \rhd$ map_D stores nodes on each level of $\mathcal{T}_D$
5: $\quad$ $\text{Map} < \mathbb{Z}, \text{List} < \mathcal{T} >> map_C \leftarrow \mathsf{BFS}(\mathcal{T}_C)$ $\rhd$ map_C stores nodes on each level of $\mathcal{T}_C$
6: $\quad$ $h \leftarrow map_D.\text{size}()$
7: $\quad$ **for** $i = h - 1$ **to** 0 **do**
8: $\qquad$ $\text{List} < \mathcal{T} > list_{\mathcal{T}_\Gamma} \leftarrow map_D.\text{get(i)}$
9: $\qquad$ $\text{List} < \mathcal{T} > list_{\mathcal{T}_\Lambda} \leftarrow map_C.\text{get(i)}$
10: $\qquad$ **for** $\mathcal{T}_\gamma \in list_{\mathcal{T}_\Gamma}$ **do**
11: $\qquad\quad$ **for** $list_{\mathcal{T}_\Lambda} \neq \text{null}$ and $\mathcal{T}_\lambda \in list_{\mathcal{T}_\Lambda}$ **do**
12: $\qquad\qquad$ **if** $i = h - 1$ **then**
13: $\qquad\qquad\quad$ $result[\mathcal{T}_\gamma][\mathcal{T}_\lambda] \leftarrow \mathsf{p\text{-}hd}^\pi(\mathcal{P}_\gamma, \mathcal{P}_\lambda, \pi)$
14: $\qquad\qquad$ **else**
15: $\qquad\qquad\quad$ $result[\mathcal{T}_\gamma][\mathcal{T}_\lambda] \leftarrow (\mu^\pi(\mathcal{T}_\gamma, \pi) \times \mathsf{p\text{-}hd}^\pi(\mathcal{P}_\gamma, \mathcal{P}_\lambda, \pi))$ $+ ((1 - \mu^\pi(\mathcal{T}_\gamma, \pi)) \times \mathsf{e\text{-}set\text{-}hd}^\pi(\mathcal{E}_\gamma, \mathcal{E}_\lambda, \pi))$
16: $\qquad\qquad$ **end if**
17: $\qquad\quad$ **end for**
18: $\qquad$ **end for**
19: $\quad$ **end for**
20: $\quad$ **return** $result[0][0]$
21: **end function**
22:
23: **function** $\mathsf{e\text{-}hd}^\pi(\exists r.X, \exists s.Y, \pi)$
24: $\quad$ $hd' \leftarrow result[\mathcal{T}_X][\mathcal{T}_Y]$
25: $\quad$ **if** $hd' = \text{null}$ **then**
26: $\qquad$ $hd' \leftarrow 0$
27: $\quad$ **end if**
28: $\quad$ **return** $\gamma^\pi(r, s, \pi) \times (\hat{\mathfrak{d}}(r) + ((1 - \hat{\mathfrak{d}}(r)) \times hd'))$
29: **end function**

Figure 4.5 Pseudo Code for hd^π using Bottom-Up Fashion.

What is the time complexity of this approach? It should be quintic because the algorithm considers the similarity of all the different pairs of two concept names for h times (*cf.* line no. 6). More formally, we know $result[\mathcal{T}_\gamma][\mathcal{T}_\lambda] \in \mathcal{O}(v^2)$ where v denotes the set cardinality of $\mathcal{P}_x$ (and $\mathcal{E}_x$) for any description tree x. Let $m(i)$ and $n(i)$ be the number of nodes on level i of description trees D and C, respectively. Then, the number of times operation $result[\cdot][\cdot]$ is executed (say C) is equal to:

$$
\begin{aligned}
C &= \textstyle\sum_{i=0}^{h-1} \sum_{j=0}^{m(i)} \sum_{k=0}^{n(i)} v^2 \\
&= v^2 \textstyle\sum_{i=0}^{h-1} \sum_{j=0}^{m(i)} \sum_{k=0}^{n(i)} 1 \\
&= v^2 \textstyle\sum_{i=0}^{h-1} \sum_{j=0}^{m(i)} (n(i) + 1) \\
&= v^2 \textstyle\sum_{i=0}^{h-1} (n(i) + 1)(m(i) + 1) \\
&= v^2 \big[[(n(0) + 1)(m(0) + 1)] + [(n(1) + 1)(m(1) + 1)] + \cdots \\
&\qquad + [(n(h-1) + 1)(m(h-1) + 1)] \big]
\end{aligned}
$$

Thus, the algorithm makes the similar number of operations as the top-down approach, plus an additional amount of extra space. On the positive side, the algorithm has never recursively invoked itself to determine the similarity of different pairs of nested concepts, *i.e.* it directly uses values stored in the table. The algorithm also shows that computing the similarity of nodes from level i, where i is greater than the minimum height of description trees (*cf.* the condition $\text{list}_{\mathcal{T}_\Lambda}$!= null at line no. 11), is irrelevant to the computation.

The bottom-up approach does work productively in an environment where recursion is fairly expensive. For example, imperative languages, such as Java, C, and Python, are typically faster if using a loop and slower if doing a recursion. On the other hand, for some implementations of functional programming languages, iterations may be very expensive and recursion may be very cheap. In many implementations of them, recursion is transformed into a simple jump but changing the loop variables (which are mutable) requires heavy operations. Subsection 4.5.1 reports that the practical performance agrees to this theoretical analysis that the bottom-up approach is more efficient when implemented by imperative languages, such as Java.

4.5 Empirical Evaluation

This section evaluates the practical performance of both algorithms against sim[17], reassures pragmatically the backward compatibility of sim^π under π_0 (Theorem 4.1 already proves this), and discusses the applicability of sim^π in potential use cases.

[17] We have re-implemented sim (proposed in (Tongphu & Suntisrivaraporn, 2015)) based on the same technologies and techniques as sim^π.

4.5.1 Performance Analysis and Backward Compatibility of sim^{π}

Both versions of sim^{π} (*cf.* Subsection 4.4.1 and Subsection 4.4.2) are implemented in Java version 1.8 with the usage of Spring Boot version 1.3.3. RELEASE. All the dependencies are managed by Apache Maven version 3.2.5. We also implement unit test cases along with the development of both versions to verify the correctness of their behaviors. In the current state (when we are writing this work), there are 111 unit test cases. All of them are written to cover important parts of both implementations.

To perform benchmarking, we have selected SNOMED CT as a test ontology. As mentioned in Appendix A, it is one of the largest and the most widely used medical ontologies currently available, and also, is expressible in $\mathcal{ELH}$. In our experiments, we employ a SNOMED CT ontology version from January 2005 (hitherto referred as $\mathcal{O}_{\text{SNOMED CT}}$) which contains 379,691 concept names and 62 role names. Moreover, each defined concept is categorized into the 18 mutually exclusive top-level concepts. In the sense of subsumption relation, concepts belonging to the same category should be more similar than those belonging to different categories.

For our experiments, we used a 2.4 GHz Intel Core i5 with 8 GB RAM under OS X El Capitan. Unfortunately, the overall number of concept pairs in $\mathcal{O}_{\text{SNOMED CT}}$ is approximately 10^{11}. Suppose an execution of simπ takes around a millisecond, we still need around 1,158 days in order to complete the entire ontology. According to this reason, we consider 2 out of 18 categories, *viz. Clinical Finding* and *Procedure*, although there are more category pairs. Then, we randomly select 0.5% of *Clinical Finding*, *i.e.* 206 concepts, denoted by $\mathbb{C}_1'$. After that, we randomly select the same number of concepts from *Procedure*, *i.e.* 206 concepts, denoted by $\mathbb{C}_2'$. This sampled set is denoted by $\mathcal{O'}_{\text{SNOMED CT}}$ *i.e.* $\mathcal{O'}_{\text{SNOMED CT}} = \mathbb{C}_1' \cup \mathbb{C}_2'$. Then, we create three test datasets from this sampled set, *viz.* $\mathbb{C}_1' \times \mathbb{C}_1'$, $\mathbb{C}_1' \times \mathbb{C}_2'$, and $\mathbb{C}_2' \times \mathbb{C}_2'$.

Firstly, we estimate the practical performance of the top-down fashion. For each concept pair in each set, we 1) employ the default preference profile π_0 on (top-down) sim^{π}; 2) measure the similarity of concepts in $\mathcal{O'}_{\text{SNOMED CT}}$ by peeking on $\mathcal{O}_{\text{SNOMED CT}}$ to help unfolding; 3) repeat the previous step with (top-down) sim; 4) repeat steps 2)-3) three times and calculate the statistical results (in milliseconds). Results are gathered on Table 4.1. We note that avg, max, and min represent the execution time for measuring similarity of a concept pair in the average case, in the worst case, and in the best case, respectively.

Table 4.1 Execution Time of Top-Down sim and Top-Down sim^{π_0} on $\mathcal{O}'_{\text{SNOMED CT}}$.

Pairs	Number of Pairs	sim (avg/max/min)	sim^{π_0} (avg/max/min)
$\mathbb{C}_1' \times \mathbb{C}_1'$	25	2.280/7.000/0.000	1.800/10.000/0.000
$\mathbb{C}_1' \times \mathbb{C}_2'$	215	2.291/97.000/0.000	2.278/84.000/0.000
$\mathbb{C}_2' \times \mathbb{C}_2'$	1,849	3.395/45.000/0.000	3.931/128.000/0.000

Secondly, we estimate the practical performance of the bottom-up fashion by following the same steps as we did previously. Indeed, we exclude the time used to determine the height of each description tree, *i.e.* our benchmark begins from line no. 7 to 21 of the bottom-up algorithm. Table 4.2 gathers up the results.

Table 4.2 Execution Time of Bottom-Up sim and Bottom-Up sim^{π_0} on $\mathcal{O}'_{\text{SNOMED CT}}$.

Pairs	Number of Pairs	sim (avg/max/min)	sim^{π_0} (avg/max/min)
$\mathbb{C}_1' \times \mathbb{C}_1'$	25	2.200/6.000/0.000	1.693/5.000/0.000
$\mathbb{C}_1' \times \mathbb{C}_2'$	215	2.040/32.000/0.000	1.946/10.000/0.000
$\mathbb{C}_2' \times \mathbb{C}_2'$	1,849	3.368/55.000/0.000	3.435/45.000/0.000

The experiment shows that the practical performance of sim^{π} is likely equal to the performance obtained by sim – as ones may not expect. The results show that the bottom-up sim^{π} performs approximately three times faster that the counterpart top-down sim^{π} (in the worst case) when implemented by imperative languages (*e.g.* Java as in our case). This conforms to our analysis discussed in Subsection 4.4.2.

Lastly, we evaluate the backward compatibility of sim^{π} with sim. Our goal is to ascertain that sim^{π} can be used interchangeably as the original sim by setting preference profile to the default one (Theorem 4.1 already proves this). To this point, we have performed an experiment on concept pairs defined in $\mathcal{O}'_{\text{SNOMED CT}}$. The experiment evaluates results from sim and sim^{π} and found that both coincide, as desired. Table 4.3 gathers the results, where "td" and "bu" are abbreviation forms of top-down and bottom-up, respectively.

Table 4.3 Results of Executing sim and sim^{π_0} on $\mathcal{O}'_{\text{SNOMED CT}}$.

Pairs	Number of Pairs	td sim (avg/max/min)	td sim^{π_0} (avg/max/min)	bu sim (avg/max/min)	bu sim^{π_0} (avg/max/min)
$\mathbb{C}'_1 \times \mathbb{C}'_1$	25	0.87597/ 1.00000/ 0.67953	0.87597/ 1.00000/ 0.67953	0.87597/ 1.00000/ 0.67953	0.87597/ 1.00000/ 0.67953
$\mathbb{C}'_1 \times \mathbb{C}'_2$	215	0.57801/ 0.66546/ 0.24594	0.57801/ 0.66546/ 0.24594	0.57801/ 0.66546/ 0.24594	0.57801/ 0.66546/ 0.24594
$\mathbb{C}'_2 \times \mathbb{C}'_2$	1,849	0.79690/ 1.00000/ 0.35360	0.79690/ 1.00000/ 0.35360	0.79690/ 1.00000/ 0.35360	0.79690/ 1.00000/ 0.35360

4.5.2 Effects of Tuning sim^{π}

4.5.2.1 Tuning via $\mathfrak{i}^c$ and $\mathfrak{d}$

We show the applicability of $\mathfrak{i}^c$ and $\mathfrak{d}$ through similarity measuring on SNOMED CT. Figure 4.6 depicts an example unfoldable terminology extracted from $\mathcal{O}_{\text{SNOMED CT}}$.

$$
\begin{aligned}
\mathsf{NeonatalAspirationOfAmnioticFluid} &\equiv \mathsf{NeonatalAspirationSyndromes} \\
&\quad \sqcap \exists \mathsf{roleGroup}.(\exists \mathsf{causativeAgent}.\mathsf{AmnioticFluid}) \\
\mathsf{NeonatalAspirationOfMucus} &\equiv \mathsf{NeonatalAspirationSyndromes} \\
&\quad \sqcap \exists \mathsf{roleGroup}.(\exists \mathsf{causativeAgent}.\mathsf{Mucus}) \\
\mathsf{Hypoxemia} &\equiv \mathsf{DisorderOfRespiratorySystem} \sqcap \mathsf{DisorderOfBloodGas} \\
&\quad \sqcap \exists \mathsf{roleGroup}.(\exists \mathsf{interprets}.\mathsf{OxygenDelivery}) \\
&\quad \sqcap \exists \mathsf{roleGroup}.(\exists \mathsf{findingSite}.\mathsf{ArterialSystemStructure}) \\
\mathsf{BodySecretion} &\sqsubseteq \mathsf{BodySubstance} \\
\mathsf{BodySubstance} &\sqsubseteq \mathsf{Substance} \\
\mathsf{BodyFluid} &\sqsubseteq \mathsf{BodySubstance} \sqcap \mathsf{LiquidSubstance} \\
\mathsf{AmnioticFluid} &\sqsubseteq \mathsf{BodyFluid} \\
\mathsf{Mucus} &\sqsubseteq \mathsf{BodySecretion} \\
\mathsf{causativeAgent} &\sqsubseteq \mathsf{associatedWith}
\end{aligned}
$$

Figure 4.6 Example of $\mathcal{ELH}$ Concept Definitions Defined in $\mathcal{O}_{\text{SNOMED CT}}$. .

Considering merely objective factors regardless of the agent's preferences, it yields that $\mathrm{sim}^{\pi_0}(\text{NAOAF,NAOM}) \approx 0.9$[18] and $\mathrm{sim}^{\pi_0}(\text{NAOAF,H}) \approx 0.2$. The results yielding to the quite similar concepts NAOAF and NAOM, which reflects the fact that both are resided in the same cluster of SNOMED CT. However, the result yielding that

[18] Obvious abbreviations are used here for the sake of succinctness.

the concepts NAOAF and H shares a little similarity controverts the fact that both carry neither implicit nor explicit relationship. This is indeed caused by the usage of the special-purpose role called roleGroup – informally read as *relation group*.

In SNOMED CT, the use of relation group is widely accepted to nestedly represent a group of existential information (Schulz, Suntisrivaraporn, & Baader, 2007). As a consequence, it increases unintentionally the degree of similarity due to role commonality (*i.e.* γ^{π}). Since roleGroup precedes every existential restriction, it is useless to regard an occurrence of this as being similar. The importance contribution of roleGroup in $\mathcal{O}_{\text{SNOMED CT}}$ should be none. Hence, the agent S who measures similarity on SNOMED CT should set $\mathfrak{d}_S(\text{roleGroup}) = 0$.

Furthermore, the SNOMED CT top concept SCT-TOP subsumes every defined concept of each category. This means this special concept is shared by every expanded concept description. Intuitively, this special top concept is of no importance for measuring similarity on SNOMED CT and we can treat the top-level concepts as directly subsumed by $\top$. As a result, the agent S should also set $\mathfrak{i}^{c}_{S}(\text{SCT-TOP})$=0.

Tuning the measure with this expertise knowledge yields more realistic result. That is, the similarity of concepts under the same category which uses roleGroup in their definitions is slightly reduced. Also, the similarity of concepts under different categories is totally dissimilar. Continuing the case, $\text{sim}^{\pi_S}(\text{NAOAF,NAOM}) \approx 0.84$ and $\text{sim}^{\pi_S}(\text{NAOAF,H}) = 0.0$, as desired.

4.5.2.2 Tuning via $\mathfrak{s}^{r}$

Let us use the ontology given below to query for places similar to ActivePlace.

ActivePlace $\sqsubseteq$ Place $\sqcap$ $\exists$canSail.Kayaking
Mangrove $\sqsubseteq$ Place $\sqcap$ $\exists$canWalk.Trekking
Supermarket $\sqsubseteq$ Place $\sqcap$ $\exists$canBuy.FreshFood

Suppose the agent feels ‘walking’ and ‘sailing’ are similar and are ‘still satisfied much’ on both actions. Taking $\mathfrak{s}^{r}(\text{canWalk,canSail}) = 0.6$ yields $\text{sim}^{\pi}(\text{M,AP}) > \text{sim}^{\pi}(\text{S,AP})$, which conforms to the agent’s preferences and needs.

4.5.2.3 Tuning via $\mathfrak{s}^c$

Let us use the ontology given below to query for a product which offers features the agent is satisfied with most.

$\mathrm{WantedFeatures} \sqsubseteq \mathrm{F}_0 \sqcap \mathrm{F}_1 \sqcap \mathrm{F}_2$
$\mathrm{Item}_1 \sqsubseteq \mathrm{F}_0 \sqcap \mathrm{F}_3$
$\mathrm{Item}_2 \sqsubseteq \mathrm{F}_0 \sqcap \mathrm{F}_4$

According to the ontology, $\mathrm{WantedFeatures}$ represents a collection of desired features and F_i (where i ∈ N) represents a feature. A purchase decision is sometimes affected by satisfied alternations, which are varied by different people. Assume that the agent feels satisfaction to have F_3 if the agent cannot have F_1. Taking $\mathfrak{s}^c(\mathrm{F}_1, \mathrm{F}_3) = 0.8$ yields $\mathrm{sim}^\pi(\mathrm{WF,I1}) > \mathrm{sim}^\pi(\mathrm{WF,I2})$, which conforms to the agent's perceptions.

4.5.2.4 Tuning via $\mathfrak{i}^r$

Let us use the ontology given in Example 4.1 to query for places which are most similar to ActivePlace. Typically, a human decision is affected by a priority of concerns, which are varied by different people. Suppose that the agent weights more on places which permit to 'walk' more than other activities. Taking $\mathfrak{i}^r(\mathrm{canWalk}) = 2$ yields $\mathrm{sim}^\pi(\mathrm{M,AP}) > \mathrm{sim}^\pi(\mathrm{B,AP})$, which conforms to the agent's preferences.

4.6 Comparison with Related Works

As we develop the notion $\sim^\pi_\tau$ as a generalization of $\sim_\tau$, this section relates our development to others in two areas, *viz.* ordinary concept similarity measures (which do not take into account the agent's preferences) and preference-based concept similarity measures.

4.6.1 Ordinary Concept Similarity Measure

In the standard perception, concept similarity measure refers to the study of similar concepts inherited by nature, *i.e.* the ones similar regardless of the agent's preferences. Our concrete developments, which employ structural subsumption, can be considered as the semantic similarity approach. Hence, we merely compare our approaches to other approaches of semantic similarity as follows.

A simple method was developed in (Jaccard, 1901) for the DL $\mathcal{L}_0$ (*i.e.* no use of roles) and was known as *Jaccard Index*. Its extension to the DL $\mathcal{ELH}$ was proposed in (Lehmann & Turhan, 2012). This work also introduced important properties of concept similarity measure and suggested a general framework called $simi$ which satisfied most of the properties. In $simi$, functions and operators, such as t-conorm and the fuzzy connector, were to be parameterized and thus left to be specified. The framework also did not contain implementation details. This may cause implementation difficulties since merely promising properties were given and no guideline of how concrete operators are chosen is provided.

In (Janowicz & Wilkes, 2009), the measure SIM-DL_A was proposed for the DL $\mathcal{SHI}$. The measure was not completely defined in mathematical terms and some text descriptions were not precise. Roughly, the measure had three stages. First, two concept descriptions in question were converted into the negation normal form (NNF). A modified version of the tableau was used to generate a completion tree for each NNF concept. In this modified version, the $\sqcup$-rule was modified and another $\forall$-rule was added. Second, a set of *proxy models* was generated from the completion tree. A proxy model was a tree where each was labeled by a role name and each node was labeled by a concept name. Third, both sets of proxy models were used to compute the degree of similarity. This was done by measuring similarity among all pairs of proxy models (using tree similarity). The final result was evaluated from either the maximum, the minimum, or the average. However, the paper did not explain the selection rule when more than one tableau rules could be applied.

Two similarity measures for DL $\mathcal{FL}_0$ was proposed in our proceeding papers (Racharak & Suntisrivaraporn, 2015; Racharak & Tojo, 2018). In (Racharak & Suntisrivaraporn, 2015), similarity measures were conformed to the different skeptical aspect of their computation and were derived from the structural subsumption of the language inclusion. The skeptical concept similarity was further generalized and was extended toward the notion of preference profile in (Racharak & Tojo, 2018). It was also shown in (Racharak & Tojo, 2018) that, like in this thesis, when tuning with the default preference profile, it measures similarity of concepts w.r.t. their structure merely.

The notion of homomorphism degree was originally introduced in (Suntisrivaraporn, 2013) and is thereof extended toward the development of sim^{π} for the DL $\mathcal{ELH}$ in this chapter. Theorem 4.1 suggests that simπ can be used to measure similarity of concepts inherently by nature through the setting π_0, *i.e.* sim^{π_0}.

As inspired by the tree homomorphism, sim^{π} differs (Lehmann & Turhan, 2012) from the use of μ^{π} to determine how important the primitive concepts are to be considered and the use of γ^{π} to determine a degree of role commonality between matching edges of the description trees. We further discuss about preference-based similarity measures in the next subsection.

4.6.2 Preference-based Concept Similarity Measure

Most concept similarity measures are objective-based. However, there exists work (Lehmann & Turhan, 2012; Racharak & Tojo, 2018; Tongphu & Suntisrivaraporn, 2015) which provides methodologies for tuning. We discuss their differences to our approaches in the following.

In (Racharak & Tojo, 2018), a concept similarity measure $\sim_s^{\pi}$ under preference profile was proposed for $\mathcal{FL}_0$ concept descriptions. Both were developed from a characterization of language inclusion in description logics. Unfortunately, $\sim_s^{\pi}$ merely supports some preferential elements in preference profile.

In an extended work of sim (Tongphu & Suntisrivaraporn, 2015), a range of number for discount factor (ν) and the neglect of special concept names were used in the similarity application of SNOMED CT. For instance, when roleGroup was found, the value of ν was set to 0. These ad hoc approaches can be viewed as specific applications of $\mathfrak{d}$ and $\mathfrak{i}^c$, respectively, of preference profile. Unfortunately, no other aspects of π appear in its use.

In $simi$ (Lehmann & Turhan, 2012), the function pm was used to define the similarity degree of primitive concept pairs and role pairs. Using pm with primitive concept pairs invokes the equivalent intuition as $\mathfrak{s}^c$; however, this does not mean so in the aspect $\mathfrak{s}^r$. Allowing to define the similarity of defined role names, as in (Lehmann

& Turhan, 2012), may be not appropriate since defined role names are contributed by primitive role names. For example, let $r_1 \sqsubseteq s_1$ and $r_2 \sqsubseteq s_2$ are defined in $\mathcal{T}$. It is clear that $r_1, r_2 \in \mathrm{RN}^{\mathrm{def}}$. By defining $pm(r_1, r_2)$, the defined similarity should be also propagated to the similarity of s_1 and s_2. However, this point was not discussed in (Lehmann & Turhan, 2012). In respect of this, $\mathrm{RN}^{\mathrm{pri}}$ is merely used in $\mathfrak{s}^{\mathrm{r}}$ and γ^{π} is defined for the similarity of defined role names. The authors of (Lehmann & Turhan, 2012) also defined the function $\mathrm{g}\colon \mathrm{N_A} \to \mathrm{R}_{>\mathbb{0}}$ representing the weight for concept names and existential restriction atoms (based on their definition). Ones may feel the resemblance of g and $\mathfrak{i}^{\mathrm{c}}, \mathfrak{i}^{\mathrm{r}}$; however, they are also different in three perspectives. Firstly, the mapping of g is reached to the infinity whereas $\mathfrak{i}^{\mathrm{c}}$ and $\mathfrak{i}^{\mathrm{r}}$ are bounded. This characteristic of g is impractical to use as it may lead to the unbalance of weight assignments. For instance, one may define $\mathrm{g}(A_1) = 1$ but $\mathrm{g}(A_2) = 10^{12}$ where $A_1, A_2 \in \mathrm{CN}^{\mathrm{pri}}$. To avoid this situation, the authors should provide a guideline for weight assignments. Secondly, the mapping of g is lower bounded by one. This clearly makes an impossibility to define the intuition of having no importance. Thus, the situation given in Subsubsection 4.5.2 is not expressible. Lastly, the domain of g is the set of atoms whereas $\mathfrak{i}^{\mathrm{c}}$ (and $\mathfrak{i}^{\mathrm{r}}$) is the set of primitive concept names (and the set of role names, respectively). Using the set of atoms as the domain is also impractical since there can be infinitely many existential restriction atoms and the interpretation of functions is slightly dubious. For instance, given $\mathrm{g}(\exists r.C) = 2$ and $\mathrm{g}(\exists r.D) = 3$, do both r intentionally contribute the equal importance? Thus, this definition is inappropriate to represent the agent's perception. Moreover, the aspect $\mathfrak{d}$ disappeared from (Lehmann & Turhan, 2012). Lacking of fully $\mathfrak{i}^{\mathrm{c}}$ and $\mathfrak{d}$ makes the framework inappropriate to use for Snomed ct applications. These distinctions of simi and ours are radically caused by their different motivations. Table 4.4 summarizes this discussion, where • denotes totally identical to the specified function whereas ° denotes partially identical to the specified function.

Not only distinct on the mathematical representation of $simi$ and our measures, the desired properties presented in each work are also different. While the properties introduced in (Lehmann & Turhan, 2012) were motivated for (ordinary) concept similarity measure, our properties are developed under the consideration of the agent's preferences ($\sim_{\tau}^{\pi}$). Hence, some properties introduced for concept similarity measure are revised in subjective manners and the new property is introduced.

Table 4.4 Concept Similarity Measures which Embed Preference Elements.

Similarity Measure	DL	$\mathfrak{i}^c$	$\mathfrak{i}^r$	$\mathfrak{s}^c$	$\mathfrak{s}^r$	$\mathfrak{d}$
sim^{π}	$\mathcal{ELH}$	●	●	●	●	●
$\sim_{s}^{\pi}$ (Racharak & Tojo, 2018)	$\mathcal{FL}_0$	●		●		
the extended work of sim (Tongphu & Suntisrivaraporn, 2015)	$\mathcal{ELH}$	●				●
$simi$ (Lehmann & Turhan, 2012)	$\mathcal{ELH}$	○		●	○	

CHAPTER 5
COMPUTATION OF ACCEPTED ANALOGICAL ARGUMENTS

We have discussed the theoretical analysis of using ABA framework to model the argumentation scheme for argument from analogy and concept similarity under preferences for understanding the degree of similarity between concepts in Section 2.1 and Section 3.2, respectively. Though using ABA alone could model the argumentation scheme for argument from analogy, it came up with several difficulties as follows.

First, ABA does not concretely describe where the source of similarity premises comes from, how a notion of concept similarity should be involved, how ‘relevance’ of concept similarity is defined and effects the degree of analogical arguments, and how analogical arguments should interact with normal arguments in case of persuasion. These problems are basically related to redefining both the notion of structured arguments and the framework in a way that arguments’ types can be classified.

Second, an analogical argument should be associated with a particular degree since each analogy used to support a claim is associated with a unit interval [0, 1]. This degree should also contribute to the attack relation between arguments. It is worth mentioning that similarity could be ‘qualitative’ in a sense that ones may only perceive if two concepts are similar or not. In this case, a certain threshold should be defined for being similar and each analogical argument could be associated with a binary [0, 1] where 1 indicates ‘similar’ and 0 indicates ‘not similar’.

Third, different rational agents may value arguments supported by analogies unequally, depending on their characteristics. This point is related to different styles of making judgment. For example, there could be a ‘gullible’ agent who always gives a high degree on every analogical argument; or a ‘skeptical’ agent vice versa.

To address the first difficulty, we extend the original ABA framework to *assumption-based argumentation with predicate similarity* (denoted by $ABA^{(p)}$) by identifying necessary components to form analogical arguments. In the following, the extended framework considers any arbitrary description language although DL terminological formalism is used in our running example.

Definition 5.1. An ABA[(p)] is a 10-tuple $\langle \mathcal{L}_D, \mathcal{R}, \mathcal{A}, \overline{}, \mathcal{L}_T, \mathcal{T}, \ ^{\mathcal{M}}, \sim_\tau^{\mathfrak{p}}, \mathfrak{p}, \mathcal{F} \rangle$ where $(\mathcal{L}_T, \mathcal{T})$ is a module formalizing descriptions of concepts with a language $\mathcal{L}_T$ and a set $\mathcal{T}$ of formulae (constructed from $\mathcal{L}_T$) representing definitions of concepts, $^{\mathcal{M}}$is a partial mapping from the predicate of sentences in $\mathcal{L}_D$ to concepts in $\mathcal{L}_T$, $\sim_\tau^{\mathfrak{p}}: \mathcal{L}_T \times \mathcal{L}_T \to [0,1]$ is a certain concept similarity w.r.t. $\mathcal{T}$ under preference context $\mathfrak{p}$, $\mathcal{F}$ is an annotation function for each entire argument to a numerical value[19], $\overline{}$ is a total function mapping from $\mathcal{A} \cup \mathcal{AN}$, where $\mathcal{AN} := \{P \sim_\tau^{\mathfrak{p}} Q \mid P^{\mathcal{M}} \sim_\tau^{\mathfrak{p}} Q^{\mathcal{M}} \in (0,1]$, for any $P(t_1, \ldots, t_p), Q(t_1, \ldots, t_p) \in \mathcal{L}_D\}$[20] representing a set of analogies, and $\mathcal{L}_D, \mathcal{R}, \mathcal{A}$ are as defined in ABA framework. An *argument* for $c \in \mathcal{L}_D$ (the *conclusion* or *claim*) supported by $\mathcal{S} \subseteq \mathcal{A} \cup \mathcal{AN}$, is a tree with nodes labeled by sentences in $\mathcal{L}_D \cup \mathcal{AN}$, by sentences of the special form $?(\varphi, \psi, \xi)$ representing a defeasible condition of sentence φ concluded from an analogy between ψ and ξ, or by the special symbol □ representing an empty set of premises, such that:

- the root is labelled by *c*;
- for every node *N*,
  - if *N* is a leaf, then *N* is labeled by an assumption in $\mathcal{A} \cup \mathcal{AN}$, an assumption of the form $?(\varphi, \psi, \xi)$, or by □,
  - if *N* is not a leaf, l_N is the label of *N*, and there is an inference rule $l_N \leftarrow b_1, \ldots, b_m$ $(m \geq 0)$ in $\mathcal{R}$, then
    - either $m = 0$, and the child of *N* is □
    - or $m > 0$ and *N* has m children, labeled by $b_1, \ldots, b_m$, respectively,
  - if *N* is not a leaf, l_N is the label of *N* where $l_N := P(t_1, \ldots, t_p)$, there is an analogy $P \sim_\tau^{\mathfrak{p}} Q$ in $\mathcal{AN}$, and there is either an inference rule $Q(t_1, \ldots, t_p) \leftarrow b_1, \ldots, b_m$ $(m \geq 0)$ in $\mathcal{R}$ or $Q(t_1, \ldots, t_p)$ in $\mathcal{A}$, then
    - *N* has 3 children, labeled by $P \sim_\tau^{\mathfrak{p}} Q$, $?(l_N, P, Q)$, $Q(t_1, \ldots, t_p)$;
- $\mathcal{S}$ is the set of all assumptions labelling the leaves.

---

[19] See Definition 6, for its formal definition.
[20] If $p = 0$, both *P* and *Q* are called *propositions.*

$(\mathcal{L}_T, \mathcal{T})$ can be defined for any kinds of terminological formalism specified by means of a language $\mathcal{L}_T$ and a set of formulae $\mathcal{T}$. For example, a DL terminological knowledge base can be recast as $\mathcal{L}_T := \text{CN} \cup \text{RN}$ and $\mathcal{T}$ is a TBox constructed from $\mathcal{L}_T$. Furthermore, a choice of terminological formalism can affect particular constraints. For instance, if $(\mathcal{L}_T, \mathcal{T})$ represents a DL knowledge base, then the predicates in $(\mathcal{L}_D, \mathcal{R})$ which has a mapping in $\mathcal{L}_T$ must be unary.

We note that $?(\varphi, \psi, \xi)$ can be read as "conclusion φ supported by an analogy between ψ and ξ is opened for challenging". A challenge of φ could be the contrary of φ, which may be possibly drawn from other analogies (*aka.* counter-analogies) or chains of inference rules. For example, a challenge of "*sound*$_2$ created by *bird*$_2$ is duck's sound" is an evidence that *sound*$_2$ is *honk* sound. Like ABA, assumptions are the only defeasible component in ABA$^{(p)}$ and they are used to support a conclusion. For the sake of simplicity, we clearly separate analogical assumptions from standard assumptions. That is, an argument for c supported by standard assumption $\mathcal{S}^{\mathcal{A}} \subseteq \mathcal{A}$ and analogical assumption $\mathcal{S}^{\mathcal{AN}} := \mathcal{S} \setminus \mathcal{S}^{\mathcal{A}}$ is denoted by $\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c$ (*i.e.* $\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} = \mathcal{S}$ such that $\mathcal{S}^{\mathcal{A}} \cap \mathcal{S}^{\mathcal{AN}} = \emptyset$). When $\mathcal{S}^{\mathcal{AN}}$ is empty *i.e.* $\mathcal{S}^{\mathcal{A}} \cup \emptyset \vdash c$, we call such an argument a *standard argument*. Otherwise, we call it an *analogical argument*. This style of writing helps recognizing analogical arguments and standard arguments at first glance.

It is worth noting that the study of analogical reasoning in logical systems is not new since several studies do exist. For example, (Goebel, 1989a) provided a form of analogical reasoning in terms of a system of hypothetical reasoning, (Sun, 1995b) integrated rule-based and similarity-based reasoning in a connectionist model. In argumentation systems, (Racharak, Tojo, Hung, & Boonkwan, 2016) studied an implementation of analogical reasoning using an argument-based logic programming and (Racharak et al., 2017b) proposed an idea to combine answer set programming with description logic. This work makes a continuous study of these papers by generalizing (Racharak et al., 2017b) to ABA.

To address the second difficulty, we define the function $f: \mathcal{S} \to [0,1]$ for annotating (both standard and analogical) assumptions as follows:

Definition 5.2. Given a set $\mathcal{S}$ of assumptions, a partial mapping $^{\mathcal{M}}$ from the predicate of sentences in $\mathcal{L}_D$ to concepts in $\mathcal{L}_T$, and $\sim_\tau^{\mathfrak{p}}: \mathcal{L}_T \times \mathcal{L}_T \to [0,1]$ is a certain concept similarity w.r.t. terminological formalism $\mathcal{T}$ under preference context $\mathfrak{p}$, the (total) annotation function $f: \mathcal{S} \to [0,1]$ is defined, for any $a \in \mathcal{S}$, as:

$$f(a) = \begin{cases} P^{\mathcal{M}} \sim_\tau^{\mathfrak{p}} Q^{\mathcal{M}} \text{ if } a \text{ is of the form } P \sim_\tau^{\mathfrak{p}} Q \\ P^{\mathcal{M}} \sim_\tau^{\mathfrak{p}} Q^{\mathcal{M}} \text{ if } a \text{ is of the form } ?(l_N, P, Q) \\ 1 \text{ otherwise} \end{cases} \tag{5.1}$$

Intuitively, standard assumptions are labeled with 1 to correspond with the fact that similarity relation is bound by 1 (we note that 1 is used in $\sim_\tau^{\mathfrak{p}}$ to indicate the maximal similarity). Next, we extend f to the function $\mathcal{F}$ for annotating arguments. Each annotation represents the degree of each entire argument.

Definition 5.3. Let $\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c$ be an argument. Then, a function $\mathcal{F}$ for annotating an entire argument is defined as:

$$\mathcal{F}(\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c) = \begin{cases} \otimes\{f(a_i), f(an_j)\} \text{ if } \mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \neq \emptyset \\ 1 \text{ otherwise} \end{cases} \tag{5.2}$$

where $a_i \in \mathcal{S}^{\mathcal{A}}$, $an_j \in \mathcal{S}^{\mathcal{AN}}$, and $\otimes$ is a triangular norm (t-norm).

Since the above definition employs the notion of t-norm, we include its basis here for self-containment. A function $\otimes: [0,1]^2 \to [0,1]$ is called a *t-norm* iff it fulfills the following properties for all $x, y, z, w \in [0,1]$: (1) $x \otimes y = y \otimes x$ (commutativity); (2) $x \leq z$ and $y \leq w \Rightarrow x \otimes y \leq z \otimes w$ (monotonicity); (3) $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ (associativity); (4) $x \otimes 1 = x$ (identity). A t-norm is called *bounded* iff $x \otimes y = 0 \Rightarrow x = 0$ or $y = 0$. There are several reasons for the use of a t-norm. Firstly, it is the generalization of the conjunction in propositional logic. Secondly, the operator *min* (*i.e.* $x \otimes y = min\{x, y\}$) is an instance of a bounded t-norm. This reflects an intuition that the strength of an argument depends on the used 'weakest' analogical assumptions. Lastly, 1 acts as the neutral element for t-norms.

Concerning the third difficulty, the choice of $\otimes$ (*cf.* Table 1 for its examples) can represent a type of a rational agent in analogical reasoning. For example, a gullible/skeptical agent may give a high/low degree to his answer when his answer is derived from analogies. We formalize this characteristic as follows[21].

Definition 5.4. Let $\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c$ be an argument; also, $\mathcal{F}_1$ and $\mathcal{F}_2$ be two different functions representing different agents. Then, $\mathcal{F}_1$ is *more gullible* than $\mathcal{F}_2$ if $\mathcal{F}_1(\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c) \geq \mathcal{F}_2(\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c)$. On the other hand, $\mathcal{F}_1$ is *more skeptical* than $\mathcal{F}_2$ if $\mathcal{F}_1(\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c) \leq \mathcal{F}_2(\mathcal{S}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{AN}} \vdash c)$. Lastly, $\mathcal{F}_1$ and $\mathcal{F}_2$ are *identical* if $\mathcal{F}_1$ are *both gullible and skeptical* to $\mathcal{F}_2$.

The following theorem is an aid to help deciding which operator $\otimes$ should be chosen for $\mathcal{F}$ in ABA$^{(p)}$. That is, if an agent strongly recognizes analogical principles, we may choose the most gullible function (*i.e.* $\otimes_{min}$). On the other hand, we may choose the skeptical function (*i.e.* $\otimes_{\text{mlt}}$) if an agent weakly recognizes analogical principles.

Table 5.1 Some Instances of The Operator $\otimes$.

Name	Notation	$x_1 \otimes x_2 =$
Minimum	$\otimes_{min}$	$min\{x_1, x_2\}$
Product	$\otimes_{\text{mlt}}$	$x_1 \cdot x_2$
Hamacher product	$\otimes_{H_0}$	0 if $x_1 = x_2$; otherwise $\dfrac{x_1 \cdot x_2}{x_1 + x_2 - x_1 \cdot x_2}$

Theorem 5.1. From Table 5.1 and let $x_1, x_2 \in (0,1]$. Then, $\otimes_{\text{mlt}} \leq \otimes_{H_0} \leq \otimes_{min}$.

Proof. (Sketch) We show the following inequality:

$$x_1 \cdot x_2 \leq \frac{x_1 \cdot x_2}{x_1 + x_2 - x_1 \cdot x_2} \leq min\{x_1, x_2\}$$

That is, we show $x_1 \cdot x_2 \leq \frac{x_1 \cdot x_2}{x_1 + x_2 - x_1 \cdot x_2}$ as follows:

$$x_1 \cdot x_2 \leq \frac{x_1 \cdot x_2}{x_1 + x_2 - x_1 \cdot x_2} \Leftrightarrow 1 \leq \frac{1}{x_1 + x_2 - x_1 \cdot x_2} \Leftrightarrow x_1 + x_2 - x_1 \cdot x_2 \leq 1$$

[21] The choice of $\sim_\tau^{\mathrm{p}}$ also contributes to the type of a rational agent. That is, different concrete measures may have different skepticism. However, the definition only pays attention to how gullible is contributed from $\mathcal{F}$.

$\Leftrightarrow x_2 - x_1 \cdot x_2 \leq 1 - x_1 \Leftrightarrow (1 - x_1) \cdot x_2 \leq 1 - x_1 \Leftrightarrow x_2 \leq 1$ (by assumption)

Lastly, we show $\frac{x_1 \cdot x_2}{x_1 + x_2 - x_1 \cdot x_2} \leq \min\{x_1, x_2\}$ in the similar fashion.

Attacks in ABA are defined in terms of the contrary of assumptions (*cf.* Subsection Structured Argumentation). However, argument trees and their supporting assumptions in $\mathrm{ABA}^{(\mathrm{p})}$ are labeled with numbers. This is clear that the current definition of attacks in ABA is not appropriate for handling attacks in $\mathrm{ABA}^{(\mathrm{p})}$. To define the notion of attacks in $\mathrm{ABA}^{(\mathrm{p})}$, we extend the original definition of attacks in ABA to take into account the numbers. In addition, the extended definition imposes a particular restriction on the usage of analogical reasoning for 'persuasion' *i.e.* analogical arguments are always preferable to standard arguments. These characteristics are formally defined as follows.

Definition 5.5. Let function $\overline{}$, function $\mathcal{F}$, and function *f* be as defined in Definition 4, Definition 5, and Definition 6, respectively. An argument $\mathcal{S}_1^{\mathcal{A}} \cup \mathcal{S}_1^{\mathcal{AN}} \vdash c_1$ *attacks* an argument $\mathcal{S}_2^{\mathcal{A}} \cup \mathcal{S}_2^{\mathcal{AN}} \vdash c_2$ iff the following satisfies:

- If $\mathcal{S}_1^{\mathcal{AN}} \neq \emptyset$ and $\mathcal{S}_2^{\mathcal{AN}} = \emptyset$, then c_1 is the contrary of an assumption in $\mathcal{S}_2^{\mathcal{A}}$;
- Otherwise, c_1 is the contrary of an assumption $\mathcal{S}_2^{\mathcal{A}} \cup \mathcal{S}_2^{\mathcal{AN}}$ (*i.e.* $x \in \mathcal{S}_2^{\mathcal{A}} \cup \mathcal{S}_2^{\mathcal{AN}}$ and $\overline{c_1} = x$) and $\mathcal{F}(\mathcal{S}_1^{\mathcal{A}} \cup \mathcal{S}_1^{\mathcal{AN}} \vdash c_1) \geq f(x)$.

The first condition spells out that an analogical argument may attack a standard argument. This certain characteristic corresponds to the investigation in (Waller, 2001), where analogical arguments can be used for persuasion. For instance, saying "geese can quack because they are similar to ducks" may effect the belief's changing on the opponent if no evidences to falsify the argument can be shown up. To put it more precisely, an opponent can be persuaded to believe a conclusion and that conclusion is inherently subject to be challenged. Hence, the burden of proof is shifted back to an opponent after he/she is persuaded to believe in that conclusion.

The second condition associates with another circumstance *i.e.* an analogical argument can attack an assumption only if the argument has been labeled with the number higher than or equal to the number associated with the assumption. This way of treatment is not used in (Waller, 2001; Walton et al., 2008).

$\mathcal{R}$: $quack(B) \leftarrow duck(A, B)$; $honk(B) \leftarrow cord(A, B), built\_for\_honk(A)$;
$cord(cord_1, sound_2)$; $built\_for\_honk(cord_1)$

$\mathcal{A}$: $duck(bird_1, sound_1)$; $goose(bird_2, sound_2)$

$\overline{}$: $\overline{duck(bird_1, sound_1)} = \neg duck(bird_1, sound_1)$;
$\overline{?(duck(bird_2, sound_2), duck, goose)} = honk(sound_2)$;
$\overline{duck \sim^{\mathfrak{p}}_{\tau} goose} = duck \nsim^{\mathfrak{p}}_{\tau} goose$

$\mathcal{T}$: $\text{Duck} \sqsubseteq \text{WaterBird}$; $\text{Goose} \sqsubseteq \text{WaterBird}$

$\cdot^{\mathcal{M}}$: $duck^{\mathcal{M}} = \text{Duck}$; $goose^{\mathcal{M}} = \text{Goose}$

$\sim^{\mathfrak{p}}_{\tau}$: sim^{π}

$\mathfrak{p}$: π_0

Figure 5.1 ABA$^{(\mathfrak{p})}$ Framework for The Running Example.

Example 5.1. Figure 5.1 illustrates an overall ABA$^{(\mathfrak{p})}$ framework for the running example. According to the figure, the framework uses sim^{π} and π_0 as concrete instances of $\sim^{\mathfrak{p}}_{\tau}$ and $\mathfrak{p}$, respectively. The figure also uses $\nsim^{\mathfrak{p}}_{\tau}$ to indicate 'being not similar under preference context $\mathfrak{p}$ w.r.t. $\mathcal{T}$'. The following suggests two arguments which can be constructed from the framework.

- $\{goose(bird_2, sound_2) \cup$
  $duck \sim^{\mathfrak{p}}_{\tau} goose, ?(quack(sound_2), duck, goose) \vdash quack(sound_2)\}$
  representing "*sound*$_2$ created by *bird*$_2$ is quack sound because *bird*$_2$ is a goose and geese are similar to ducks";
- $\emptyset \vdash honk(sound_2)$ representing "*sound*$_2$ is honk sound".

Hence, the second argument attacks the first argument. It is also worth observing that, in this case, varying each choice of $\otimes$ does not effect on the attack relation between these two arguments even though the degree of an argument is changed. For example, if $\otimes_{min}$ is used, then the degree of the first argument is equal to 0.5. On the other hand, if $\otimes_{\text{mlt}}$ is used, then the degree of the first argument is equal to 0.25.

The following theorizes an observation which can be derived from Definition 5.5.

Theorem 5.2. An analogical argument cannot attack a standard argument which does not use assumptions to support a claim.

Proof. Let argument $\mathcal{G}_1$ be defined as $\mathcal{S}_1^{\mathcal{A}} \cup \mathcal{S}_1^{\mathcal{AN}} \vdash c_1$ and argument $\mathcal{G}_2$ be defined as $\emptyset \vdash c_2$. We need to show that $\mathcal{G}_1$ cannot attack $\mathcal{G}_2$.

Since $\mathcal{G}_2$ contains no assumptions, we conclude that $\mathcal{G}_1$ cannot attack $\mathcal{G}_2$.

Theorem 5.2 shows that when an agent supports a claim from the grounded truth, it is impossible for other agents to persuade him/her by analogies. This corresponds to how analogical arguments are treated in practical reasoning.

5.1 Acceptability of Arguments in ABA$^{(p)}$

ABA$^{(p)}$ extends from ABA by equipping with predicate similarity and its attack definition is also extended for handling the degree of each argument and the preference between different types of arguments. Hence, ABA$^{(p)}$ can be considered as an instance of Dung's abstract argumentation. This implies that it can be used to determine whether a given claim is 'accepted' by a rational agent. In a sense of analogical argumentation, the claim could be a potential belief to be justified from analogies.

In order to determine the 'acceptability' of a claim, the agent needs to find an argument for the claim that can be defended against attacks from other arguments. To defend an argument, other arguments must be found and may need to be defended in turn (Dung et al., 2009). We formally define these characteristics as follows:

- A set of arguments Arg_1 *attacks* a set of arguments Arg_2 if an argument in Arg_1 attacks an argument in Arg_2;
- A set of arguments *Arg defends* an argument *arg* if *Arg* attacks all arguments that attack $\{arg\}$.

As in Dung's abstract argumentation, the notion of 'acceptability' can be formalized in many ways. In this work, we focus on the following notions:

- A set of arguments is *admissible* iff it does not attack itself and it attacks every argument that attacks it;
- An admissible set of arguments is *complete* if it contains all arguments that it defends;

- The least (w.r.t. set inclusion) complete set of arguments is *grounded*.

We observe that the correspondence between 'acceptability' of arguments and 'acceptability' of assumptions in ABA$^{(p)}$ can be argued in the same way as in (Dung et al., 2007) for the link between ABA and AA. Hence, we know:

- If a set of assumptions S is admissible/grounded, then the union of all arguments supported by any subset of S is admissible/grounded;
- If a set of arguments S is admissible/grounded, then the union of all sets of assumptions supporting the arguments in S is admissible/grounded.

The above notion of acceptable sets of arguments provides a non-constructive specification. Now, we show how to turn the specification into a constructive proof procedure. The method we focus here is defined for a 'grounded' set of arguments and is extended from (Dung et al., 2007) for handling analogical arguments.

Informally, this constructive proof procedure is known as a *dispute derivation* which is defined as a sequence of transition steps from one state of a dispute to another. For each state, we maintain these following information. Component $\mathcal{P}$ maintains a set of (both standard and analogical) assumptions, which are used to support potential arguments of the proponent. Component $\mathcal{O}$ maintains multiple sets of assumptions, which are used to support all attacking arguments of the opponent. Component D holds a set of assumptions, which have already been used by the proponent. Component C holds a set of assumptions, which have already been used by the opponent and have been attacked by the proponent. Component SP maintains a set of triples holding an opponent's attacked assumption, a set of proponent's assumptions supporting a contrary of the attacked assumption, and a set of opponent's assumptions supporting the argument. Component SO maintains a set of triples holding a proponent's attacked assumption, a set of proponent's assumptions supporting the argument, and a set of opponent's assumptions supporting a contrary of the attacked assumption. In the following, we formally define the dispute derivation for a 'grounded' set of arguments.

Definition 5.6. Let an ABA$^{(p)}$ is a 10-tuple $\langle \mathcal{L}_D, \mathcal{R}, \mathcal{A}, \overline{}, \mathcal{L}_T, \mathcal{T}, \quad ^{\mathcal{M}}, \sim^{\mathfrak{p}}_{\tau}, \mathfrak{p}, \mathcal{F} \rangle$. Given a 'patient' selection function[22], a 'grounded belief' dispute derivation of a defence set Δ for a sentence δ is a finite sequence:

[22] A patient selection function always prefers a non-assumption to an assumption in its selection.

$\langle \mathcal{P}_0, \mathcal{O}_0, D_0, C_0, SP_0, SO_0 \rangle, \dots, \langle \mathcal{P}_i, \mathcal{O}_i, D_i, C_i, SP_i, SO_i \rangle, \dots, \langle \mathcal{P}_n, \mathcal{O}_n, D_n, C_n, SP_n, SO_n \rangle$

where $\mathcal{P}_0 := \{\{\delta\}\}$, $D_0 := \mathcal{A} \cap \{\delta\}$, $\mathcal{O}_0 := \emptyset$, $C_0 := \emptyset$, $\mathcal{P}_n := \{\emptyset\}$, $\mathcal{O}_n := \emptyset$, $SP_0 := \emptyset$, $SO_0 := \emptyset$, $\Delta := D_n$, and for every $0 \leq i < n$, only one *S* in $\mathcal{P}_i$ or one *S* in $\mathcal{O}_i$ is selected, and:

1. if *S* is selected in $\mathcal{P}_i$ and σ is selected in *S*, then
    a. if σ is an assumption, then

$\mathcal{P}_{i+1} := (\mathcal{P}_i \setminus \{S\}) \cup \{ S \setminus \{\sigma\}\}$, $\mathcal{O}_{i+1} := \mathcal{O}_i \cup \{\{\bar{\sigma}\}\}$,
and $SO_{i+1} := SO_i \cup \{\langle \sigma, S, \{\bar{\sigma}\}\rangle\}$

    b. else if there exists an inference rule $\sigma \leftarrow R \in \mathcal{R}$ such that $C_i \cap R = \emptyset$, then

$\mathcal{P}_{i+1} := (\mathcal{P}_i \setminus \{S\}) \cup \{ S \setminus \{\sigma\} \cup R\}$, $D_{i+1} := D_i \cup (\mathcal{A} \cap R)$,
and $SP_{i+1} := (SP_i \setminus \{\langle \varphi, PA, OA \rangle\}) \cup \{\langle \varphi, PA \setminus \{\varphi\} \cup R, OA \rangle\}$
for any $\langle \varphi, PA, OA \rangle \in SP_i$ such that $\sigma \in PA$

and if $R \subseteq \mathcal{A}$, then further validation needs to be checked:
for any $\langle \varphi, PA, OA \rangle \in SP_{i+1}$ such that $PA \cup OA \subseteq \mathcal{A} \cup \mathcal{AN}$, we have
either $PA \subseteq \mathcal{AN}$ and $OA \subseteq \mathcal{A}$
or $\mathcal{F}(PA) \geq \mathcal{F}(\varphi)$

    c. else if $\sigma := P(t_1, \dots, t_p)$ and there exists $\phi := Q(t_1, \dots, t_p)$ such that $P^{\mathcal{M}} \sim_\tau^{\mathfrak{p}} Q^{\mathcal{M}} \in (0,1]$, then

$\mathcal{P}_{i+1} := (\mathcal{P}_i \setminus \{S\}) \cup \{S \setminus \{\sigma\} \cup \{P \sim_\tau^{\mathfrak{p}} Q, ?(\sigma, P, Q), \phi\}\}$,
$D_{i+1} := D_i \cup \{P \sim_\tau^{\mathfrak{p}} Q, ?(\sigma, P, Q)\} \cup (\mathcal{A} \cap \{\phi\})$,
and $SP_{i+1} := (SP_i \setminus \{\langle \varphi, PA, OA \rangle\}) \cup \{\langle \varphi, PA \setminus \{\sigma\} \cup \{P \sim_\tau^{\mathfrak{p}} Q, ?(\sigma, P, Q), \phi\}, OA \rangle\}$
for any $\langle \varphi, PA, OA \rangle \in SP_i$ such that $\sigma \in PA$

and if $\phi \in \mathcal{A}$, then the same validation as in Case 1.b is required

2. if S is selected in $\mathcal{O}_i$ and σ is selected in S, then
   a. if σ is an assumption, then
      i. either σ is ignored *i.e.*

$$\mathcal{O}_{i+1} := (\mathcal{O}_i \setminus \{S\}) \cup \{S \setminus \{\sigma\}\}$$

      ii. or $\sigma \notin D_i$ and

$\mathcal{O}_{i+1} := \mathcal{O}_i \setminus \{S\}$, $\mathcal{P}_{i+1} := \mathcal{P}_i \cup \{\{\bar{\sigma}\}\}$, $D_{i+1} := D_i \cup (\{\bar{\sigma}\} \cap \mathcal{A})$,
$C_{i+1} := C_i \cup \{\sigma\}$, and $SP_{i+1} := SP_i \cup \{\langle \sigma, \{\bar{\sigma}\}, S\rangle\}$

   b. else if $\mathcal{A} := \{\, R \mid \sigma \leftarrow R \in \mathcal{R}\}$ and $A \neq \emptyset$, then

$$\mathcal{O}_{i+1} := (\mathcal{O}_i \setminus \{S\}) \cup \bigcup_{R \in A} \{S \setminus \{\sigma\} \cup R\}$$

and $SO_{i+1} := (SO_i \setminus \{\langle \varphi, PA, OA\rangle\}) \cup \bigcup_{R \in A} \{\langle \varphi, PA, OA \setminus \{\sigma\} \cup R\rangle\}$
for any $\langle \varphi, PA, OA\rangle \in SO_i$ such that $\sigma \in OA$

and further validation must be satisfied:
for any $\langle \varphi, PA, OA\rangle \in SO_{i+1}$ such that $PA \cup OA \subseteq \mathcal{A} \cup \mathcal{AN}$, we have
either $OA \subseteq \mathcal{AN}$ and $PA \subseteq \mathcal{A}$
or $\mathcal{F}(OA) \geq \mathcal{F}(\varphi)$

   c. else if $\sigma \triangleq P(t_1, \ldots, t_p)$, $A := \{\, Q(t_1, \ldots, t_p) \mid P^{\mathcal{M}} \sim_{\tau}^{\mathrm{p}} Q^{\mathcal{M}} \in (0,1]\}$, and $A \neq \emptyset$, then

$$\mathcal{O}_{i+1} := (\mathcal{O}_i \setminus \{S\}) \cup \bigcup_{Q(t_1,\ldots,t_p) \in A} \left\{S \setminus \{\sigma\} \cup \left\{P \sim_{\tau}^{\mathrm{p}} Q, ?(\sigma, P, Q), Q\big(t_1, \ldots, t_p\big)\right\}\right\}$$

and $SO_{i+1} := (SO_i \setminus \{\langle \varphi, PA, OA\rangle\}) \cup$

$$\bigcup_{Q(t_1,\ldots,t_p) \in A} \{\langle \varphi, PA, OA \setminus \{\sigma\} \cup \left\{P \sim_{\tau}^{\mathrm{p}} Q, ?(\sigma, P, Q), Q\big(t_1, \ldots, t_p\big)\right\}\rangle\}$$

for any $\langle \varphi, PA, OA\rangle \in SO_i$ such that $\sigma \in OA$

plus, the same validation as in Case 2.b is required

d. else $\mathcal{O}_{i+1} := \mathcal{O}_i \setminus \{S\}$ and $SO_{i+1} := SO_i \setminus \{ \langle \varphi, PA, OA \rangle \mid \langle \varphi, PA, OA \rangle \in SO_i$ and $PO = S \}$

A dispute derivation can be seen as a way of representing a 'potential' winning strategy for a proponent to win a dispute against an opponent. The proponent starts by putting forward a claim whose acceptability is under dispute. After that, there are many possibilities as follows. The opponent can try to attack the proponent's claim by arguing for its contrary (*cf.* Case 1.a). The proponent argues for a non-assumption by using an inference rule (*cf.* Case 1.b). If an inference rule does not exist, the proponent can use an analogy to support the initial claim (*cf.* Case 1.c). Moreover, the proponent can select an assumption in one of the opponent's attacks and either ignores it because it is not selected as a culprit (*cf.* Case 2.a.i) or decides to counter-attack it by showing its contrary (*cf.* Case 2.a.ii). Otherwise, the opponent can argue for a non-assumption by using either an inference rule (*cf.* Case 2.b) or an analogy (*cf.* Case 2.c). Unfortunately, the opponent may not have even a reason to argue for it (*cf.* Case 2.d). In addition, every attacking argument of the opponent to the proponent's claim is maintained inside SO *i.e.* $\langle \sigma, S, \{\bar{\sigma}\} \rangle$ is read as "assumption σ in a set of proponent's assumptions S is attacked by a set of assumptions $\{\bar{\sigma}\}$". Every attacking argument of the proponent to the opponent's claim is also maintained inside SP *i.e.* $\langle \sigma, \{\bar{\sigma}\}, S \rangle$ is read as "assumption σ in a set of opponent's assumptions S is attacked by a set of assumptions $\{\bar{\sigma}\}$".

We give an informal dispute derivation for the running example.

Example 5.2. Consider an ABA$^{(p)}$ given in Figure 5.1 and let $\otimes_{min}$ be used. Table 2 shows that there does not exist a grounded belief dispute derivation for $quack(sound_2)$, where[23] ♥, ♣$_1$, ♣$_2$, ♣$_3$, ♣$_4$, ♠$_1$, and ♠$_2$ denotes $\{d \sim_\tau^{\mathrm{p}} g, ?\,(d(b_2, s_2), d, g), g(b_2, s_2)\}$, $\langle ?\,(d(b_2, s_2), d, g), ♥, \{h(s_2)\} \rangle$, $\langle ?\,(d(b_2, s_2), d, g), ♥, \{c(c_1, c_2), bfh(c_1)\} \rangle$, $\langle ?\,(d(b_2, s_2), d, g), ♥, \{bfh(c_1)\} \rangle$, $\langle ?\,(d(b_2, s_2), d, g), ♥, \emptyset \rangle$, $\langle g(b_2, s_2), \{d \sim_\tau^{\mathrm{p}} g, g(b_2, s_2)\}, \{\neg g(b_2, s_2)\} \rangle$, and $\langle g(b_2, s_2), \{d \sim_\tau^{\mathrm{p}} g, g(b_2, s_2)\}, \{d \nsim_\tau^{\mathrm{p}} g\} \rangle$, respectively.

[23] Obvious abbreviations are used here for the sake of succinctness.

At step 2, the proponent ($\mathcal{P}$) has completed the construction of an argument for $q(s_2)$ supported by ♥, saying that "s_2 is a quack sound because goose b_2 makes s_2 and geese are similar to ducks". At step 3, the opponent ($\mathcal{O}$) has decided to attack on assumption $?(d(b_2, s_2), d, g)$ by showing its contrary $h(s_2)$. This argument is fully constructed at step 6, in which no assumptions have been used. Nonetheless, this attacking argument needs to be checked at SO_6 if it satisfies the requirements of argument from analogy. Since it satisfies, step 6 is valid. Finally, no arguments of the proponent can defend the opponent's argument at step 10, this dispute derivation fails.

With an analogous manner, we can find a grounded belief dispute derivation of $\{d(b_1, s_1)\}$ for $q(s_1)$ with three transition steps.

Table 5.2 A Grounded Belief Dispute Derivation for $quack(sound_2)$.

Step	$\mathcal{P}$	$\mathcal{O}$	D	C	SP	SO
0	$\{\{q(s_2)\}\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{\{d(b_2, s_2)\}\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{$♥$\}$	$\emptyset$	♥	$\emptyset$	$\emptyset$	$\emptyset$
3	$\{\{d \sim_\tau^{p} g, g(b_2, s_2)\}\}$	$\{\{h(s_2)\}\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_1\}$
4	$\{\{d \sim_\tau^{p} g, g(b_2, s_2)\}\}$	$\{\{c(c_1, c_2), bfh(c_1)\}\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_2\}$
5	$\{\{d \sim_\tau^{p} g, g(b_2, s_2)\}\}$	$\{\{bfh(c_1)\}\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_3\}$
6	$\{\{d \sim_\tau^{p} g, g(b_2, s_2)\}\}$	$\{\emptyset\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_4\}$
7	$\{\{d \sim_\tau^{p} g\}\}$	$\{\emptyset, \{\neg g(b_2, s_2)\}\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_4,$ ♠$_1\}$
8	$\{\{d \sim_\tau^{p} g\}\}$	$\{\emptyset\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_4\}$
9	$\{\emptyset\}$	$\{\emptyset, \{d \nsim_\tau^{p} g\}\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_4,$ ♠$_2\}$
10	$\{\emptyset\}$	$\{\emptyset\}$	♥	$\emptyset$	$\emptyset$	$\{$♣$_4\}$

5.2 Relationship to Argumentation Scheme for Argument from Analogy

Since ABA$^{(p)}$ extends from ABA with the capability for supporting the conclusion from similarity premises, the notion of argument trees in ABA$^{(p)}$ can be also used to display the structural relationships between conclusions and assumptions including standard assumptions and analogical assumptions. Figure 5.2 illustrates an example of argument trees for arguments discussed in Example 5.1. The figure uses a rounded rectangle for indicating an argument tree, a number floating near a rounded rectangle for indicating an annotated degree of that entire argument, a number floating near an assumption for indicating an annotated degree of that assumption, and a dashed arrow for indicating an attack. For example, the top rounded rectangle shows the structural relationship of argument "*sound*$_2$ created by *bird*$_2$ is quack sound of ducks because ducks are similar to geese and we know that *bird*$_2$, which is a goose, creates *sound*$_2$" whereas the bottom rounded rectangle shows the structural relationship of argument "*sound*$_2$ is honk sound because it is created from *cord*$_1$ and that cord is built for honk". The figure also depicts that the bottom one attacks the top one.

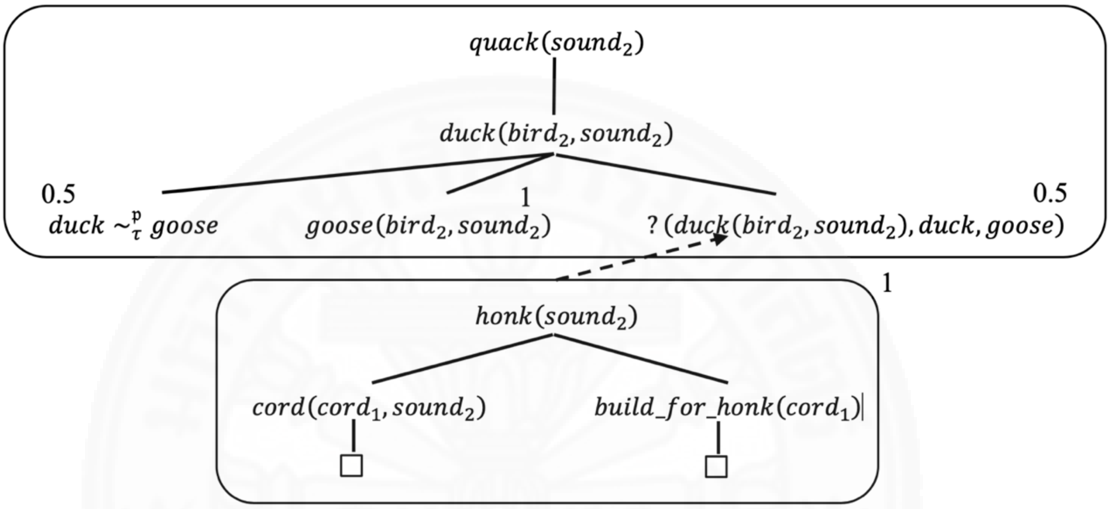

Figure 5.2 An Example of Argument Trees and Their Relationship.

Ones may observe that the structural relationship represented by an argument tree directly corresponds to the relationship between premises and a conclusion used in the argumentation scheme. That is, a similarity premise appears as an assumption of the form $P \sim^{p}_{\tau} Q$ and a base premise appears as either an assumption in $\mathcal{A}$ or an inference rule with the empty body in $\mathcal{R}$. They appear as nodes in an argument tree. A conclusion drawn from the use of the argumentation scheme is represented as a parent of those nodes in an argument tree. This structure clearly explains the relationship indicated in the argumentation scheme.

The critical questions can also be captured in ABA$^{(p)}$. Let us repeat that page 2 writes down each critical question (CQ) matching the scheme argument from analogy. Firstly, asking CQ1 is captured by the provability of a claim *i.e.* a backward deduction from a claim to its supporting assumptions. Secondly, CQ2 and CQ3 are formalized by the use of a similarity measure together with a supplied terminological formalism. Since similarity measure of concepts identifies the degree of commonalities, it automatically models the questions. Lastly, the notion of counter-analogies can be also modeled by the construction of arguments from another analogies drawing the contrary of the defeasible condition of the former argument.

Argumentation schemes employ the idea of asking critical questions to evaluate the acceptability of generated arguments. In $\mathrm{ABA}^{(\mathrm{p})}$, we evaluate by employing the notion of attack together with a semantics of argumentation framework (Dung, 1995) insisting that sets of acceptable arguments do not attack themselves and counter-attack all the opponent's arguments (*aka.* admissible sets of arguments).

5.3 Comparison with Related Works

There were attempts on modeling analogical reasoning including these recent work (Racharak et al., 2017b; Racharak, Tojo, et al., 2016) in which their results are continued to study in this work. We note that both formalized the scheme argument from analogy and provided a logical language which enables finding analogical conclusions. On the other hand, (Racharak, Tojo, et al., 2016) extended syntax and argumentative features of DeLP for handling analogical arguments whereas (Racharak et al., 2017b) translated the logical language to the represented answer set program and an answer set solver would be used to compute analogical conclusions. As (Racharak, Tojo, et al., 2016) extended DeLP, this work differs to (Racharak, Tojo, et al., 2016) in the structure of an argument's notion. Another difference is that (Racharak, Tojo, et al., 2016) is more computationally oriented and has restricted expressiveness whereas $\mathrm{ABA}^{(\mathrm{p})}$, like ABA, is a more general framework for analogical argumentation. With (Racharak et al., 2017), it is worth observing that their definition of knowledge base can be captured by an $\mathrm{ABA}^{(\mathrm{p})}$ framework. That is, a logic program $\mathcal{LP}$ is mapped to an ABA component, $\mathcal{O}$ is a concrete instance of $(\mathcal{L}_T, \mathcal{T})$, and $\sim^{\pi}_{\tau}$ is an abstract instance of $\sim^{\mathrm{p}}_{\tau}$. However, the development in (Racharak et al., 2017b) ignored analogical degrees in their computational method. We have completed that part and generalized the approach in this work.

A similar attempt to (Racharak et al., 2017b; Racharak, Tojo, et al., 2016), *i.e.* combing rules and similarities, was proposed in (Sun, 1995b). In that work, a two-level connectionist model was developed. The first level (called CL) had one node for each domain concept whereas the second level (called CD) had fine-grained features in which all domain concepts could be decomposed to. Characteristics of similarity measures (denoted by $\sim$ in (Sun, 1995b)) was also discussed and the formula based on the above two-level model was proposed for concepts A,B as: $A \sim B = (|F_A \cap F_B|)/(|F_B|)$ where F_A, F_B are features defined in CD. It is worth observing that those two levels and similarity formula can be represented as $(\mathcal{L}_T, \mathcal{T})$ and $\sim^{\mathrm{p}}_{\tau}$, respectively. However, how defeasible conditions and the notion of relevence should be handled was not discussed concretely.

In (Goebel, 1989a), the form of analogical reasoning was cast as hypothetical reasoning as: *source knowledge* $\cup$ *target knowledge* $\cup$ *equality assumptions* $\models$ *conclusions* where equality assumptions can be viewed as similarity between the source and the target. If there were many equality assumptions, certain explicit preferences, *e.g.* the highest number of shared properties, were used. However, the defeasible conditions and the notion of relevance were also not concretely discussed. It is also worth observing that source knowledge and target knowledge can also be recast in $(\mathcal{L}_T, \mathcal{T})$ and the criterion for forming equality assumptions can be made explicitly in $\sim^{\mathrm{p}}_{\tau}$.

In (Haraguchi & Arikawa, 1987) , the source domain and the target domain were represented by logic programs and their intended models were the least (Herbrand) model of them. Then, an analogy was considered as a partial identity between their least models. According to their definitions, the partial identity was a function mapping two different ground terms of two different domains so that the compatible mapped terms could be treated like a single term in the analogy-based reasoning process. Technically, the authors defined a formal reasoning based on the partial identity as an admissible method to extend the least (Herbrand) model. It is worth noticing that the authors used the partial identity to transfer knowledge about ground terms from the source domain to the target domain; thus, the terms were applicable if their related predicate symbols were the same i.e. the knowledge was transferred in the level of terms. This point differs to our work in a sense that knowledge was transferred in the level of predicates using similarity measure between two concepts in terminological formalism. Other different points are that the reasoning process proposed in (Haraguchi & Arikawa, 1987) was monotonic and did not consider the relevancy.

Case-based reasoning (CBR) can also be viewed as a form of analogical reasoning. In CBR, dimensions and factors are used for comparing cases and the decision in the precedent case is then taken as the decision into the current case. Examples of CBR systems are HYPO (Ashley, 2006) and CATO (Aleven, 1997). With $ABA^{(p)}$, CBR can be recast by consisting the rules: $c_i \leftarrow f_1, \ldots, f_n$ in $\mathcal{T}$, the rules: $p_i \leftarrow c_i$ in $\mathcal{R}$, and similarity between two cases c_i is measured from their common features f_i.

Comparing this work with defeasible reasoning formalism, particularly Nute's d-Prolog (Gabbay, Hogger, & Robinson, 1998, pp.353-396), different forms of rules were introduced *viz.* strict (unchallengeable) rules, defeasible (challengeable) rules, and defeater (exceptionable) rules. Examples of strict rules, defeasible rules, and defeater rules are "all penguins are bird", "birds normally fly", and "sick birds do not fly", respectively. Like ABA, inference rules in $ABA^{(p)}$ can be seen as strict rules and a simple transformation (as used in Theorist (Poole, 1988)) can be employed to convert defeasible rules into strict rules with assumptions. Moreover, we may observe that $ABA^{(p)}$ does not need to supply with defeater rules since it can find counter-arguments, including counter-analogies, among arguments it is able to build.

Ones may would like to compare between $ABA^{(p)}$ and an abstract framework of argumentation equipped with a preorder relation *e.g.* preference-based argumentation framework (PAF) introduced in (Amgoud & Cayrol, 2002). Formally, a PAF is a triple $\langle Args, Attack, \preccurlyeq \rangle$ where *Args* is a set of arguments, *Attack* is an attack relation, and $\preccurlyeq$ is used to define a 'defeat' relation on each attack. It is not difficult to observe the correspondence between an $ABA^{(p)}$ framework and a PAF framework. Informally, each argument tree in $ABA^{(p)}$ is mapped to an argument in *Args* and an attack in $ABA^{(p)}$ between argument trees is mapped to a defeat relation, in which the usage of an argument's degree and the preference on analogical arguments can be captured in a preorder relation. Their further theoretical relationship is left for future work.

CHAPTER 6
CONCLUDING REMARKS

This thesis investigated and formally defined a structured argumentation framework called ABA$^{(p)}$, which formalizes the argumentation scheme for argument from analogy. The main objectives of this work were to provide well understanding on the computational aspect of analogical reasoning in argumentation, rather than the psychological modeling. As a result, ABA$^{(p)}$ offers ways to encode the pattern of reasoning in argument from analogy and its critical questions, where concepts (or states of affairs) are represented by predicates in an underlying language and are defined by a particular terminological formalism (such as description logics). Its underlying mechanism consists in four mainstreams, *viz.* an ABA framework, a terminology, and a concept similarity under preferences, and a preference context. When no assumptions are available to construct an argument tree, additional assumptions can be constructed from the use of a similarity measure w.r.t. a terminology and a preference context. In other words, it draws a connection between two different formalisms, *i.e.* inference rules and terminological sentences, for dealing with analogical argumentation. Figure 6.1 shows a general review of our proposed framework, in which English alphabets represent examples of predicate symbols in the inference rules and concepts in the terminological formalism[24].

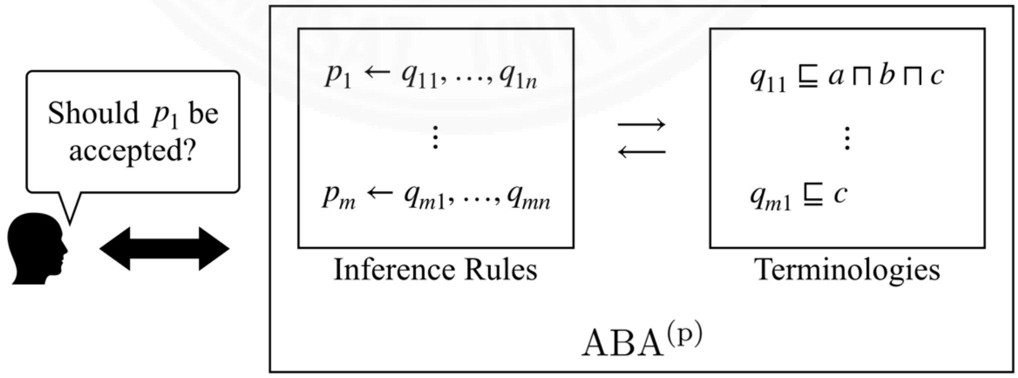

Figure 6.1 An Overview of Our ABA$^{(p)}$ Framework.

[24] Though, the syntax of description logics is used, other kinds of terminological formalisms are also supported as discussed in Chapter 5.

To achieve these goals, we exploited two different reasoning paradigms *viz.* rule reasoning and schemata reasoning. In particular, the original ABA was extended to incorporate with description logics for handling the acceptability of (analogical) arguments in question. Our framework uses the semantics of abstract argumentation and the degree of concept similarity is determined by aggregating subsumption degrees of two corresponding concepts. Figure 6.1 also indicates the interactions between the two reasoning paradigms. Specifically, we can view that when the rule reasoning cannot deductively infer new knowledge, it will make a query to the schemata reasoning so that the rule reasoning has more (potential) knowledge with a similarity score.

We also developed algorithmic procedures for evaluating warranted arguments and a numerical value indicating the similarity under subjective factors between concepts. In the following sections, major technical and empirical results of this thesis are discussed.

6.1 Discussion of Achieved Results

The major results achieved in this thesis can be classified as follows:

1. The development of concept similarity measure under preference profile in description logics (particularly, a sub-Boolean logic $\mathcal{ELH}$);
2. The design of algorithmic procedures for our proposed measure sim^{π} and their empirical evaluation w.r.t. realistic ontologies; and
3. The development of assumption-based argumentation with predicate similarity $\mathrm{ABA}^{(\mathrm{p})}$ framework, which gives the structure and computation of accepted (analogical) arguments.

6.1.1 The Development of Concept Similarity Measure under Preference Profile in Description Logics

Concept similarity measure can be regarded as a generalization of the classical reasoning problem of equivalence in description logics. That is, any two concepts are equivalent if and only if their similarity degree is one (*cf.* Equation 3.2). Regarding this observation, we have investigated an approach to compute the degree of subsumption between concepts in sub-Boolean description logics since their subsumption reasoning problems are tractable and they are shown to be expressive enough for formulating realistic ontologies. As a result, we developed a formal definition for calculating the degree of subsumption between $\mathcal{ELH}$ concepts (*cf.* Chapter 3). This subsumption degree function also gives us an approach to produce a numerical degree indicating concept similarity. For instance, the measure sim was defined as the average of the two corresponding subsumption degrees of $\mathcal{ELH}$ concepts. When two concepts are not in equivalence relationship, sim is capable of providing the degree of relation w.r.t. their common and different features. Accordingly, they play a major role in the discovery of similar concepts in an ontology and are often used by many application areas such as ontology alignment algorithms. It is worth mentioning that there are other concept similarity measure we have developed for $\mathcal{FL}_0$ concepts but they are not included in the thesis (*cf.* (Racharak & Suntisrivaraporn, 2015; Racharak & Tojo, 2018)).

An experiment in (Bernstein et al., 2005) reported that similarity measure might depend on target applications and should be personalized to the agent's similarity judgment style. We did consider this point and extended the definition of concept similarity measure in such a way that the degree of concept similarity is calculated w.r.t. subjective factors (such as the agent's preferences). This generalized notion is called *concept similarity measure under preference profile*. In particular, we developed a formalism for expressing the agent's preferences in concept similarity called *preference profile* and further refined sim according to each aspect of preference profile to sim^π for the DL $\mathcal{ELH}$.

Apart from the definition of concept similarity measure under preference profile, Chapter 4 also identified a set of desirable properties that any concrete measures of this notion should satisfy. We have provided proofs of satisfied properties for the developed concrete measures. Understanding their satisfied properties is important for employing the measures in any applicable areas since their users can predict the expected behaviors. The measures can also be used regardless of the agent's preferences *i.e.* sim^π is tuned with the special preference profile called the default preference profile π_0. Finally, we have provided proofs that sim^π can be computed in polynomial time.

6.1.2 The Design of Algorithmic Procedures for sim^π and Their Empirical Evaluation w.r.t. Realistic Ontologies

In Section 4.4, two concrete algorithms *viz.* the top-down approach and the bottom-up approach for implementations of sim^{π} were developed. The computational complexity of each algorithm was clearly analyzed. Concretely, both algorithms make the similar number of executions; however, the bottom-up additionally requires an amount of extra space due to the employed dynamic programming technique. Unlike the top-down approach, the bottom-up approach has never recursively invoked itself to determine the similarity of different pair of nested concepts. The algorithm directly uses values stored in the table. Both approaches have different benefits and drawbacks. On the one hand, the bottom-up requires an additional extra space. On the other hand, it does work productively an environment where recursion is fairly expensive.

In Section 4.5, our defined notion $\sim^{\pi}$ has been evaluated with realistic ontologies w.r.t. several use cases. In this thesis, we used simπ to show the practical performance of both developed algorithms and usefulness of tuning the measure via preference profile. Both algorithms of sim^{π} were implemented using Java version 1.8 with the usage of Spring Boot version 1.3.3.RELEASE as application programming interfaces (APIs). These APIs can also be used by application developers to use simπ with their working ontologies. Results of the empirical evaluation are summarized as follows:

1. We compared the practical performance of the top-down sim^{π} and the bottom-up sim^{π} w.r.t. the medical ontology SNOMED CT. The experiment showed that the bottom-up sim^{π} performs approximately three times faster than the top-down sim^{π}. This result conforms to our theoretical analysis as discussed earlier;
2. We re-implemented the existing measure sim based on the same technologies and techniques as sim^{π}. Then, we compared the practical performance of sim^{π} and sim w.r.t. SNOMED CT and found that they perform equally;
3. We evaluated the backward compatibility of sim^{π} with sim. This experiment would like to ensure that the default preference profile can be used when preferences are not given by the agent. Our experiment has guaranteed this;
4. We showed the usefulness of our defined notion through measuring the similarity of SNOMED CT concepts. Due to its special characteristics, measuring similarity of SNOMED CT concepts requires special ways of tuning

the measure. We showed that tuning sim^{π} under the special setting yields the more intuitive results. We also compared[25] the use of $simi$, which is another measure for the same $\mathcal{ELH}$, and found that lacking (even some) aspects of preference profile may not be suitable to use with an ontology where some special cases of tuning are required; and

5. We also showed the usefulness of our defined notion in several use cases of query answering systems with realistic ontologies. The discussion showed that $\sim^{\pi}$ is appropriate to identify the degree of similarity w.r.t. the agent's preferences.

Though we designed and developed the notion of concept similarity measure under relevant factors for construction and evaluation of analogical arguments, this notion also has great potential use in knowledge engineering such as the development of recommendation systems based on the agent's preferences, the development of domain-specific knowledge bases, and the ontology engineering. Moreover, it may be used with heterogeneous ontologies by identifying duplicated primitive concepts and primitive roles among ontologies via $\mathfrak{s}^{c}$ and $\mathfrak{s}^{r}$ (*cf.* Section 6.2). In the next subsection, we discuss how analogical arguments can be constructed and evaluated for persuasive reasoning, which is the main purpose of our developed similarity measure.

6.1.3 Construction and Evaluation of Analogical Arguments

We recall that analogical reasoning is a complex process based on a comparison between two pairs of concepts or states of affairs (*aka.* the *source* and the *target*) for characterizing certain features from one to another. Arguments which employ this process to support their claims are called *analogical arguments*. Chapter 5 investigated and explored the structure and the computation for their defeasibility in light of the argumentation theory. As a result, we introduced *assumption-based argumentation with predicate similarity* $\mathrm{ABA}^{(p)}$ framework, which can be seen as an extension of assumption-based argumentation framework (ABA), in which not only assumptions can be used but also similarity of predicates are used to support a claim.

[25] We also compared sim^{π} with measures in $\mathcal{ELH}$ and $\mathcal{FL}_0$ (*cf.* Table 4.4).

ABA$^{(p)}$ consists of four mainstreams *viz.* an ABA framework, a terminology, a concept similarity under preferences, and a preference context. When no assumptions are available to construct an argument tree, additional assumptions can be constructed from the use of a similarity measure w.r.t. a terminology and a preference context. In particular, ABA$^{(p)}$ exploits benefits of two different formalisms, *viz.* inference rules and terminological sentences, for dealing with analogical argumentation.

ABA$^{(p)}$ was designed to be a general framework for analogical argumentation. Thus, other notions apart from an ABA framework are also remained in general. For instance, ones may express a terminology as inference rules in $\mathcal{T}$ underlying a language $\mathcal{L}_T$ and $\sim_\tau^{\mathrm{p}}$ may be defined as a proportion of common features to different features as discussed in Chapter 5. As an exemplification, we have discussed how ones can use a particular description logic to express terminological formulae and our recent developed measure sim^π is also demonstrated. One benefit of using description logics is that their expressivity and computational complexities were clearly studied (Baader et al., 2007).

Like ABA, all semantic notions for determining the acceptability of arguments in AA also apply to arguments in ABA$^{(p)}$. Thus, we investigate a constructive proof procedure for determining a grounded set of assumptions in this work. Since different agents may value analogies for their reasoning unequally, we also study how each choice of operator $\otimes$ can influence different types of agents in analogical reasoning. Concerning other semantic notions of acceptability, this becomes an obvious future work to investigate on a dispute derivation for them and to further study how each semantic notion contributes to analogical argumentation in practice. It is also worth noting that we also developed other two formalisms based on the argumentation scheme (*cf.* (Racharak et al., 2017a, 2017b)) but they are not included in the thesis. The first formalism exploited benefits of extensive tools from answer set programming together with our developed notions (and our developed APIs). This provides a rough-and-ready method for building an analogical inference engine. On the other hand, the second one developed an argument-based logic-programming-like language which provides the possibility of representing information in terms of strict, defeasible, and similarity rules in a declarative manner. Their differences were also discussed in Section 5.3.

6.2 Potential Applications in Service Science Area

Service science is a new discipline emerging from the rapid development of services across the industrial world. Its root is an interdisciplinary study of computer science, operations research, industrial engineering, mathematics, business strategy, management science, decision theory, and social and cognitive science, and legal science. Its goal is to improve essential nature of service, *i.e.* the joint co-creation between service providers and service consumers. It is worth noting that developing $\mathrm{ABA}^{(p)}$ causes sim^{π} to come out as a by-product. Thus, we would like to discuss some potential applications of them in service science area.

First, our developed similarity measure has potential applications of knowledge engineering. For instance, sim^{π} can be used in the development of recommendation systems based on the agent's preferences, the development of domain-specific knowledge bases, and the ontology engineering. We exemplify in Subsection 4.5.2 a development of recommendation systems based on the agent's preferences via the sections about tuning $\mathfrak{s}^{r}$, $\mathfrak{s}^{c}$, and $\mathfrak{i}^{r}$, and a development of the domain-specific knowledge base in case of SNOMED CT (*cf.* page 68).

Second, if service applications would like to exploit similar concepts across heterogeneous ontologies, one approach is to identify duplicated primitive concepts and primitive roles among ontologies via $\mathfrak{s}^{c}$ and $\mathfrak{s}^{r}$ of preference profile, respectively. In fact, some existing ontologies partially contain duplicated information. It has been revealed in (Dhombres & Bodenreider, 2016) that concepts used by different terminologies may unintentionally mean the same. In (Dhombres & Bodenreider, 2016), 30% of Human Phenotype Ontology (HPO) concepts are semantically duplicated with Snomed ct concepts. For example, the HPO concept 'Multicystic Dysplastic Kidney' (HP:0000003) is identical to the SNOMED CT concept 'Multicystic Renal Dysplasia' (SCTID:204962002). In such a case, a mapping between these two ontologies should be formed. After the mapping, similar concepts from multiple ontologies can be found out.

Last, $\mathrm{ABA}^{(p)}$ can be employed to suggest the reasoning by analogy. This in fact corresponds to the two different phases of a decision making situation *viz.* the reasoning phase and the applicability phase. Suppose this framework is employed in a court case, then the framework can give suggestion that seems to be appropriate with a target situation (*e.g.* a case) and the expert (*e.g.* the law people) may consider the applicability after our suggestion – this is at their disposal.

6.3 Directions of Future Research

Several directions for further research on analogical argumentation and similarity measure of concepts are in order:

- In light of argumentation schemes, (Macagno, Walton, & Tindale, 2017) developed some inferential structures and defeasibility conditions for analogical arguments. Thus, we aim at investigating if such inferential structures can be captured by $ABA^{(p)}$;
- Preference-based argumentation framework (PAF) extended abstract argumentation by equipping with $\preccurlyeq$ to define a 'defeat' relation among each attack. This gives us an obvious future direction to investigate theoretical relationship between $ABA^{(p)}$ and PAF frameworks. Understanding this will allow us to transfer some proven properties of PAF to $ABA^{(p)}$;
- Apart from theoretical research directions, we also intend to apply our proposed framework in some practical domains where analogical reasoning is extensively used *e.g* in clinical practices. In clinical domain, many terminologies do exist and are represented in description logics *e.g.* SNOMED CT and Go. The remaining tasks will be then encoding the actual methods of medical experts in terms of inference rules;
- The proposed similarity measure is not meant to be the universal measure. Indeed, it is restricted to the DL $\mathcal{ELH}$ with unfoldable TBox. While it came with the limitation in terms of expressivity, its computation was proven to be tractable; thereby, provided practically acceptable response time which is a key requirement in the design and the development of large-scale ontologies. As for future work, we are interested in exploring other techniques of concept similarity measure under preference profile for more expressive DLs and other formalisms corresponding to $ABA^{(p)}$ framework;

- The current structure of preference profile also restricts its expressivity on sub-Boolean logics, particularly $\mathcal{ELH}$. Hence, it appears to be a natural step to extend preference profile to support more expressive DLs *e.g.* concept negation, and also, to support capabilities to express preferences on an ABox;
- As reported in (Bernstein et al., 2005) about the need of having multiple measures, we are interested to investigate the possible classes of similarity measures w.r.t. their potential use cases and applications. Understanding this would help the agent to select the right measure for a dealing situation;
- The proposed approach of concept similarity measure under preference profile has an advantage of computing the degree of commonalities under the agent's preferences. On the other hand, it cannot provide a good reason why two concepts are considered as 'being similar'. As for future work, we are interested in extracting the computational content which makes two concepts considered as being similar. This will give more informative answer to analogical arguments since the supporting reasons of a claim are more explainable. To do this, we may investigate the deduction systems *e.g.* a sequent calculus and a natural deduction system as developed in (Rademaker, 2012);
- Apart from our developed Java APIs, we intend to extend our development as a plug-in of ontology editors such as Protégé. Doing this would undoubtedly spread out their usability to a wider group of users; and
- The current usage of preference profile appears only in the task of concept similarity measure, which is a TBox-related problem. Now, we are interested in exploring ways to adopt preference profile on ABox-related problem *e.g.* non-standard instance checking under preference profile. The idea in the nutshell is to use concept similarity measure under preference profile for ABox instance checking rather than using the standard instance checking techniques. This may also involve extending the structure of preference profile with some capabilities of defining preferences over each instance in the ABox.

REFERENCES

Aleven, V. (1997). *Teaching Case-based Argumentation through a Model and Examples*. University of Pittsburgh, Pittsburgh, Pennsylvania: PhD diss.

Amgoud, L., & Cayrol, C. (2002). Inferring from Inconsistency in Preference-Based Argumentation Frameworks. *Journal of Automated Reasoning*, *29*(2), 125–169. https://doi.org/10.1023/A:1021603608656

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., … Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, *25*(1), 25–29. https://doi.org/10.1038/75556

Ashley, K. (2006). Case-based reasoning. In *Information Technology and Lawyers: Advanced Technology in the Legal Domain, from Challenges to Daily Routine* (pp. 23–60). Berlin: Springer.

Baader, F. (2003). Terminological cycles in a description logic with existential restrictions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (pp. 325–330). Morgan Kaufmann Publishers Inc.

Baader, F., Brandt, S., & Küsters, R. (2001). Matching under side conditions in description logics. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 1* (pp. 213–218). Morgan Kaufmann Publishers Inc.

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the EL envelope. In *IJCAI* (Vol. 5, pp. 364–369).

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2007). *The description logic handbook: Theory, implementation, and applications*. New York, NY, USA: Cambridge University Press.

Baader, F., Horrocks, I., & Sattler, U. (2004). Description logics. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 3–28). Berlin, Heidelberg: Springer Berlin Heidelberg.

Baader, F., Küsters, R., & Molitor, R. (1999). Computing least common subsumers in description logics with existential restrictions. In *IJCAI* (Vol. 99, pp. 96–101).

Baader, F., & Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica*, *69*(1), 5–40.

Baroni, P., & Giacomin, M. (2009). Semantics of abstract argument systems. In G. Simari & I. Rahwan (Eds.), *Argumentation in Artificial Intelligence* (pp. 25–44). Boston, MA: Springer US.

Bartha, P. (2010). By parallel reasoning: The construction and evaluation of analogical arguments. *By Parallel Reasoning: The Construction and Evaluation of Analogical Arguments*, 1–384.

Bermejo-Luque, L. (2014). The uses of analogies. In *Systematic Approaches to Argument by Analogy* (pp. 57–71). Springer.

Bernstein, A., Kaufmann, E., Bürki, C., & Klein, M. (2005). How similar is it? Towards personalized similarity measures in ontologies. In *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety* (pp. 1347–1366). Heidelberg: Physica-Verlag HD.

Bondarenko, A., Dung, P. M., Kowalski, R. A., & Toni, F. (1997). An Abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, *93*(1–2), 63–101.

Bondarenko, A., Toni, F., & Kowalski, R. A. (1993). An assumption-based framework for non-monotonic reasoning. *Logic Programming and Non-Monotonic Reasoning (Lisbon, 1993)*, 171–189.

Borgida, A., & Walsh, T. J. (n.d.). Towards Measuring Similarity in Description Logics.

Brachman, R. J., & Levesque, H. J. (1984). The tractability of subsumption in frame-based description languages. In *AAAI* (Vol. 84, pp. 34–37).

Brandt, S. (2004). Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and -- what else? In *Proceedings of the 16th European Conference on Artificial Intelligence* (pp. 298–302).

Calvanese, D., De Giacomo, G., Lenzerini, M., & Nardi, D. (2001). Reasoning in expressive description logics. *Handbook of Automated Reasoning*, *2*, 1581–1634.

Copi, I. M., Cohen, C., & McMahon, K. (2016). *Introduction to logic*. Routledge.

d'Amato, C., Fanizzi, N., & Esposito, F. (2009). A Semantic Similarity Measure for Expressive Description Logics, 13. Artificial Intelligence; Logic in Computer Science. Retrieved from http://arxiv.org/abs/0911.5043

D'Amato, C., Fanizzi, N., & Esposito, F. (2006). A dissimilarity measure for ALC concept descriptions. *In Proceedings of the 2006 ACM Symposium on Applied Computing*, 1695–1699.

D'Amato, C., Staab, S., & Fanizzi, N. (2008). On the influence of description logics ontologies on conceptual similarity. *In Proceedings of Knowledge Engineering: Practice and Patterns*, 48–63.

Davies, T. R. (1988). Determination, uniformity, and relevance: Normative criteria for generalization and reasoning by analogy. In *Analogical reasoning* (pp. 227–250). Springer.

De Giacomo, G., & Lenzerini, M. (1996). TBox and ABox reasoning in expressive description logics. *KR*, *96*(316–327), 10.

Dhombres, F., & Bodenreider, O. (2016). Interoperability between phenotypes in research and healthcare terminologies?Investigating partial mappings between HPO and SNOMED CT. *Journal of Biomedical Semantics*, *7*(1), 3.

Donini, F. M., lenzerini, M., Nardi, D., & Schaerf, A. (1996). Principles of Knowledge Representation. In G. Brewka (Ed.) (pp. 191–236). Stanford, CA, USA: Center for the Study of Language and Information.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*(2), 321–357.

Dung, P. M., Kowalski, R. A., & Toni, F. (2009). Assumption-based argumentation. In G. Simari & I. Rahwan (Eds.), *Argumentation in Artificial Intelligence* (pp. 199–218). Boston, MA: Springer US.

Dung, P. M., Mancarella, P., & Toni, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence*, *171*(10), 642–674. https://doi.org/https://doi.org/10.1016/j.artint.2007.05.003

Dung, P. M., & Thang, P. M. (2008). Towards an argument-based model of legal doctrines in common law of contracts. *Proc. CLIMA IX*, *7*.

Fanizzi, N., & D'Amato, C. (n.d.). A similarity measure for the ALN description logic. *In Proceedings of CILC 2006 - Italian Conference on Computational Logic*, 26–27.

Gabbay, D. M., Hogger, C. J., & Robinson, J. A. (1998). *Handbook of Logic in*

Artificial Intelligence and Logic Programming: Volume 5: Logic Programming. Clarendon Press.

García, A. J., & Simari, G. R. (2004). Defeasible Logic Programming: An Argumentative Approach. *Journal of Theory and Practice of Logic Programming*, *4*(2), 95–138.

Garssen, B. (2009). Comparing the incomparable: Figurative analogies in a dialectical testing procedure. In *Pondering on problems of argumentation* (pp. 133–140). Springer.

Gelfond, M., & Lifschitz, V. (1988). The stable model semantics for logic programming. In *ICLP/SLP* (Vol. 88, pp. 1070–1080).

Glucksberg, S., & Keysar, B. (1990). Understanding metaphorical comparisons: Beyond similarity. *Psychological Review*, *97*(1), 3.

Goebel, R. (1989a). A sketch of analogy as reasoning with equality hypotheses. In K. P. Jantke (Ed.), *Analogical and Inductive Inference: International Workshop AII '89 Reinhardsbrunn Castle, GDR, October 1--6, 1989 Proceedings* (pp. 243–253). Berlin, Heidelberg: Springer Berlin Heidelberg.

Goebel, R. (1989b). A Sketch of Analogy as Reasoning with Equality Hypotheses. *Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1-6, 1989, Proceedings*, *397*, 243–253.

Gover, T. (2018). Problems in argument analysis and evaluation.

Guarini, M., Butchart, A., Smith, P. S., & Moldovan, A. (2009). Resources for research on analogy: A multi-disciplinary guide. *Informal Logic*, *29*.

Haraguchi, M., & Arikawa, S. (1987). Reasoning by analogy as a partial identity between models. In K. P. Jantke (Ed.), *Analogical and Inductive Inference* (pp. 61–87). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hastings, A. C. (1963). A Reformulation of the Modes of Reasoning in Argumentation.

Hesse, M. B. (1965). Models and analogies in science.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpeset des jura. *Bulletin de La Societe Vaudoise Des Sciences Naturellese*, *37*, 547–579.

Janowicz, K. (2006). Sim-DL: towards a semantic similarity measurement theory for the description logic ALCNR in geographic information retrieval. In *SeBGIS 2006, OTM Workshops 2006. Volume 4278 of Lecture Notes in Computer Science*. Springer.

Janowicz, K., & Wilkes, M. (2009). SIM-DL_A: A Novel Semantic Similarity Measure for Description Logics Reducing Inter-concept to Inter-instance Similarity. *In Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, 353–367.

Kakas, K. C., & Toni, F. (1999). Computing argumentation in logic programming. *Journal of Logic and Computation*, *9*(4), 515–562.

Kohlas, J. (2003). Probabilistic argumentation systems A new way to combine logic with probability. *Journal of Applied Logic*, *1*(3–4), 225–253. https://doi.org/10.1016/S1570-8683(03)00014-4

Lehmann, K., & Turhan, A.-Y. (2012). A framework for semantic-based similarity measures for ELH-concepts. In L. F. del Cerro, A. Herzig, & J. Mengin (Eds.), *JELIA* (Vol. 7519, pp. 307–319). Springer.

Macagno, F. (2014). Analogy and redefinition. In *Systematic approaches to argument by analogy* (pp. 73–89). Springer.

Macagno, F., Walton, D., & Tindale, C. (2017). Analogical arguments: Inferential structures and defeasibility conditions. *Argumentation*, *31*(2), 221–243.

Matt, P.-A., Toni, F., Stournaras, T., & Dimitrelos, D. (2008). Argumentation-based agents for eprocurement. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track* (pp. 71–74).

Modgil, S., & Prakken, H. (2014). The $ASPIC^+$ Framework for Structured Argumentation: A Tutorial. *Argument and Computation*, *5*(1), 31–62.

Nebel, B. (1990). Terminological reasoning is inherently intractable. *Artificial Intelligence*, *43*(2), 235–249.

Nebel, B. (1991). Terminological cycles: semantics and computational properties.

O'Neil, M., Payne, C., & Read, J. (1995). Read Codes Version 3: a user led terminology. *Methods of Information in Medicine*, *34*(1–2), 187–192.

Patel-Schneider, P. F., & Swartout, B. (1993). Description-logic knowledge representation system specification from the KRSS group of the ARPA knowledge sharing effort. *KRSS Group of the ARPA*.

Perelman, C. (1969). *The New Rhetoric: A Treatise on Argumentation [by] C. Perelman and L. Olbrechts-Tyteca. Translated by John Wilkinson and Purcell Weaver*.

Pollock, J. L. (1992). How to reason defeasibly. *Artificial Intelligence*, *57*(1), 1–42.

Poole, D. (1988). A logical framework for default reasoning. *Artificial Intelligence*, *36*(1), 27–47.

Racharak, T. (2018). *Concept Similarity and Agent's Preferences in Description Logics: Computations and Applications*. School of Information Science, Japan Advanced Institute of Science and Technology. Retrieved from http://hdl.handle.net/10119/15429

Racharak, T., & Suntisrivaraporn, B. (2015). Similarity measures for FL_0 concept descriptions from an automata-theoretic point of view. *In Proceedings of the 6th Annual International Conference on Information and Communication Technology for Embedded Systems (ICICTES 2015)*.

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2016a). sim^{π}: A Concept Similarity Measure under an Agent's Preferences in Description Logic ELH. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence* (pp. 480–487). https://doi.org/10.5220/0005813404800487

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2016b). *Identifying an agent's preferences toward similarity measures in description logics*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9544). https://doi.org/10.1007/978-3-319-31676-5_14

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2018). Personalizing a Concept Similarity Measure in the Description Logic ELH with Preference Profile. *Computing and Informatics*, *37*(3), 581–613.

Racharak, T., & Tojo, S. (2017a). Analogical reasoning in clinical practice with description logic ELH. In *Agents and Artificial Intelligence - 9th International Conference, ICAART 2017, Porto, Portugal, February 24-26, 2017, Revised Selected Papers* (pp. 179–204). Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-93581-2_10

Racharak, T., & Tojo, S. (2017b). Tuning agent's profile for similarity measure in description logic ELH. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence, ICAART 2017, Volume 2, Porto, Portugal, February 24-26, 2017.* (pp. 287–298). https://doi.org/10.5220/0006249602870298

Racharak, T., & Tojo, S. (2018). Inherited Properties of FL_0 concept similarity measure under preference profile. In *Agents and Artificial Intelligence - 9th International Conference, ICAART 2018, Porto, Portugal, February 24-26, 2017, Revised Selected Papers*. Springer International Publishing.

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2016). Argument-Based Logic Programming for Analogical Reasoning. In *New Frontiers in Artificial Intelligence - JSAI-isAI 2016 Workshops* (pp. 253–269).

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2017a). *Argument-based logic programming for analogical reasoning*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10247 LNAI). https://doi.org/10.1007/978-3-319-61572-1_17

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2017b). *Combining answer set programming with description logics for analogical reasoning under an agent's preferences*. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10351 LNCS). https://doi.org/10.1007/978-3-319-60045-1_33

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2019). On construction and evaluation of analogical arguments for persuasive reasoning. *Applied Artificial Intelligence*, *33*. https://doi.org/https://doi.org/10.1080/08839514.2019.1646026

Rademaker, A. (2012). *A Proof Theory for Description Logics*. Springer Science & Business Media.

Raha, S., Hossain, A., & Ghosh, S. (2008). Similarity based approximate reasoning: Fuzzy control. *Journal of Applied Logic*, *6*(1), 47–71.

Rector, A. (2003). Medical informatics. In *The description logic handbook* (pp. 406–426).

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, *13*(1–2), 81–132.

Schild, K. (1991). A correspondence theory for terminological logics: preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1* (pp. 466–471). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Schulz, S., Suntisrivaraporn, B., & Baader, F. (2007). SNOMED CT's problem list: ontologists' and logicians' therapy suggestions. *Studies in Health Technology and Informatics*, *129*(1), 802.

Smolka, M. S.-S. G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, *48*(1), 1–26.

Spackman, K. A. (2005). Rates of change in a large clinical terminology: three years experience with SNOMED Clinical Terms. In *AMIA*.

Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). SNOMED RT: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium* (p. 640).

Spackman, K. A., Dionne, R., Mays, E., & Weis, J. (2002). Role grouping as an extension to the description logic of Ontylog, motivated by concept modeling in {\sc Snomed}. In *Proceedings of the AMIA Symposium* (p. 712).

Stearns, M. Q., Price, C., Spackman, K. A., & Wang, A. Y. (2001). SNOMED clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium* (pp. 662–666). College of American Pathologists, Northfield, IL, USA. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2243297/

Sun, R. (1995a). Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, *75*(2), 241–295. https://doi.org/10.1016/0004-3702(94)00028-Y

Sun, R. (1995b). Robust reasoning: Integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, *75*(2), 241–295.

Suntisrivaraporn, B. (2013). A similarity measure for the description logic EL with unfoldable terminologies. *In INCoS*, 408–413.

Tongphu, S., & Suntisrivaraporn, B. (2015). Algorithms for measuring similarity between ELH concept descriptions: a case study on SNOMED CT. *Journal of Computing and Informatics (Accepted on May 7; to Appear)*.

Toni, F. (2007). Assumption-based argumentation for selection and composition of services. In *International Workshop on Computational Logic in Multi-Agent Systems* (pp. 231–247).

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327.

Van Eemeren, F. H., & Garssen, B. (2014). Argumentation by analogy in stereotypical argumentative patterns. In *Systematic approaches to argument by analogy* (pp. 41–56). Springer.

Van Gelder, A., Ross, K. A., & Schlipf, J. S. (1991). The well-founded semantics for general logic programs. *Journal of the ACM (JACM)*, *38*(3), 619–649.

Van Harmelen, F., Lifschitz, V., & Porter, B. (2008). *Handbook of knowledge representation* (Vol. 1). Elsevier.

Waller, B. N. (2001). Classifying and analyzing analogies. *Informal Logic*, *21*(3).

Walton, D. (2010). Similarity, precedent and argument from analogy. *Artificial Intelligence and Law*, *18*(3), 217–246.

Walton, D., Reed, C., & Macagno, F. (2008). *Argumentation schemes*. Cambridge University Press.

Weinreb, L. L. (2016). *Legal reason: The use of analogy in legal argument*. Cambridge University Press.

APPENDICES

APPENDIX A
THE SYSTEMATIZED NOMENCLATURE OF MEDICINE: SNOMED CT

The Systematized Nomenclature of Medicine, Clinical Terms (*aka.* SNOMED CT)[26] is one of the largest and the most widely used medical ontologies currently available. Figure A.1 depicts its web interface which can be accessed via the link given at the footnote. It was produced by merging SNOMED Reference Terminology (RT) (Rector, 2003; Spackman, Campbell, & Côté, 1997) with Clinical Terms version 3 (CTV3) (O'Neil, Payne, & Read, 1995).

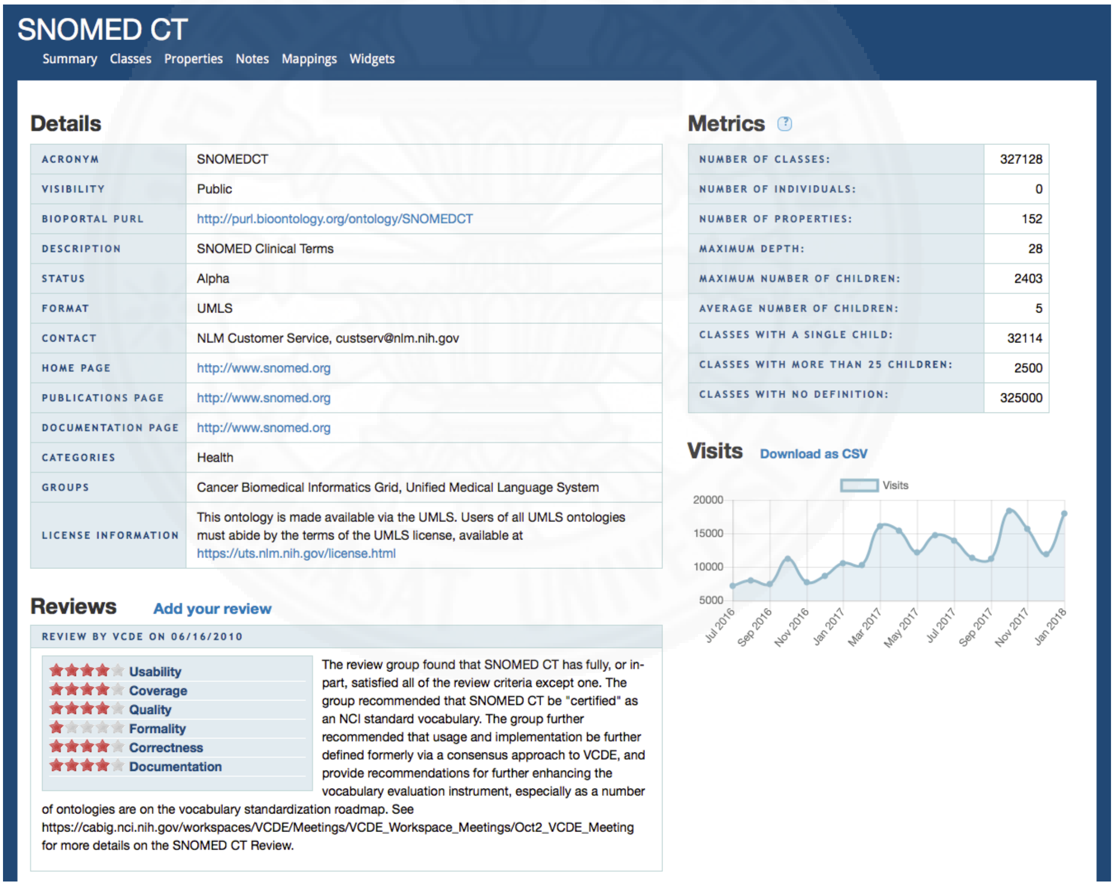

Figure A.1 SNOMED CT BioPortal (Accessed on February 21, 2018).

Historically, SNOMED RT was developed by the College of American Pathologists (CAP) with the aim to be a comprehensive clinical reference terminology

---
[26] http://bioportal.bioontology.org/ontologies/SNOMEDCT

e.g. the retrieval and analysis of data relating the causes of diseases, the treatment of patients, and retrieval of health care information (Spackman et al., 1997). The RT version was the first generation of the SNOMED terminology to use the formal semantics through the KRSS syntax (Patel-Schneider & Swartout, 1993).

In 1993, the UK National Health Service (NHS) has adopted the Read codes, which had been developed by a medical practitioner Read, for health electronic records. Later on, the terminology has been expanded and enhanced to become Clinical Terms version 3 (CTV3).

Between 1999 and 2002, CAP and the UK NHS together with Keiser Permanente have jointly worked to merge SNOMED RT and CTV3. Its resulting SNOMED CT contained 55% of the source concepts from CTV3 and 31% from RT. Moreover, the ontology become freely available in both the US and UK.

Nowadays, SNOMED CT is already used by more than 50 countries. Furthermore, it is the most comprehensive, multilingual clinical health-care terminology in the world and is mapped to other international standards. As reported in (Spackman, 2005; Stearns et al., 2001), SNOMED CT can be seen as the DL $\mathcal{ELH}$ with an unfoldable TBox. SNOMED CT has several inherent characteristics. We discuss several of them in the following.

Firstly, SNOMED CT purposefully uses the special role $\mathsf{roleGroup}$ to group two or more existential quantifications in a definition. Spackman et al. has illustrated the use of $\mathsf{roleGroup}$ for the concept ‘Tetralogy of Fallot’ in (Spackman, Dionne, Mays, & Weis, 2002) as follows.

$$
\begin{aligned}
&\mathsf{TetralogyOfFallot}\\
&\qquad\equiv \exists \mathsf{rG}.\,(\exists \mathsf{s.RightVentricle} \sqcap \exists \mathsf{m.Hypertrophy})\\
&\qquad\sqcap \exists \mathsf{rG}.\,(\exists \mathsf{s.Aorta} \sqcap \exists \mathsf{m.Overriding})\\
&\qquad\sqcap \exists \mathsf{rG}.\,(\exists \mathsf{s.Pulmonary} \sqcap \exists \mathsf{m.Stenosis})\\
&\qquad\sqcap \exists \mathsf{rG}.\,(\exists \mathsf{s.InterventricularSeptum} \sqcap \exists \mathsf{m.IncompleteClosure})
\end{aligned}
$$

where rG, s, and m are abbreviations for roles $\mathsf{roleGroup}$, site, and $\mathsf{morphology}$, respectively.

Secondly, individuals (*i.e.* the ABox) are omitted. On the other hand, SNOMED concepts such as $\mathsf{Germany}$, Japan, and $\mathsf{Thailand}$ are used to represent unique individuals. Indeed, they are seen as ‘instances’ of the concept $\mathsf{GeographicLocation}$.

Thirdly, SNOMED CT has 18 mutually exclusive top-level concepts for dividing the entire ontology into disjoint categories. However, the disjointness is not logically specified as axioms; hence, some concept names may happen to belong to more than one category.

Lastly, the SNOMED CT top concept SCT-Top subsumes every defined concept of each category. This means this special concept is shared by every expanded concept.

In this thesis, we use SNOMED CT ontology version from January 2005 which contains 13 role inclusions, 38,719 concept definitions, 340,972 primitive concept definitions, 379,691 concept names and 62 role names.

APPENDIX B
IMPLEMENTATION OF THE MEASURE sim^{π}

We have implemented sim^{π} as a collection of application programming interfaces (APIs) as well as command-line interfaces (CLIs) using Java version 1.8 with the usage of Spring Boot version 1.3.3.RELEASE. The ultimate goal of these APIs is to provide a tool for identifying the degree of concept similarity under preference profile for the DL $\mathcal{ELH}$. As shown in Theorem 4.2, the computation of sim^{π} can be performed in polynomial time.

Since sim^{π} is targeted on $\mathcal{ELH}$, we summarize the provided constructors as follows:

- top concept '$\top$',
- conjunction '$C \sqcap D$', and
- full existential quantification '$\exists r.C$'; and

the following means of expressivity to construct an ontology as follows:

- primitive concept definition '$A \sqsubseteq D$',
- concept definition '$A \equiv D$', and
- role hierarchy axiom '$r \sqsubseteq s$'.

Currently, sim^{π} accepts two formats of inputs *viz.* in KRSS[27] (Knowledge Representation System Specification) (Patel-Schneider & Swartout, 1993), OWL (Web Ontology Language), and OWL 2[28]. Our APIs wrap OWL API[29] version 3.4.4. In the following, we have summarized shortly both KRSS and OWL syntaxes only the parts relevant to our APIs for self-containment of the thesis.

In KRSS, an ontology contains the following sorts of statements:

- primitive concept definition '(define-primitive-concept CN C)',
- concept definition '(define-concept CN C)', and
- role hierarchy axiom '(define-primitive-role RN_1 RN_2)',

[27] http://dl.kr.org/krss-spec.ps
[28] https://www.w3.org/TR/owl2-overview/
[29] http://semanticweb.org/wiki/OWL_API.html

where CN be a concept name, RN_1 and RN_2 be two different role names, and concept C can be either CN, ⊤, or formed as follows:

- conjunction '(and C_1 ... C_n)',
- full existential quantification '(some RN C)',

where concept $C, C_1, \ldots, C_n$ are recursively defined as above.

Figure B.1 depicts an overview of OWL 2. In the center, the ellipse represents the abstract notion of an ontology, which can be thought of as an abstract ontology structure or an RDF graph. The top of the figure shows each concrete syntax based on the abstract notion which can be serialized and exchanged. The bottom shows the two specification of semantics defining the meaning of an ontology. As aforementioned, our APIs wrap the OWL API, which can handle these various syntaxes and semantics. Thus, this capability automatically transfer to our APIs for free. We refer the readers to check the official documentation for the full descriptions of each syntax and semantics.

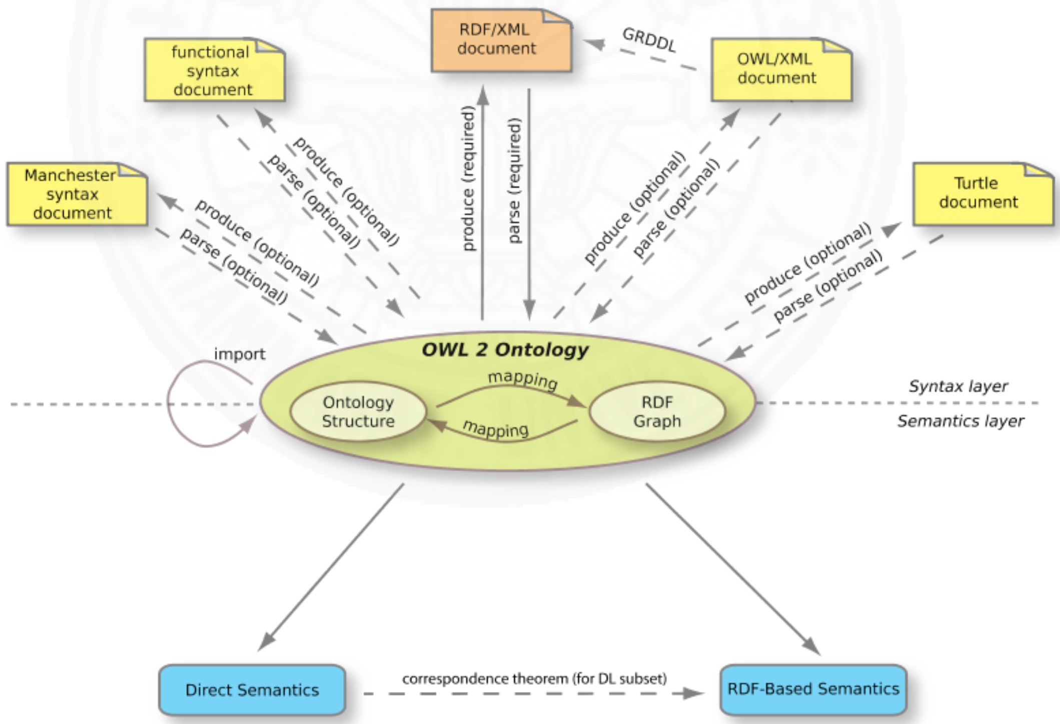

Figure B.1 The Structure of OWL 2 (Source: https://www.w3.org/TR/owl2-overview/OWL2-structure2-800.png).

To use our APIs in Java, four classes may be involved *viz.* 'KRSSServiceContext', 'OWLServiceContext', 'PreferenceProfile', and 'SimilarityService'. First, KRSSService- Context and OWLServiceContext are used to initialize the ontology from a given file path. Initializing the ontology is mandatory and is required to do once prior to the query of concept similarity. Second, PreferenceProfile is used to configure each aspect of preference profile (*cf.* Section 4.1). If this class is not explicitly used, it will automatically use the default value (*cf.* the default preference profile). Third, SimilarityService encapsulates functionalities to compute the degree of concept similarity based on a syntax and a computational approach (*cf.* Section 4.4) as follows:

- measureOWLConceptsWithTopDownSimPi(conceptName1 : String, conceptName2 : String) : BigDecimal;
- measureOWLConceptsWithDynamicProgrammingSimPi(conceptName1 : String, conceptName2 : String) : BigDecimal;
- measureKRSSConceptsWithTopDownSimPi(conceptName1 : String, conceptName2 : String) : BigDecimal; and
- measureKRSSConceptsWithDynamicProgrammingSimPi(conceptName1 : String, conceptName2 : String) : BigDecimal.

Figure B.2 demonstrates how ones can use our APIs in Java.

```
public class Example {

    ... // Initialize logger, owlServiceContext and similarityService

    private void run(String ... args) {
        String owlFilepath = StringUtils.trimWhitespace(args[0]);
        String conceptName1 = StringUtils.trimWhitespace(args[1]);
        String conceptName2 = StringUtils.trimWhitespace(args[2]);

        owlServiceContext.init(owlFilepath);
        BigDecimal value = similarityService.measureOWL
                              ConceptsWithTopDownSimPi(conceptName1, conceptName2);

        logger.info ("Done! The similarity between " + conceptName1 + " and "
                              + conceptName2 + " is " + value.toPlainString() + " %.");
    }
}
```

Figure B.2 Example of Using sim$^{\pi}$ APIs in Java.

We have also implemented several batch programs based on the APIs and used them on the part of our empirical evaluation of the thesis (*cf.* Section 4.5). For the current implementation, each program stores each concept pair in question as a text file separated by a space. Each aspect of preference profile is stored on its own file but is collectively kept together in the same folder. Their outputs after the execution is stored in another text file. Figure B.3 depicts the idea as described above. We also implemented other batch programs based on the same techniques for the measure sim (as discussed in Section 4.5) with the purpose of benchmarking. In total, we have implemented 8 programs. Each uses the same structure as shown in the figure.

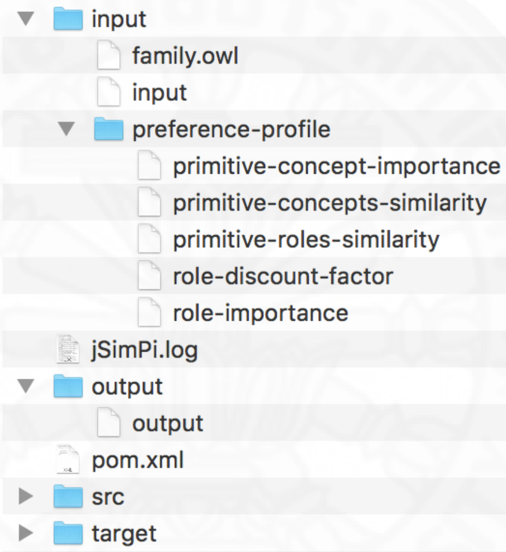

Figure B.3 Our Batch Program's Structure.

To run each batch program, we have to execute the command 'mvn spring-boot:run'. When the program is run, it will take each concept pair defined in a given ontology (such as 'family.owl' in this case), compute the degree of similarity under a defined preference profile, and pipe the results to output file. Figure B.4 illustrates an example after the execution. The figure shows that the degree of similarity between both concepts is 0.96.

```
Son     SonInLaw      0.96000
```

Figure B.4 The Degree of Similarity between Son and SonInLaw.

Finally, we have written 111 unit test cases to ensure that all batch programs and the core APIs function correctly. These test cases were written to cover important parts of the implementation. Concepts in both the family ontology (family.owl) and Snomed ct were used by the test cases. To execute the test, we use the command 'mvn test'. Figure B.5 depicts the results.

```
Results :

Tests run: 110, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------
[INFO] Total time: 49.587 s
[INFO] Finished at: 2018-02-14T22:55:32+09:00
[INFO] Final Memory: 16M/226M
[INFO] ------------------------------------------------
```

Figure B.5 Results of Unit Tests.

BIOGRAPHY

Name	Mr. Teeradaj Racharak
Date of Birth	September 18, 1988
Education	2010: Bachelor of Engineering (Software and Knowledge Engineering) Kasetsart University 2012: Master of Engineering (Computer Science) Asian Institute of Technology 2018: Doctor of Philosophy (Information Science) Japan Advanced Institute of Science and Technology

Publications

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2019). On construction and evaluation of analogical arguments for persuasive reasoning. *Applied Artificial Intelligence*, 33.

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2018). Personalizing a Concept Similarity Measure in the Description Logic $\mathcal{ELH}$ with Preference Profile. *Computing and Informatics*, *37*(3), 581–613.

Racharak, T., & Tojo, S. (2018). Inherited Properties of $\mathcal{FL}_0$ concept similarity measure under preference profile. In *Agents and Artificial Intelligence - 9th International Conference, ICAART 2018, Porto, Portugal, February 24-26, 2017, Revised Selected Papers*. Springer International Publishing.

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2017a). Argument-based logic programming for analogical reasoning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10247 LNAI).

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2017b). Combining answer set programming with description logics for analogical reasoning under an agent's preferences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10351 LNCS).

Racharak, T., & Tojo, S. (2017a). Analogical reasoning in clinical practice with description logic $\mathcal{ELH}$. In *Agents and Artificial Intelligence - 9th International Conference, ICAART 2017, Porto, Portugal, February 24-26, 2017, Revised Selected Papers* (pp. 179–204). Springer International Publishing.

Racharak, T., & Tojo, S. (2017b). Tuning agent's profile for similarity measure in description logic $\mathcal{ELH}$. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence, ICAART 2017, Volume 2, Porto, Portugal, February 24-26, 2017.* (pp. 287–298).

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2016a). sim$^{\pi}$: A Concept Similarity Measure under an Agent's Preferences in Description Logic $\mathcal{ELH}$. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence* (pp. 480–487).

Racharak, T., Suntisrivaraporn, B., & Tojo, S. (2016b). Identifying an agent's preferences toward similarity measures in description logics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9544).

Racharak, T., Tojo, S., Hung, N. D., & Boonkwan, P. (2016). Argument-Based Logic Programming for Analogical Reasoning. In *New Frontiers in Artificial Intelligence - JSAI-isAI 2016 Workshops* (pp. 253–269).

Racharak, T., & Suntisrivaraporn, B. (2015). Similarity measures for $\mathcal{FL}_0$ concept descriptions from an automata-theoretic point of view. *In Proceedings of the 6th Annual International Conference on Information and Communication Technology for Embedded Systems (ICICTES 2015)*.