Comenius University
Faculty of Mathematics, Physics and Informatics
Department of Applied Informatics

Semantic Investigations in Distributed Ontologies

PhD. dissertation

Martin Homola
author

Ján Šefránek & Luciano Serafini
advisors

Bratislava April 2010

Semantic Investigations in Distributed Ontologies

PhD. dissertation by Martin Homola

Comenius University

Semantic Investigations in Distributed Ontologies

PhD. dissertation by

Martin Homola

under supervision of

Ján Šefránek

and additional advisory by

Luciano Serafini

Comenius University

Semantic Investigations in Distributed Ontologies

PhD. dissertation

study programme	11-80-9 theoretical computer science
author	RNDr. Martin Homola Comenius University Faculty of Mathematics, Physics and Informatics Department of Applied Informatics
advisor	doc. PhDr. Ján Šefránek, CSc. Comenius University Faculty of Mathematics, Physics and Informatics Department of Applied Informatics
special advisor	Dr. Luciano Serafini Fondazione Bruno Kessler Via Sommarive 18, 381 23 Trento, Italy
submited	April 2010

Typeset in LaTeX 2ε

Comenius University
Faculty of Mathematics, Physics and Informatics
Department of Applied Informatics
Mlynská dolina, 842 48 Bratislava, Slovakia

Under heaven all can be seen as beauty, even the ugliness; all can be seen as good, even the evil.

Therefore, being and not being arise from each other; difficult and easy create one another; long and short contrast one another; high and low lean upon each other; sound and voice harmonize each other; before and after follow one another.

And therefore, the true man acts by not doing anything; teaches without words, without commenting upon the ten thousand things. Lives without possessing; acts without directing; and successfully concludes without being attached. And the one who is not attached to anything, does not miss anything!

Laozi (circa 6th century BCE)

Acknowledgement

My deepest thanks I owe to Michaela, for her care and also for her patience, which I both needed in order to conclude this thesis. I want to thank to my family, for their support, and to all my friends, but especially those who kept encouraging me to finish and submit this thesis.

I am also grateful and indebted to my two exceptional advisors Ján Šefránek and Luciano Serafini who have given me guidance and have always served as good example to me. In addition I want to thank the organisers and the lecturers of the ICCL Summer School 2005, which was influential for this thesis.

This work was supported from projects no. APVV-20-P04805 and APVV-99-P05005 of Slovak Agency for Promotion of Research and Development, from projects VEGA no. 1/0173/03, 1/3112/06 and 1/0668/10 of Slovak Ministry of Education and Slovak Academy of Sciences, from projects GUK no. 314/2005 and 365/2008 awarded by Comenius University.

Abstracts

Abstract in English

Distributed and modular ontology representation is a young and promising research field with exciting motivation, ranging from the idea of facilitating the ontology development process by introduction of modularity, up to devision of a distributed ontology representation layer for the envisaged Semantic Web. Several distributed ontological knowledge representation frameworks have been introduced. Among them, especially Distributed Description Logics (DDL) have captured our attention. DDL offer a natural extension of Description Logics, and allow to connect multiple ontologies by means of directional semantic mapping. The DDL framework enables knowledge reuse and combines well with basic intuitions behind the Semantic Web.

Formal ontologies represent concepts and relations between them, the most essential of these relations being concept subsumption. In DDL, subsumption propagates from one ontology to another thanks to semantic mapping expressed between them. Subsumption propagation is an essential feature of DDL, and it has been studied in the literature. In this line of research lies also this thesis, in which we offer a comprehensive study of the phenomenon of subsumption propagation within DDL knowledge bases.

At first, our attention is caught by complex concepts. If mapping is expressed between more elementary concepts and subsumption propagates on these concepts thanks to the mapping, one would expect that subsumption would appear also between certain complex concepts constructed on top of them. This is not always true in DDL, as pointed out in the literature. We propose a strengthened version of the semantics that does not suffer from this problem.

Another interesting issue is the problem of subsumption propagation between remote ontologies in DDL. Remote ontologies are not connected directly, but only by mapping paths spanning through multiple ontology units. We show, that subsumption propagation between remote ontologies is minute under the original DDL semantics. Aiming to cope with this issue, we propose and study three different strengthenings of the semantics. For the one of these novel semantics, which we find the most viable, we also provide a distributed tableaux reasoning algorithm.

Finally, in its last part, this thesis presents a comparison of DDL with three other distributed and modular ontology representation frameworks ($\mathcal{E}$-connections, P-DL and IDDL). Although, some comparisons of these approaches have already been published, they are usually mostly qualitative, disregarding

some notable exceptions. In this thesis we perform a formal comparison of the expressive power, and we present two new reductions between these frameworks.

Abstrakt v Slovenčine (Abstract in Slovak)

Vývoj distribuovaných a modulárnych ontologických reprezentačných formalizmov je pomerne mladou oblasťou vedeckého bádania. Motiváciou je snaha o uľahčenie procesu tvorby ontológií, ale rovnako dobre tiež potreba skutočne distribuovanej reprezentácie pre aplikácie ako je napríklad sémantický web. V literatúre nájdeme hneď niekoľko distribuovaných ontologických formalizmov. Medzi najzaujímavejšie patrí DDL (Distributed Description Logics). Ide o nadstavbu nad deskripčnými logikami, umožňujúcu prepojiť viacero ontológií za pomoci orientovaného sémantického mapovania medzi prvkami týchto ontológií. DDL umožňuje znovuvyužitie existujúcich ontológií pri tvorbe nových a je v súlade so základnými intuíciami, a predpokladmi sémantického webu.

Formálne ontológie reprezentujú koncepty a vzťahy medzi nimi, a to predovšetkým subsumciu. V DDL pozorujeme prenos subsumcie medzi ontológiami, ktorý je dôsledkom sémantického mapovania medzi týmito ontológiami. Prenos subsumcie, rovnajúci sa prenosu a znovuvyužitiu znalostí, je dôležitou črtou DDL, a bol už v literatúre študovaný. Prenosom subsumcie v DDL sa zaoberá aj táto dizertačná práca, prinášajúca ucelenú štúdiu tohto fenoménu.

V práci sa najprv zameriavame na problém prenosu subsumcie na zložené koncepty v prípade, že mapovanie existuje medzi konceptmi, z ktorých sú tieto zložené koncepty skonštruované. V DDL však k prenosu subsumcie v takýchto prípadoch nedochádza vždy v dostatočnej miere, ako sa môžeme dočítať v literatúre. V tejto práci navrhujeme dve zosilnené sémantiky pre DDL, ktoré pomáhajú tento problém odstrániť.

Ďalším zaujímavým problémom je prenos subsumcie medzi vzdialenými ontológiami. Ide o ontológie, medzi ktorými neexistuje priame mapovanie, ale predsa sú spojené nepriamo, mapovaním prechádzajúcim cez niekoľko ďalších ontológií v báze znalostí. V tejto práci ukazujeme, že v takýchto prípadoch dochádza v DDL k prenosu subsumcie len vo veľmi obmedzenej miere. Ako možné riešenia študujeme tri zosilnené sémantiky pre DDL, ktoré čiastočne pomáhajú tento problém prekonať. Pre jednu z týchto sémantík navrhujeme tiež distribuovaný tablový algoritmus, ktorý umožňuje usudzovanie s touto sémantikou.

Posledná časť práce sa zameriava na porovnanie DDL s troma ďalšími formalizmami pre distribuované ontologie ($\mathcal{E}$-connections, P-DL, IDDL). Diskutujeme existujúce výsledky týkajúce sa porovnania ich výrazovej sily a prinášame dve nové redukcie medzi týmito formalizmami, a to medzi P-DL a DDL, ako aj medzi IDDL a DDL.

Keywords

knowledge representation, distributed and modular ontologies, subsumption, semantics, description logics, tableaux reasoning algorithm

Contents

Chapter 1

Introduction

1.1 Ontologies in Context of the Semantic Web

When the vision of the Semantic Web was formulated by Berners-Lee et al. (2001) it has given great attention to ontologies. The Semantic Web of future would contain information with well defined meaning that would enable computers to work with this information in unforeseen ways and ontologies would provide reference points for this meaning. In the current Web of 2001, and in most cases still in the current Web of today, when one computer agent publishes data on the Web and another computer agent discovers the data, it is not immediately apparent how to interpret and process this data if this has not been previously hardcoded into the program of the second agent. However, if the agent that publishes the data provides an annotation based on some ontology, and the other agent uses the very same ontology, the interpretation of the data becomes clear even without previous hardcoding of this interpretation within the agent and without any human interaction.

As an example consider a situation in which a transport company publishes a timetable of bus no. 39 on its Web page (we will call this Web page the *publisher*). This timetable includes a record indicating that: "Bus 39 takes you from the bus stop Main Station to the bus stop University at 9:45 am and the journey takes 15 minutes." While this kind of information is immediately understandable and useful for a human reader, if the page is discovered by another computer agent (we will call this agent the *spider*) that belongs to another company which offers an integrated searchable database of local transportation on their own Web page, this agent does not immediately know how to interpret this data. If the author of this computer program has not preprogrammed this, it has no way of knowing that "Bus 39" is the same as its own "mean of transport," that "bus stop Main Station" is an identifier of an "entry point" and that the words "journey takes" on the publisher's Web page mean the same as "time of travel" in its own program. At this point an ontology comes onto the scene. Imagine there would exist a third Web page (we will call it the *ontology*) which contains a precise and unique definition for each of these notions. All of the "transport line," "transport line station," "arrival time," and "departure time," etc. are defined here. Now the programmer reprograms the spider and instructs it to search the Web for pages annotated by the ontology. Whenever it finds

a four-tuple of values annotated with "transport line," "transport line station," "arrival time," and "departure time" it downloads this data and stores it in its own database. Now also when the publisher publishes a new update of the timetables, its developers annotate the data with references to the ontology. Consequently the spider is now able to autonomously process the data on the publisher's Web page and to store it in its database and make it available for its users.

The philosophical study of *ontology* originated in the work of Aristotle (circa 40BCE) and it started as the study of the things around us and their names. In contemporary philosophy, ontology has become a study of entities that exist or can be assumed to exist, their types and relations. It was philosophy where computer science, and particularly knowledge representation (KR) borrowed this term (Sowa 1999). Within KR, an ontology is a specific kind of knowledge base that is concerned with terminological data. As concisely defined by Gruber (1993): "An ontology is an explicit specification of a conceptualization." That is to say: an ontology is an explicitly stated representation which encodes what are the concepts of a specific domain of discourse and what are the relations between these concepts. This domain of discourse is usually rather narrow. Typically, ontologies deal with three kinds of notions:

individuals: also called objects, are particular things the ontology talks about;

concepts: also classes, are types of objects. An object possibly belongs to one or more concepts. In such a case we say it is an instance of each of them;

roles: also properties, are binary relations on the objects. A pair of objects possibly participates in one ore more of the roles.

An important notion is that of *subsumption*, a subconcept–superconcept relation between the concepts within each ontology. This relation induces a hierarchical ordering on the concepts which is called *taxonomy* and sometimes also the classification of an ontology or subsumption hierarchy. All the other relations are usually treated equally, as roles, however in certain cases also the subpart–superpart relation is given special attention. The hierarchy induced by this relation is called *partonomy*.

A common graphical representation of an ontology is a diagram in which concepts are depicted as nodes which are connected with lines (edges) representing the taxonomy. Usually superconcepts are placed above their subconcepts. Such diagrams are very convenient and easy to read. For an example see Fig. 1.1 (a), where an excerpt of a classification of animal species is depicted. A partonomy may also be depicted in a similar fashion, as we show in Fig. 1.1 (b) which depicts a partonomy of a passenger car. We see on the figure that both ontologies have a tree structure. This is often but not always the case. Especially in the taxonomy, where one concept possibly has two or more superconcepts. We also observe from these examples that in both cases the domain of discourse is quite specific, in the first case it is animal species, and it the latter case it is vehicles and their compounds.

The true power of ontologies is associated with the logical foundations of KR. A number of logics have been developed for manipulation with terminological data. These logics are to be understood simply as formal languages for representation of terminological knowledge. The most prominent space in theoretical

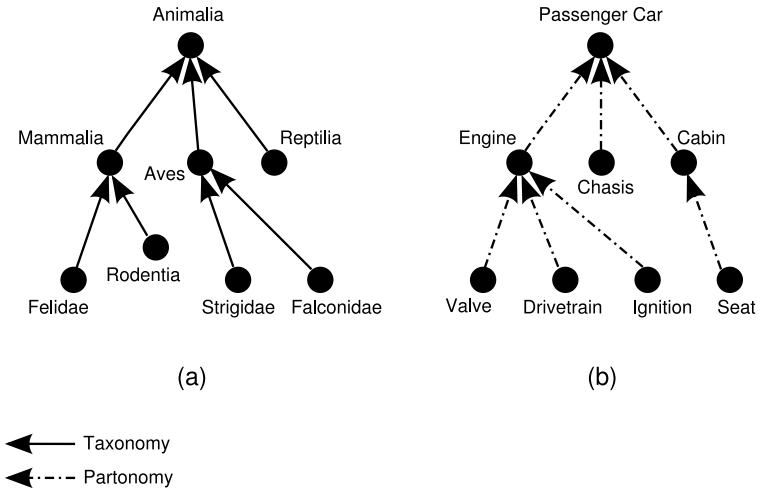

Figure 1.1: Two example ontologies (excerpts): a) classification of animal species; b) partonomy of a passenger car

investigation of logical foundations of ontologies is occupied by a family of terminological languages called Description Logics (DL) (Baader et al. 2003). There are also other languages, for instance the F-Logic (Kifer et al. 1995). There are also ontological languages introduced specifically for the Semantic Web: Resource Description Framework (RDF) (Beckett 2004, Hayes 2004, Brickley & Guha 2004) and OWL Web Ontology Language (OWL) (McGuinness & van Harmelen 2004, W3C OWL Working Group 2009). Each of these languages provides a unified syntax for representing ontological data. With this syntax we are able to record relations between concepts, and also to create complex definitions of concepts. Thus, more complex concepts are precisely defined on top of simpler concepts (e.g., we are able to define that PassengerCar is such a vehicle which has an engine, four wheels, at least two seats and is used to transport passengers, provided that Vehicle, Engine, hasPart, and all the other concepts and roles that we use in the definition are already defined).

Logical semantics of these languages allows to check whether the recorded ontology is consistent and to infer out implicit knowledge from the ontology. That is, when we write down some knowledge using the syntax of the language, some additional knowledge is possibly implied. For instance in Fig. 1.1 (a) it is explicitly recorded that Rodentia is a subconcept of Mamalia and in turn the latter is a subconcept of Animalia, in this case it is implied that Rodentia is also a subconcept of Animalia because the subsumption relation is transitive, as required by the semantics. Such implications are precisely defined by the semantics and are always the same, given a fixed collection of explicitly recorded knowledge. The problem of entailment (i.e., of determining whether some data is implied from a given knowledge base) has been studied as a decision problem for these languages and algorithms that decide entailment have been developed. These algorithms have been implemented into software tools called *reasoners*. Some well known ontological reasoners include FaCT++ (Tsarkov & Horrocks 2006), RACER (Haarslev & Müller 2001) and Pellet (Sirin et al. 2007).

All these properties and the factual availability of reasoning tools have made the logical ontology languages attractive and useful whenever a need of explicit conceptualization of some domain arises. Loosely speaking, ontologies are good for fixing the meaning of terms, especially when these terms are to be used by several parties in communication and data exchange. This pattern has been employed in human-to-human, human-to-computer and computer-to-computer interaction. Successful application domains include database schema integration, software engineering, configuration (design of complex systems created by combining multiple components), natural language processing, and medicinal informatics (see for instance the surveys by Baader et al. (2003) and by Staab & Studer (2004)). In these application areas large-scale ontologies have been developed, comprising of hundreds of thousands of concepts, such as for instance GALEN (Rector et al. 1993) and SNOMED (Spackman et al. 1999).

One of the application areas of ontologies, as we have learned, is also the envisaged Semantic Web. Construction of a large number of suitable ontologies that would cover all of the domains of interest can possibly provide references for meaning of terms to such an extent, that data publishers may easily use these ontologies for annotations, and with help of these annotations intelligent software agents may use of the annotated data to an unforeseen extent. However, the essence of the Web lies in its distributed and decentralized nature. In fact, it is by large this distributed nature, that has enabled the Web to grow to its current extent. Therefore, if the Semantic Web is to be an extension of the current Web, if it is to blend with the current Web and if it shall keep the potential to grow to size proportional to the size of the current Web, also the ontological data contained therein must be distributed and decentralized. And therefore, in many cases it will also be incomplete, duplicated, and even contradicting, because if information is decentralized, it is unavoidable. Thus, one of the challenges we face in order to make the Semantic Web possible is to extend the current ontology languages and reasoners to be able to cope with such distributed scenarios. The contribution of this thesis lies within this line of research.

1.2 Modular and Distributed Ontology Frameworks

The past decade has seen a rise of the need for modular and distributed ontology representation techniques and tools. Dealing with ever larger ontologies, maintained by whole teams of ontology engineers and domain experts, have increasingly called for introduction of modularity into the ontology engineering process. This development is largely inspired by modular software engineering, where software is organized into modular packages which are possibly reused and combined in novel ways and thus the software engineering process is facilitated and simplified, and quality and production are increased. Such a modular ontology representation framework would bring in the possibility of construction of encapsulated ontology modules, which would then serve as reusable building blocks for larger ontologies. The two basic research questions associated with this issue are:

- what are the formal requirements for an encapsulated and reusable ontology module;
- under which requirements and how could two or more ontology modules be combined.

A somewhat different but closely related idea is the one of reusing bits of knowledge which are distributed, stored in multiple locations and possibly governed by autonomous agents with reasoning capabilities. This idea even predates the Semantic Web vision (Subrahmanian 1994, Fagin et al. 1995, Ghidini & Serafini 1998, Giunchiglia & Ghidini 1998). On the other hand, it is particularly the Semantic Web vision, what adds to the relevance and topicality of this idea. There are two important research goals which differentiate this approach from the previous one:

- the aim to combine ontologies even if they possibly include duplicated, contradictory and locally inconsistent knowledge;
- the interest in distributed reasoning scenarios where the reasoning task is distributed between multiple autonomous local reasoners.

The first goal is justified by the assumption that in the Semantic Web there will be numerous ontologies available, many of them partially overlapping and sometimes contradictory. Already in the current Web of today we find a great number of various ontologies, independently published by autonomous sources. The aim is to find ways how to reason with all this knowledge and to reuse it in newly constructed ontologies whenever desired and possible. The second goal is evident: if the knowledge is governed by multiple autonomous reasoners, as envisaged in the Semantic Web, distributing the reasoning task among them may possibly yield improved reasoning efficiency. In addition, it may also be the case that some of the autonomous agents would not allow us to copy all the knowledge it governs in order to perform the reasoning centrally, instead it could allow us to query its knowledge base by some interface. In such a case, distributed reasoning is a must, not an option.

Most of the formalisms that fall under modular and distributed ontology frameworks make use of DL as the local language on top of which they consecutively build. The common aspect of all of these formalisms is that they deal with the problem how to combine multiple ontology units into a larger ontology. There is, however, some difference in what means the formalisms offer for this operation. In this sense we distinguish two main approaches:

- ontology units are combined simply by taking the union of the logical theories that represent each unit;
- ontology units are not directly united, instead they are connected using novel syntactic constructs introduced specifically for this purpose.

In the first approach the combination is only possible if certain conditions are met. These conditions may be understood as ontology modularization requirements. Two of such conditions investigated in the literature are known as *locality of knowledge* (Cuenca Grau, Horrocks, Kutz & Sattler 2006, Cuenca Grau et al. 2007) and *conservative extension* (Ghilardi et al. 2006, Lutz et al.

2007). However this approach lies behind the scope of this work as the rest of this thesis concentrates exclusively on the second approach.

In the second line of research a number of works has been published. All of these approaches constitute a semantic extension of DL in the sense that they introduce novel syntactic constructs on top of DL in order to relate elements across the ontology units that are being combined. In this sense there are three main lines of research:

ontology linking: inside an ontology, two instances are possibly coupled with a role. The idea of ontology linking extends this, and enables individuals from two distinct ontology units to be coupled with *links*. See an example in Fig. 1.2, where we have two ontologies, in the first one the instances are companies, and in the second one they are products. The link produces connects companies and products. A strict requirement of this approach is that the domains of the ontology units that are being combined with links are disjoint. As in our example, where companies and products are strictly separated. As much as normal roles, links allow for complex concepts to be constructed, and they basically act as cross-ontology roles. This approach is followed by $\mathcal{E}$-connections, a framework originally introduced by Kutz et al. (2002);

ontology mapping: conversely to the previous approach, ontology mapping has been introduced in order to relate ontologies on the same domain or on partially overlapping domains. Elements from different ontologies are related by special *mapping* constructs that indicate how these elements are semantically related. The mapping possibly states that two concepts or two roles from different ontologies are equal, that one is more general or more specific than the other, or that they are disjoint. Also two individuals from different ontologies may be possibly related as equal or as distinct. As an example consider Fig. 1.3. Mapping is possibly expressed between two or more ontologies which model the same domain but each from a different perspective, and it is possibly motivated by the resolution of semantic heterogeneity between ontologies. Two logical frameworks that allow to connect multiple ontologies via semantic mapping are Distributed Description Logics, originally introduced by Borgida & Serafini (2002) and Integrated Distributed Description Logics, originally introduced by Zimmermann (2007);

ontology import: this approach aims to provide a flexible mechanism for partial ontology reuse. This is to be achieved by *importing* a distinguished subset of the entities (namely concepts, relations and individuals) defined in one ontology unit into another ontology unit where they are then reused. The semantics of importing takes care of propagating also the semantics relations that already exist between these entities in their home ontology unit into the ontology unit that imports them. For instance, if Sedan is a subconcept of PassengerCar in some ontology module $\mathcal{O}_1$, that deals with classification of automobiles, and these two concepts are both imported into some other ontology module $\mathcal{O}_2$, this subsumption relation is entailed also within $\mathcal{O}_2$. For another example have a look at Fig. 1.4, where roles are being imported. Semantic foundations of this approach have been

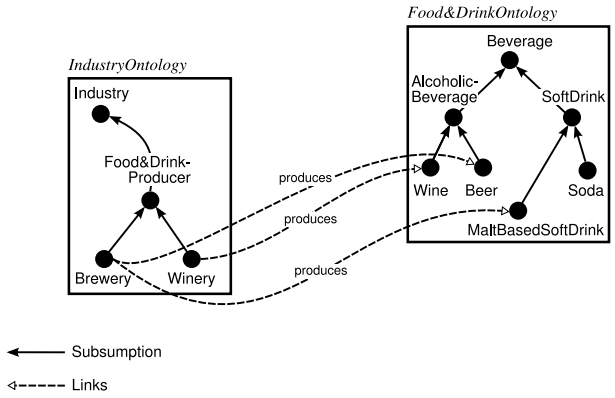

Figure 1.2: Example ontology linking scenario: using the link produces it is possible to relate concepts from different ontologies

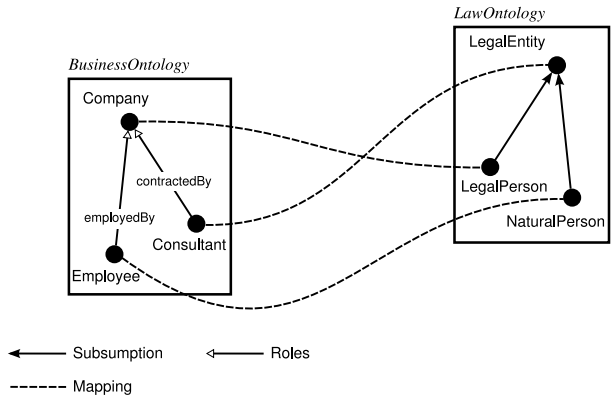

Figure 1.3: Example ontology mapping scenario: related concepts from the business ontology and law ontology are mapped between

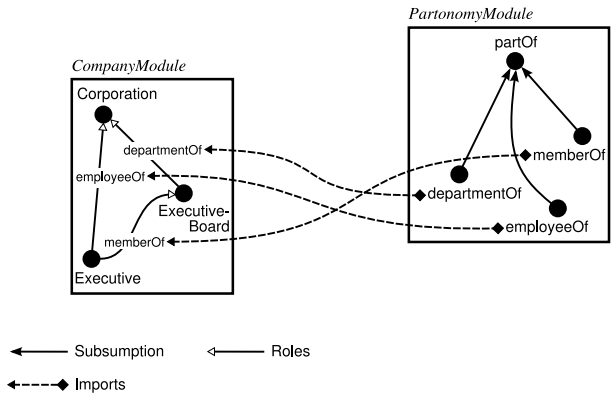

Figure 1.4: Example ontology importing scenario: different types of subpart roles are imported from a specific partonomy module

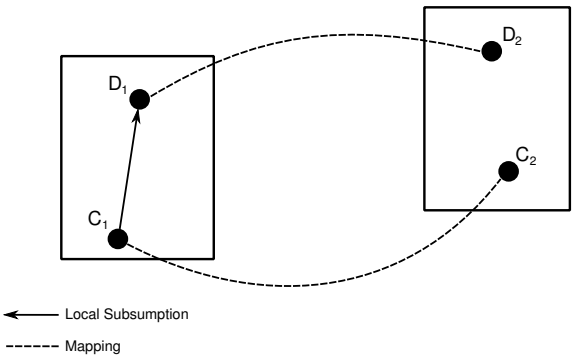

Figure 1.5: Example of subsumption propagation

laid out by Pan et al. (2006). A framework of Package-based Description Logics, first introduced by Bao & Honavar (2004), follows this approach.

Also related to modular and distributed ontologies is the problem of *ontology partitioning*. This is motivated by the problem of dealing with large and very complex ontologies by decomposing them into modules. An example is when a large ontology O is partitioned in a set of ontology units $O_1, \ldots, O_n$ such that the combination of $O_1, \ldots, O_n$ expressed by some framework is equivalent to the initial ontology O. Ontology partitioning has been investigated in association with $\mathcal{E}$-connections (Cuenca Grau et al. 2005, Cuenca Grau, Parsia, Sirin & Kalyanpur 2006). This approach is different from the previous three in that it is more concerned with identifying parts of ontologies which are sufficiently unrelated, in contrast from the previous three focusing on describing the relation between the components.

1.3 Contribution of This Work

As we have outlined, theoretical research in modular and distributed ontologies is motivated by such goals as easing the ontology engineering process, improving the maintainability of large ontologies, facilitating knowledge reuse, and providing a distributed ontology layer for the envisaged Semantic Web. However, there are still many fundamental issues to be addressed in order to achieve all these goals. Or as for instance Kalfoglou & Schorlemmer (2005) put it: "... ontology mapping nowadays faces some of the challenges we were facing ten years ago when the ontology field was at its infancy. We still do not understand completely the issues involved."

The main concern of this thesis is with Distributed Description Logics (DDL). DDL follow the ontology mapping paradigm, and offer a particularly robust framework that is capable to express semantic mapping between ontologies. DDL deal with partially overlapping modeling domains and provide a semantics which is able to cope with accidental inconsistency which may appear within a distributed ontology modeling scenario. A fully distributed reasoning algorithm has been developed for DDL (Serafini & Tamilin 2004), and it has been implemented in the DRAGO reasoner (Serafini et al. 2005). All these reasons

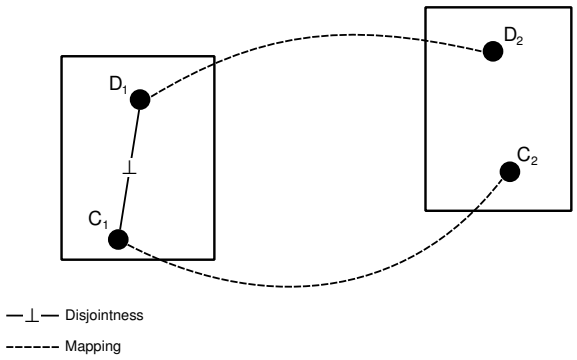

Figure 1.6: A more complex example in which relation between concepts (disjointness) is not propagated between ontology units

make DDL particularly well aligned with the Semantic Web vision, where partial duplication of knowledge between the ontologies has been described as an important feature, accidental inconsistency is unavoidable and it has to be dealt with, and distributed reasoning is one of the requirements.

One of the important properties of DDL is that the semantic mapping, once expressed between ontology entities, allows for propagation of semantic relations of these entities from one ontology to another. Most important relation that is propagated is the subsumption between concepts. Subsumption propagation in DDL has been a subject to multiple studies (Borgida & Serafini 2003, Serafini & Tamilin 2004). The most basic case of subsumption propagation is depicted in Fig. 1.5: given two ontology units $\mathcal{O}_1$, containing concepts C_1 and D_1 where C_1 is subsumed by D_1, and $\mathcal{O}_2$, containing concepts C_2 and D_2 – if C_1 is mapped to C_2 and D_1 is mapped to D_2, then subsumption between C_1 and D_1 is propagated and it is implied also between C_2 and D_2.[1] This property always holds if the scenario is represented using DDL.

However, in more complex modeling scenarios, the original DDL semantics may behave counterintuitively. One such a case has been described by Cuenca Grau, Parsia & Sirin (2006), see Fig. 1.6. Let us again have $\mathcal{O}_1$ with concepts C_1 and D_1, however, in this case C_1 and D_1 are disjoint. $\mathcal{O}_2$ contains C_2 and D_2. If C_1 is mapped to C_2 and D_1 is mapped to D_2, the disjointness between C_1 and D_1 does not propagate into $\mathcal{O}_2$, where C_2 and D_2 are possibly not disjoint. Note that concept disjointness and subsumption are closely related: C and D are disjoint if and only if C is subsumed by the complement of D.

Problematic cases, such as the one described above, call for a more thorough study of subsumption propagation in DDL. This thesis follows two main research issues, which are both associated with subsumption propagation:

- propagation of subsumption on more complex concepts in cases when the mapping is expressed between concepts from which these complex concepts are constructed;

[1] Note that we abstract from unnecessary technical details in Fig. 1.5 and 1.6. Later on we will revisit these examples formally and we will see how exactly they are represented in DDL (i.e., what particular type of mappings and their combination is required).

- propagation of subsumption between remote ontologies which are not directly mapped between but they are connected by mapping paths spanning through multiple ontology units.

In both of these lines of research, we have identified cases in which subsumption propagation exhibited by the original DDL semantics does not reach the extent that we would intuitively expect. In the first case it has been already known that subsumption propagates on complex concepts derived by means of the concept union constructor. Unfortunately, this is not true in case of the concept intersection constructor. In order to deal with this issue, we propose two possible ways how to strengthen the semantics in order to achieve subsumption propagation also in case of intersection, and we study their properties.

To our best knowledge, the second research issue, that is, propagation of subsumption between remote ontology units, has not been studied before in context of DDL. Our investigation suggests that the amount of subsumption propagation between remote ontology units is minute under the original DDL semantics. In order to improve this, we consider and study three possible extensions, each derived by strengthening the original semantics by additional semantic requirements. This step has been inspired by similar requirements that have been previously employed by Package-based Description Logics (Bao et al. 2006*b*, 2009). One of the strengthened semantics solves the subsumption propagation problem, but it exhibits undesirable side effects. The other two provide a reasonable trade-off between the level of subsumption propagation and the overall behaviour of the semantics. In addition, we provide a distributed reasoning algorithm for the most viable approach.

Later on, we shift our attention on the relation of the DDL framework and other relevant distributed and modular ontology frameworks. These include: $\mathcal{E}$-connections, Package-based Description Logics (P-DL) and Integrated Distributed Description Logics (IDDL). A common characteristic of these frameworks is that they all extend DL by new syntactic constructs which express relations between entities across the ontology units. Although each of the approaches follows different intuitions and serves for different modeling scenarios, there are notable similarities. At several occasions relations between these frameworks have been studied (Kutz et al. 2004, Bao et al. 2006*a*, Cuenca Grau & Kutz 2007, Wang et al. 2007) with aim to compare their expressivity. In this work we extend these results as follows:

- we prove that P-DL and DDL are interreducible, assuming a specific DDL semantics with restricted domain relations;

- we prove that IDDL is reducible into DDL, again relying on a specific restricted semantics of DDL, a different one as in the previous result.

These results contribute to the effort aiming to compare the expressivity of the formalisms that are available in distributed and modular ontologies, the novel syntactic constructs they employ, and their semantics. The results imply that these frameworks share certain similarities, however, as the reductions are not unconditional, it cannot be claimed that one framework is strictly more expressive than the others.

Most of our results have been already published. In the following, we list the publications that are related to this thesis:

- Homola (2007*b*). *Distributed description logics revisited.* Presented at the 20th International Workshop on Description Logics, Brixen-Bressanone, Italy;
- Homola (2008). *Subsumption propagation between remote ontologies in distributed description logic.* Presented at the 21st International Workshop on Description Logics, Dresden, Germany;
- Homola & Serafini (2008). *Towards distributed tableaux reasoning procedure for DDL with increased subsumption propagation between remote ontologies.* Presented at the KR Ontology Workshop, 11th International Conference on Principles of Knowledge Representation and Reasoning, Sydney, Australia;
- Serafini & Homola (2010). *Modular knowledge representation and reasoning in the Semantic Web.* A chapter in: 'Semantic Web Information Management: A Model-Based Perspective'. Springer;
- Homola & Serafini (2010*a*). *Augmenting subsumption propagation in distributed description logics.* Applied Artificial Intelligence, 24(1-2):137–174, Taylor & Francis;
- Homola & Serafini (2010*b*). *Towards formal comparison of ontology linking, mapping and importing.* Presented at the 23rd International Workshop on Description Logics, Waterloo, Canada;
- Frank, Dziuban & Homola (2010). *Sémantický web: niektoré aktuálne výzvy.* A chapter in: 'Umelá inteligencia a kognitívna veda II'. Vydavateľstvo STU.

In addition, this thesis' project and research results were presented at the following PhD symposia:

- International Students Conference on Applied Mathematics 2007 in Bratislava, Slovakia;
- Knowledge Web PhD Symposium, held in association with the 4th European Semantic Web Conference in Innsbruck, Austria, 2007, where it was mentored by Enrico Franconi;
- KR PhD Symposium, held in association with the 11th International Conference on Principles of Knowledge Representation and Reasoning in Sydney, Australia, 2008, where it was mentored by Franz Baader.

1.4 Roadmap

The thesis is structured as follows:

- Chap. 1 introduces the research field of distributed ontologies, and explains the motivation and contribution of this work.
- Chap. 2 provides a concise introduction to DL. The languages introduced in this chapter are later reused in order to build the distributed ontology frameworks that we study.

- Chap. 3 surveys the state of the art of DDL: its syntax and semantics, reasoning algorithm and known extensions.
- Chap. 4 concentrates on the problem of propagation of subsumption on complex concepts in DDL, in cases when mapping is expressed between concepts from which they are constructed.
- Chap. 5 studies subsumption propagation between remote ontologies expressed in DDL.
- Chap. 6 focuses on various different distributed ontology representation frameworks. It introduces $\mathcal{E}$-connections, IDDL and P-DL and presents a formal comparison of the expressive power and the semantics between DDL and these frameworks.
- Finally, the thesis summarized and concluded with Chap. 7.

Chapter 2

Description Logics

Description Logics (Baader et al. 2003) are powerful knowledge representation formalisms that are especially well suited for representing ontologies and ontological data, and for reasoning with such data.

A whole family of DL is known in the literature. Each of them offers different expressive power in trade-off for different effectivity of reasoning. DL belong to the decidable fragment of First Order Logic (FOL), and they provide a versatile representation framework for terminological knowledge centered around the notions of individuals, concepts and roles, which, in the respective order, correspond to individuals, unary and binary predicates in FOL. Complex concepts are built on top of atomic concepts using so called constructors, which, in FOL, correspond to logical connectives and quantifiers. The set of constructors is slightly different in each DL, and some more expressive DL allow also for complex roles being built using role constructors.

Important application areas of DL include database schema integration, medicinal informatics, and many others (Baader et al. 2003). In recent years, the importance of DL has grown especially due to their prominent role in ontology representation on the Semantic Web (Horrocks et al. 2003).

In this chapter, we survey several notable DL. The most basic of them is $\mathcal{ALC}$ (Schmidt-Schauß & Smolka 1991). On this logic, we will learn the basic DL notions: syntax, semantics, decision problems and reasoning algorithms. Gradually adding more and more expressive features, we will reach the DL $\mathcal{SHI}$ (Horrocks & Sattler 1999), $\mathcal{SHIQ}$ (Horrocks et al. 2000) and $\mathcal{SHOIQ}$ (Horrocks & Sattler 2005). These logics, especially $\mathcal{SHIQ}$ and $\mathcal{SHOIQ}$ will stand in center of our attention later on in this thesis, as most of the distributed ontology formalisms are build on top of them. In the end of this chapter, we also briefly discuss some more expressive DL, as they are also sometimes used by the distributed ontology formalisms. These include $\mathcal{SROIQ}$, on which the recently standardized OWL 2 (W3C OWL Working Group 2009) is based, and also $\mathcal{ALCQIb}$ (Tobies 2001).

2.1 $\mathcal{ALC}$, the basic DL

In this thesis, $\mathcal{ALC}$ will be the most basic and the most essential DL from which all the other languages are derived by extension. $\mathcal{ALC}$ has originated in the

work of Schmidt-Schauß & Smolka (1991), and its name comes from a simpler DL $\mathcal{AL}$ (Attribute Language). More precisely, $\mathcal{ALC}$ amounts to $\mathcal{AL}$ extended with the complement constructor, therefore the final $\mathcal{C}$. In addition $\mathcal{ALC}$ allows four other constructors: concept intersection and union, value restriction, and existential restriction.

It is worth to note at this point, that there are also simpler DL besides for $\mathcal{ALC}$ (Calvanese et al. 2005, Baader et al. 2006, Schmidt-Schauß & Smolka 1991). The main motivation for dealing with simpler languages is in lower computational complexity of reasoning. We will learn in this section, that already for $\mathcal{ALC}$, the worst-case complexity of reasoning is rather expensive (PSpace-complete). Our survey in this section is based on the work of Baader et al. (2003).

2.1.1 $\mathcal{ALC}$ Formulae

Expressions in DL are constructed from three kinds of symbols that come from an alphabet, sometimes called signature or vocabulary, which splits into three parts accordingly. The set of individual names, denoted by N_{I}, the set of atomic concept names N_{C}, and the set of atomic role names N_{R}.

Individuals represent particular objects. They will be denoted by small letters of alphabet $a, b, i, j, \ldots \in N_{\mathrm{I}}$. Concepts represent classes, or categories, types of objects. The individuals that are members of a concept are called instances of this concept. Concepts will be denoted by capital letters. We will use $A, B \in N_{\mathrm{C}}$ for atomic concepts, and $C, D, \ldots$ for more complex concepts, which we will meet below. Finally, roles represent relations between individuals. They will be denoted by capital letters, starting from $R, S, \ldots$. In $\mathcal{ALC}$ all roles will be atomic (i.e., members of N_{R}).

In more complex examples we will employ a different notation, in which these entities will be denoted by sans-serif font. Concepts start with capital letter: $\mathsf{Person}, \mathsf{Man}, \mathsf{Woman}, \ldots$; individuals and roles with small letter $\mathsf{johnSmith}, \mathsf{rickDecard}, \ldots$ (individuals), $\mathsf{friendOf}, \mathsf{partOf}, \ldots$ (roles).

Various DL typically differ in the set of concept and role constructors they offer. More constructors means greater expressive power but also worse efficiency of reasoning. As we already know, $\mathcal{ALC}$ only offers concept constructors. The following types of concepts may be constructed using these constructors.

Definition 1 ($\mathcal{ALC}$ concepts)**.** *The set of all $\mathcal{ALC}$ concepts over a signature N_{C}, N_{R} and N_{I} is recursively defined as follows:*

1. *any atomic concept $A \in N_{\mathrm{C}}$ is a concept;*
2. *concept complement $\neg C$ is concept;*
3. *intersection of concepts $C \sqcap D$ is a concept;*
4. *union of concepts $C \sqcup D$ is a concept;*
5. *value restriction $\forall R.C$ is a concept;*
6. *existential restriction $\exists R.C$ is a concept;*

where C and D are any concepts, and R is a role.

Complex concepts are also sometimes called concept descriptions. Intuitively speaking, the complement $\neg C$ stands for the set of individuals that are not of type C. The intersection of two concepts $C \sqcap D$ stands for the class of those individuals which are of type C and also of type D; while the union $C \sqcup D$ represents those individuals which are either of type C or D. The existential restriction $\exists R.C$ stands for a set of those individuals which are connected with at least one individual of type C by the role R. And finally, the value restriction $\forall C.R$ represents those individuals which are only exclusively associated to individuals of type C by the role R.

We also add some common syntactic sugar: let the *top concept* $\top$ be a syntactic shorthand for $A \sqcup \neg A$ and the *bottom concept* $\bot$ a syntactic shorthand for $A \sqcap \neg A$, with a newly introduced atomic concept A.

Our intuitive explanation of the meaning of complex concept is made explicit by the semantics of $\mathcal{ALC}$. As with all DL, and also with FOL, this semantics is model-theoretic.

Definition 2 (Semantics of $\mathcal{ALC}$ concepts)**.** *An* interpretation $\mathcal{I}$ *is pair* $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ *consisting of a* domain $\Delta^{\mathcal{I}}$*, which is a nonempty set, and of an* interpretation function $\cdot^{\mathcal{I}}$*, that assigns elements of* $\Delta^{\mathcal{I}}$ *to individuals, subsets of* $\Delta^{\mathcal{I}}$ *to concepts and subsets of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *to roles. Moreover, for any complex* $\mathcal{ALC}$ *concept of a form listed in Table 2.1 the corresponding condition holds.*

Concept	Condition
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{i \in \Delta^{\mathcal{I}} \mid (\forall j \in \Delta^{\mathcal{I}})\ \langle i,j \rangle \in R^{\mathcal{I}} \implies j \in C\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{i \in \Delta^{\mathcal{I}} \mid (\exists j \in \Delta^{\mathcal{I}})\ \langle i,j \rangle \in R^{\mathcal{I}} \wedge j \in C\}$

Table 2.1: Semantic constraints for $\mathcal{ALC}$ interpretations

The intention behind the top and the bottom concept is now clear, as it follows that in any interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ we have $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$.

In the graph-theoretic sense, each interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ is a directed graph with the set of vertices $\Delta^{\mathcal{I}}$ and the set of edges $\{\langle i,j \rangle \mid (\exists R)\ \langle i,j \rangle \in R^{\mathcal{I}}\}$. We commonly call j an R-successor[1] of i whenever $\langle i,j \rangle \in R^{\mathcal{I}}$.

Interpretations lead us directly to the notions of *concept satisfiability* and *subsumption* that constitute the two most basic decision problems in DL and hence reasoning tasks. Intuitively, a concept is satisfiable, if it has at least one instance. Otherwise, it is unsatisfiable. We care for satisfiable concepts – in a sense, unsatisfiable concepts are useless. A concept C is subsumed by a concept D, if D is more general than C. Other reasoning tasks for concepts include deciding *equivalence* and *disjointness*.

Definition 3 (Satisfiability of concepts)**.** *A concept* C *is satisfiable (or consistent) if there exists some interpretation* $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ *such that* $C^{\mathcal{I}} \neq \emptyset$.

[1] In the graph theory, in any directed graph, if there is a directed edge $\langle x, y \rangle$ between nodes x and y, we call y a successor of x and conversely x a predecessor of y.

Definition 4 (Subsumption of concepts)**.** *A concept C is subsumed by another concept D, denoted by $C \sqsubseteq D$, if for each interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.*

Definition 5 (Equivalence of concepts)**.** *Two concepts C and D are equivalent, denoted by $C \equiv D$, if for every interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ it holds that $C^{\mathcal{I}} = D^{\mathcal{I}}$.*

Definition 6 (Disjointness of concepts)**.** *Two concepts C and D are disjoint, if for every interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ it holds that $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.*

The two decision problems of subsumption and satisfiability are interreducible[2], hence, in practice, we only need to reason about one of them. Also the equivalence and disjointness problems are reducible into concept satisfiability. In the theory, but also in implementations, usually concept satisfiability is the reasoning task of choice and the remaining tasks are resolved thanks to the reductions.

Theorem 1. *A concept C is satisfiable if and only if the subsumption $C \sqsubseteq \bot$ does not hold.*

Theorem 2. *The subsumption relationship $C \sqsubseteq D$ holds if and only if $C \sqcap \neg D$ is unsatisfiable.*

Theorem 3. *Concepts C and D are equivalent if and only if both subsumptions $C \sqsubseteq D$ and $D \sqsubseteq C$ hold (i.e., if and only if both $C \sqcap \neg D$ and $D \sqcap \neg C$ are unsatisfiable).*

Theorem 4. *Two concepts C and D are disjoint if and only if the concept $C \sqcap D$ is unsatisfiable.*

In the following illustrative example, as well as in several other examples in this chapter, we take inspiration from the Wine ontology, which is used as an example ontology in the specification of OWL 1 (McGuinness & van Harmelen 2004).

Example 1. *Using the vocabulary with atomic concept names $N_{\mathrm{C}} = \{\mathsf{Wine}, \mathsf{Red}, \mathsf{BeaujolaisRegion}\}$ and role names $N_{\mathrm{R}} = \{\mathsf{hasColor}, \mathsf{producedIn}\}$, one may describe a certain kind of wine with the following concept:*

$$\mathsf{Wine} \sqcap \exists \mathsf{hasColor}.\mathsf{Red} \sqcap \forall \mathsf{producedIn}.\mathsf{BeaujolaisRegion} \ .$$

Such a complex concept description is satisfiable. Also, it is a subconcept of Wine*, that is, the following subsumption relation holds:*

$$\mathsf{Wine} \sqcap \exists \mathsf{hasColor}.\mathsf{Red} \sqcap \forall \mathsf{producedIn}.\mathsf{BeaujolaisRegion} \sqsubseteq \mathsf{Wine} \ .$$

In order to verify this, we will make use of the reduction stated in Theorem 2. By the theorem we get, that the subsumption above holds if and only if the following concept is unsatisfiable:

$$\mathsf{Wine} \sqcap \exists \mathsf{hasColor}.\mathsf{Red} \sqcap \forall \mathsf{producedIn}.\mathsf{BeaujolaisRegion} \sqcap \neg \mathsf{Wine} \ .$$

This is, however, obvious, because by the definition it is not possible for any individual to be an instance of Wine *and* $\neg$Wine *simultaneously.*

[2] For the notion of reducibility of decision problems we kindly refer the unacquainted reader to the work of Papadimitriou (1994).

We have learned above that DL are contained in FOL: concepts and roles correspond to unary and binary predicates, respectively. Pushing this correspondence further ahead, it is easy to see that constructors correspond to boolean connectives: complement corresponds to classical negation ($\neg$), concept intersection corresponds to conjunction ($\wedge$), concept union corresponds to disjunction ($\vee$), value restriction corresponds to the universal quantifier ($\forall$) and existential restriction to the existential quantifier ($\exists$). Of course, this correspondence is one way, from DL to FOL; not all universally quantified FOL formulae can be rewritten in DL, etc.

We will now list some of the basic properties of DL. These properties follow rather easily from the definitions, however, in light of the correspondence with FOL, they are simply reformulations of some of the well known tautologies that hold in FOL.

Theorem 5. *For any three concept descriptions C, D and E and for any role R, all of the following propositions are always true:*

1. $C \sqcap D \sqsubseteq C$;
2. $C \sqsubseteq C \sqcup D$;
3. $(C \sqcap D) \sqcup E \equiv (C \sqcup E) \sqcap (D \sqcup E)$;
4. $(C \sqcup D) \sqcap E \equiv (C \sqcap E) \sqcup (D \sqcap E)$;
5. $\neg\neg C \equiv C$;
6. $C \sqcap \neg C$ *is unsatisfiable (and hence* $\bot$ *is unsatisfiable);*
7. $C \sqsubseteq D \sqcup \neg D$ *(and hence* $C \sqsubseteq \top$*);*
8. $\exists R.C \sqcap \forall R.\neg C$ *is unsatisfiable.*

Sometimes we will rely on a particular normal form, into which it is possible to transform any complex concept description. Most important of these normal forms is the negation normal form:

Definition 7 (Negation normal form)**.** *A complex concept C is in the negation normal form (NNF), if the concept complement constructor ($\neg$) only appears in C in front of atomic concept names.*

To make an illustration, the concept $\neg(A \sqcap B)$ is not in NNF, because $\neg$ is applied on $A \sqcap B$ which is not atomic. On the other hand, $\neg A \sqcup \neg B$, an equivalent concept, is in NNF. The important point is, that it is possible to transform any concept into an equivalent concept in NNF. This is based on the following observation.

Lemma 6. *Given any two concept descriptions C, D and a role R, it always holds that:*

1. $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$,
2. $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$,
3. $\neg\exists R.C \equiv \forall R.\neg C$,

4. $\neg\forall R.C \equiv \exists R.\neg C$.

It is easy to see, that if the lemma is applied also recursively, on C and D, and so on, an equivalent concept in NNF is obtained for any $\mathcal{ALC}$ concept on the input.

Theorem 7. *For every $\mathcal{ALC}$ concept C, there exists another $\mathcal{ALC}$ concept C' in NNF, such that C and C' are equivalent.*

It is now safe to introduce the following notation. The NNF of C is denoted by $\mathrm{nnf}(C)$. And also, sometimes the NNF of $\neg C$ is denoted by $\dot{\neg}C$. Although not used so frequently as NNF, also the conjunctive and disjunctive normal forms, that we know from classical logic, exist in DL.

Definition 8 (Conjunctive normal form)**.** *A complex concept C is in the conjunctive normal form (CNF), if it has the form $C = E_1 \sqcap \cdots \sqcap E_n$, where E_1, ..., E_n are possibly complex concepts that do not contain the concept intersection constructor ($\sqcap$).*

Definition 9 (Disjunctive normal form)**.** *A complex concept C is in the disjunctive normal form (DNF), if it has the form $C = E_1 \sqcup \cdots \sqcup E_n$, where E_1, ..., E_n are possibly complex concepts that do not contain the concept union constructor ($\sqcup$).*

Theorem 8. *For every $\mathcal{ALC}$ concept C, there exists an equivalent concept C' in CNF, and another equivalent concept C'' in DNF.*

The theorem is a consequence of Theorem 5, clauses 3 and 4, which show how to transform complex concepts constructed by concept intersection and union. In addition, from Lemma 6 we learn how to deal with restrictions that are part of the concept description that we need to transform into CNF or DNF.

2.1.2 $\mathcal{ALC}$ Knowledge Bases

A knowledge base is a structure that contains some knowledge about the world, or about some specific domain, which is formally represented by formulae of certain language. In logic, knowledge bases are often called theories. In DL, our main interest is in terminological knowledge. Terminological structures are often called ontologies, and thence, also we will often refer to knowledge bases as ontologies.

DL formulae are built from concepts (which are possibly complex), roles and individuals. If a formula appears in a knowledge base, it is called an *axiom*. Ontologies (and thus also DL knowledge bases) typically consist of a *TBox* and an *ABox*. TBox contains terminological knowledge about concepts: what kind of concepts are there in the ontology and what are the relations between them. Here it may be stated that two concepts are equivalent or that one is more general than the other.

Another part of an ontology is the ABox, which consists of assertions about particular individuals. Here it may be stated that some individual belongs to some particular concept (even a complex one) or that two individuals are related by a particular role.

More expressive DL usually provide means to specify role inclusion and other properties of roles. Strictly speaking, this is also terminological knowledge. For practical reasons, it is often separately referred to as the *RBox*.

In this section, we describe the syntax and the semantics of TBox axioms and ABox assertions in $\mathcal{ALC}$, and we show how the decision problems are defined in context of an ontology.

TBoxes

Terminological Box, or TBox, collects terminological knowledge. This is the knowledge about concepts and relations between them. The most important relation between concepts is concept subsumption, which tells us, that some concepts are more specific and some are more general. For instance, Wine or Beverage are more general than Beaujolais. Subsumption is possibly expressed by a subsumption formula. If it appears in the TBox, it is called a subsumption axiom. If we place no restrictions on the concepts in a subsumption axiom, it is called a general concept inclusion.

Definition 10 ($\mathcal{ALC}$ TBox)**.** *An $\mathcal{ALC}$ TBox is a set of axioms called general concept inclusions (GCI) of the form*

$$C \sqsubseteq D \ ,$$

where both C and D are possibly complex concept descriptions. A syntactic shorthand $C \doteq D$ is used instead of two GCI $C \sqsubseteq D$ and $D \sqsubseteq C$.

A GCI $C \sqsubseteq D$ may also be read "D is more general than C" or "C is included in D", also $C \doteq D$ is commonly read "C and D are equivalent."

From the semantic point of view, a TBox constitutes a set of conditions that must all be satisfied in any possible "state of the world", and hence it constraints the set of admissible interpretations. Thus, it gives some (precisely determined) meaning to the vocabulary which is described by it.

Definition 11 (Model of a TBox)**.** *An interpretation $\mathcal{I}$ satisfies a GCI $C \sqsubseteq D$ (denoted by $\mathcal{I} \models C \sqsubseteq D$), if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ (denoted by $\mathcal{I} \models \mathcal{T}$), if it satisfies every GCI $C \sqsubseteq D \in \mathcal{T}$.*

The notion of equivalence is easily brought up to the level of TBoxes. Two TBoxes are equivalent, if they share the same set of models.

Definition 12 (Equivalence of TBoxes)**.** *Two TBoxes $\mathcal{T}$ and $\mathcal{T}'$ are equivalent if each model of $\mathcal{T}$ is also a model of $\mathcal{T}'$ and vice versa.*

Let us now have a brief look at an example of a TBox, and of general concept inclusion axioms. In this TBox we have an excerpt from a larger ontology that deals with classification of wine. We can see, that multiple relations between concepts are captured. Apart from subsumption, additional relations are modeled with restrictions on roles.

Example 2. *Consider a small ontology $\mathcal{T}$ providing descriptions of various kinds of wine, such as* Beaujolais, Burgundy, *or* Port*:*

$$\mathsf{Beaujolais} \sqcup \mathsf{Burgundy} \sqcup \mathsf{Port} \sqsubseteq \mathsf{Wine} \ ,$$
$$\mathsf{Beaujolais} \sqsubseteq \exists \mathsf{hasColor.Red} \sqcap \forall \mathsf{producedIn.BeaujolaisRegion} \ ,$$
$$\mathsf{Burgundy} \sqsubseteq \exists \mathsf{madeFrom.PinotNoirGrape} \sqcap \forall \mathsf{producedIn.BurgundyRegion} \ ,$$
$$\mathsf{Port} \sqsubseteq \forall \mathsf{hasBody.Full} \sqcap \forall \mathsf{producedIn.OPortoRegion} \ ,$$
$$\mathsf{BeaujolaisRegion} \sqcup \mathsf{BurgundyRegion} \sqsubseteq \mathsf{FranceRegion} \ ,$$
$$\mathsf{OPortoRegion} \sqsubseteq \mathsf{PortugalRegion} \ ,$$
$$\mathsf{FranceRegion} \sqcap \mathsf{PortugalRegion} \sqsubseteq \bot \ .$$

Once given a TBox $\mathcal{T}$, our perception of the decision tasks is altered. We are no longer interested whether something holds in general, in any model, now we only care about the models of $\mathcal{T}$. It may also be said, that $\mathcal{T}$ provides a description of the world, which restricts the set of possible models of the world that we need to consider for the decision.

Definition 13 (Satisfiability w.r.t. a TBox)**.** *A concept C is satisfiable (or consistent) with respect to a TBox $\mathcal{T}$, if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.*

Definition 14 (Subsumption w.r.t. a TBox)**.** *A concept C is subsumed by another concept D with respect to a TBox $\mathcal{T}$, denoted by $\mathcal{T} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model $\mathcal{I}$ of $\mathcal{T}$. It is also said that $\mathcal{T}$ entails $C \sqsubseteq D$.*

Definition 15 (Equivalent concepts w.r.t. a TBox)**.** *Two concepts C and D are equivalent with respect to a TBox $\mathcal{T}$, denoted by $\mathcal{T} \models C \equiv D$, if in every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $C^{\mathcal{I}} = D^{\mathcal{I}}$.*

Definition 16 (Disjoint concepts w.r.t. a TBox)**.** *Two concepts C and D are disjoint with respect to a TBox $\mathcal{T}$, if in every model $\mathcal{I}$ of $\mathcal{T}$ we have $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.*

The previously defined decision problems for the case with no TBox (Definitions 3–6) are properly generalized by these definitions. A concept C is consistent if and only if it is consistent with respect to the empty TBox $\emptyset$, a subsumption $C \sqsubseteq D$ holds if and only if $\emptyset \models C \sqsubseteq D$. Also, $C \equiv D$ if and only if $\emptyset \models C \equiv D$, and C and D are disjoint if and only if they are disjoint with respect to the empty TBox $\emptyset$. In addition, the reductions between these decision problems, as given by Theorems 1–4, also hold in the presence of a TBox.

Theorem 9. *A concept C is satisfiable with respect to a TBox $\mathcal{T}$ if and only if $\mathcal{T} \models C \sqsubseteq \bot$ does not hold.*

Theorem 10. *The subsumption relationship $C \sqsubseteq D$ holds with respect to a TBox $\mathcal{T}$ if and only if $C \sqcap \neg D$ is unsatisfiable with respect to $\mathcal{T}$.*

Theorem 11. *Concepts C and D are equivalent with respect to a TBox $\mathcal{T}$ if and only if both hold $\mathcal{T} \models C \sqsubseteq D$ and $\mathcal{T} \models D \sqsubseteq C$ (i.e., if and only if both $C \sqcap \neg D$ and $D \sqcap \neg C$ are unsatisfiable with respect to $\mathcal{T}$).*

Theorem 12. *Concepts C and D are disjoint with respect to a TBox $\mathcal{T}$ if and only if $C \sqcap D$ is unsatisfiable with respect to $\mathcal{T}$.*

Finally, after introducing TBoxes and the associated decision problems, the first signs of the true power of knowledge representation with DL become apparent. While part of the knowledge is explicitly recorded in the TBox, some knowledge is implied, even if it is not recorded explicitly. We will demonstrate this on our TBox from Example 2.

Example 3. *Please recall the ontology $\mathcal{T}$ from Example 2. Clearly, the following concept, that describes a strange kind of port wine that is produced in France, is unsatisfiable with respect to $\mathcal{T}$:*

$$\mathsf{Port} \sqcap \exists \mathsf{producedIn}.\mathsf{FranceRegion} \ .$$

Moreover, if we add some subconcept of, say, Port *to $\mathcal{T}$, let*

$$\mathcal{T}' = \mathcal{T} \cup \{\mathsf{VintagePort} \sqsubseteq \mathsf{Port} \sqcap \exists \mathsf{hasVintage}.\top\} \ ,$$

this subconcept is still a kind of wine; put more formally:

$$\mathcal{T}' \models \mathsf{VintagePort} \sqsubseteq \mathsf{Wine} \ .$$

ABoxes

Assertion Box, or ABox, collects different kind of knowledge, which we call assertional knowledge. These are assertions about particular individuals in the ontology. We may want to state that a particular individual belongs to some (possibly complex) concept, or that some pair of individuals is related by some particular role. Collections of such assertions are often called simply *data* because they basically resemble records in a relational database.

Definition 17 ($\mathcal{ALC}$ ABox)**.** *An $\mathcal{ALC}$ ABox is a set of axioms, each of one of the following forms:*

$$i : C \ , \qquad\qquad i, j : R \ ,$$

where C is a concept, R is a role and i and j are individuals. The first form is called concept assertion and the second form is called role assertion.

Sometimes an alternate notation, reminiscent of FOL is used: a concept assertion $i : C$ is written using $C(i)$, and a role assertion $i, j : R$ is written using $R(i, j)$. A concept assertion $i : C$ is read "i is of type C." Intuitively, it can be said, that the individual i is classified with concept label C, it belongs to C. A role assertion $i, j : R$ means that individuals i and j take part in the role R, they are inter-related by this role. The semantics of these expressions is formally given as follows.

Definition 18 (Semantics of ABox assertions)**.** *An interpretation $\mathcal{I}$ satisfies the concept assertion $i : C$, if $i^{\mathcal{I}} \in C^{\mathcal{I}}$. It satisfies the role assertion $i, j : R$, whenever $\langle i^{\mathcal{I}}, j^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$.*

With ABoxes we need to consider some completely new decision problems. First is the problem of ABox consistency.

Definition 19 (ABox consistency)**.** *An ABox $\mathcal{A}$ is consistent, if there exists an interpretation such that it satisfies each assertion in $\mathcal{A}$. Such interpretation is called a model of $\mathcal{A}$. An ABox $\mathcal{A}$ is consistent with respect to a TBox $\mathcal{T}$, if there exists a model of $\mathcal{T}$ such that it is also a model of $\mathcal{A}$.*

Clearly, the ABox consistency, viewed as reasoning task, allows us to determine, whether the data contained in the ABox are sound with respect to a given TBox. While ABox is possibly viewed as data, TBox is then viewed as some sort of database schema in this sense. As we have seen multiple times already, checking for consistency of some ABox $\mathcal{A}$ without any TBox is equivalent to checking for consistency of $\mathcal{A}$ with respect to the empty TBox $\emptyset$.

Another new decision problem associated with ABoxes is the problem of instance checking.

Definition 20 (Instance checking)**.** *Given an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, an individual i is an instance of a concept C, denoted by $\mathcal{A}, \mathcal{T} \models i : C$, if in each common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ it holds that $i^{\mathcal{I}} \in C^{\mathcal{I}}$. The task of verifying this condition is commonly called instance checking.*

Similarly, it is possible to define instance checking with empty TBox. Let us now briefly take a look at an example of an ABox and the two basic ABox decision problems.

Example 4. *Recall again the TBox $\mathcal{T}'$ from Example 3. Let*

$$\mathcal{T}'' = \mathcal{T}' \cup \{\mathsf{BottleOfWine} \sqsubseteq \mathsf{Bottle} \sqcap \exists \mathsf{contains.Wine}\}$$

With ABox assertions, we can model a situation concerning some particular person Jack, an owner of a particular bottle of wine. According to our notational conventions Jack will be represented by an individual jack*, as well as all other individuals will start with small letter. Consider the ABox $\mathcal{A}$:*

$$\begin{aligned} &\mathsf{bottle42} : \mathsf{BottleOfWine}\ , && \mathsf{bottle42}, \mathsf{jack} : \mathsf{ownBy}\ , \\ &\mathsf{port128} : \mathsf{Port}\ , && \mathsf{bottle42}, \mathsf{port128} : \mathsf{contains}\ , \\ & && \mathsf{port128}, \mathsf{vintage1945} : \mathsf{hasVintage}\ . \end{aligned}$$

The ABox $\mathcal{A}$ is consistent, and it is also consistent with respect to the TBox $\mathcal{T}''$. Moreover, it holds that

$$\mathcal{A}, \mathcal{T}'' \models \mathsf{port128} : \mathsf{VintagePort}\ .$$

On the other hand, the consistency of the above ABox is easily broken by additional assertions. Indeed, the ABox

$$\mathcal{A}' = \mathcal{A} \cup \{\mathsf{bottle42} : \exists \mathsf{producedIn.France}\}$$

is inconsistent with respect to $\mathcal{T}''$.

The other reasoning problems that we have got to know before are also of interest in the presence of an ABox. Here, the ABox represents additional constraints involving individuals.

Definition 21 (Satisfiability w.r.t. ABox and TBox)**.** *A concept C is satisfiable (or consistent) with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, if there exists a common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.*

Definition 22 (Subsumption w.r.t. ABox and TBox)**.** *A concept C is subsumed by another concept D with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, denoted by $\mathcal{A}, \mathcal{T} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$. It is also said that $\mathcal{A}$ and $\mathcal{T}$ entail $C \sqsubseteq D$.*

Definition 23 (Equivalence of concepts w.r.t. ABox and TBox)**.** *Two concepts C and D are equivalent with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, denoted by $\mathcal{A}, \mathcal{T} \models C \equiv D$, if for every common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ it holds that $C^{\mathcal{I}} = D^{\mathcal{I}}$.*

Definition 24 (Disjointness of concepts w.r.t. ABox and TBox)**.** *Two concepts C and D are disjoint with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, if in every common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ we have $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.*

These definitions also properly generalize the versions of these decision problems without considering an ABox (as given in Definitions 13–16). Given a TBox $\mathcal{T}$, a concept C is satisfiable with respect to $\mathcal{T}$ if and only if it is satisfiable with respect to the empty ABox $\emptyset$ and $\mathcal{T}$. Similarly, $\mathcal{T} \models C \sqsubseteq D$ if and only if $\emptyset, \mathcal{T} \models C \sqsubseteq D$; $\mathcal{T} \models C \equiv D$ if and only if $\emptyset, \mathcal{T} \models C \equiv D$; and finally, C and D are disjoint with respect to $\mathcal{T}$ if and only if they are disjoint with respect to the empty ABox $\emptyset$ and $\mathcal{T}$.

The reductions between the basic decision tasks, as they were presented in Theorems 1–4, hold also in this more general setting.

Theorem 13. *A concept C is satisfiable with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ if and only if $\mathcal{A}, \mathcal{T} \models C \sqsubseteq \bot$ does not hold.*

Theorem 14. *The subsumption relationship $C \sqsubseteq D$ holds with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ if and only if $C \sqcap \neg D$ is unsatisfiable with respect to $\mathcal{A}$ and $\mathcal{T}$.*

Theorem 15. *Concepts C and D are equivalent with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ if and only if both hold $\mathcal{A}, \mathcal{T} \models C \sqsubseteq D$ and $\mathcal{A}, \mathcal{T} \models D \sqsubseteq C$ (i.e., if and only if both $C \sqcap \neg D$ and $D \sqcap \neg C$ are unsatisfiable with respect to $\mathcal{A}$ and $\mathcal{T}$).*

Theorem 16. *Concepts C and D are disjoint with respect to an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ if and only if $C \sqcap D$ is unsatisfiable with respect to $\mathcal{A}$ and $\mathcal{T}$.*

Before we have learned, that concept satisfiability is the central decision problem, because all the other problems are reducible into it. When working with an ABox, however, the central decision problem is ABox consistency. This is because concept satisfiability and hence also all the other decision problems reduce into it.

Theorem 17. *Given an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, a concept C is satisfiable with respect to $\mathcal{A}$ and $\mathcal{T}$ if and only if the ABox $\mathcal{A}' = \mathcal{A} \cup \{i : C\}$, where i is some new individual that does not appear in $\mathcal{A}$, is consistent with respect to $\mathcal{T}$.*

Also instance checking, the new reasoning task for ABoxes, reduces into ABox consistency checking.

Theorem 18. *Given an ABox $\mathcal{A}$, a TBox $\mathcal{T}$, an individual i and a concept C, it follows that $\mathcal{A}, \mathcal{T} \models i : C$ if and only if the ABox $\mathcal{A}' = \mathcal{A} \cup \{i : \neg C\}$ is consistent with respect to $\mathcal{T}$.*

2.1.3 Reasoning with $\mathcal{ALC}$

What makes DL useful in practice, as a formalism suitable for representing ontologies, are the reasoning algorithms, which have been designed and implemented and can be used to reason about the decision problems, that we have

described above. Given an ontology, these reasoning engines allow us to check for instance whether some concept is satisfiable with respect to the ontology, that is, intuitively, whether it is meaningful. Furthermore, we may compute the classification of the ontology, that is, the complete subsumption hierarchy for all concepts. Or if a dataset is available in form of an ABox we may want to query about the data by instance checking.

In general, there exist multiple approaches in DL reasoning. The most frequently and most widely applied are the so called tableaux algorithms, that try to prove the existence of an interpretation by constructing its finite representation, so called tableau. We will focus entirely on this technique. For other reasoning techniques, see for instance the work of Hustadt et al. (2004), and also Baader et al. (2003), Tobies (2001).

Deciding Satisfiability

We start with a basic version of the tableaux algorithm, for deciding the satisfiability of a concept. As we have seen above, almost all other decision problems reduce into concept satisfiability checking. Later on, we will extend the algorithm in order to deal with TBoxes and ABoxes.

A tableaux algorithm constructs a *tableau* for a given input concept C. The tableau is a tree-shaped structure, that corresponds to an interpretation of C. Once a tableau for C is found, we know that an interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of C exists, such that $C^{\mathcal{I}}$ is not empty.

Each interpretation $\mathcal{I}$ is possibly viewed as a graph. In this graph, $\Delta^{\mathcal{I}}$ is the set of vertices and two elements of this set are connected by an edge if there is a role R that associates them. In case of the logic $\mathcal{ALC}$ we are especially lucky, since every satisfiable $\mathcal{ALC}$ concept has a finite tree-shaped interpretation.

Theorem 19. *For every $\mathcal{ALC}$ concept C, which is satisfiable, there exists an interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ with $C^{\mathcal{I}} \neq \emptyset$ that is a finite tree.*

The theorem is proven later on in this section. We say that $\mathcal{ALC}$ has the finite tree model property. It follows that $\mathcal{ALC}$ has also the finite model property and the tree model property. These findings imply, that the tableaux algorithm for $\mathcal{ALC}$ only needs to work with tree structures, since it is always possible to find an interpretation that is a finite tree if only an interpretation of the concept of question exists.

We start with an auxiliary notion of a subconcept. It turns out, that subconcepts of D, in the sense defined below, are somehow all relevant concepts that we need to consider, when building a tableau of D.

Definition 25. *Let D be an $\mathcal{ALC}$ concept. The subconcepts of D, the members of the set* $\mathrm{sub}(D)$*, are recursively defined as follows:*

1. $D \in \mathrm{sub}(D)$,
2. *if* $D = \neg C$, *then* $C \in \mathrm{sub}(D)$,
3. *if* $D = C_1 \sqcap C_2$, *then* $C_1 \in \mathrm{sub}(D)$ *and* $C_2 \in \mathrm{sub}(D)$,
4. *if* $D = C_1 \sqcup C_2$, *then* $C_1 \in \mathrm{sub}(D)$ *and* $C_2 \in \mathrm{sub}(D)$,
5. *if* $D = \forall R.C$, *then* $C \in \mathrm{sub}(D)$,

6. *if* $D = \exists R.C$, *then* $C \in \mathrm{sub}(D)$.

Note, that subconcepts of C in this "syntactic" sense are not subconcepts in the semantic sense, that is, concepts subsumed by C. This is not an issue, as it is always clear from context to which kind of subconcepts we refer. The "syntactic" subconcepts will usually appear only as members of $\mathrm{sub}(C)$.

Now we are ready to introduce tableaux formally. While we have previously noticed, that interpretations are thee shaped, we will build tableaux as labeled trees. In these trees, we will label nodes by concepts and edges by role names.

Definition 26 ($\mathcal{ALC}$ tableau)**.** *Let D be an $\mathcal{ALC}$ concept in NNF. Let R_D be the set of all roles occurring in D. A tableau T for D is a triple $(S, \mathcal{L}, \mathcal{E})$, such that S is a set of individuals, $\mathcal{L}: S \to 2^{\mathrm{sub}(D)}$ is a labeling function that labels each individual of S with a set of concepts under* $\mathrm{sub}(D)$, *$\mathcal{E}: R_D \to 2^{S \times S}$ is a labeling function that labels each role name in R_D with a set of edges, and there is some individual $s_0 \in S$ such that $D \in \mathcal{L}(s_0)$. Moreover, for each $s \in S$, for each $C, C_1, C_2 \in \mathrm{sub}(D)$ and for each $R \in R_D$ is holds that:*

1. *if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$,*
2. *if $C_1 \sqcap C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ and $C_2 \in \mathcal{L}(s)$,*
3. *if $C_1 \sqcup C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ or $C_2 \in \mathcal{L}(s)$,*
4. *if $\forall R.C \in \mathcal{L}(s)$, then for each $t \in S$ with $\langle s, t\rangle \in \mathcal{E}(R)$ we have $C \in \mathcal{L}(t)$,*
5. *if $\exists R.C \in \mathcal{L}(s)$, then there also exists some $t \in S$ such that $\langle s, t\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(t)$.*

The following theorem shows that each tableau for D has a one-to-one correspondence with some interpretation, moreover this interpretation interprets D by a non-empty set.

Theorem 20 (Baader et al. (2003))**.** *There exists a tableau T for an $\mathcal{ALC}$-concept D if and only if there exists some interpretation $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ such that $D^{\mathcal{I}} \neq \emptyset$.*

This theorem is easily proven by structural induction in D with the measure $\mathrm{sub}(D)$ (Baader et al. 2003). The tableaux algorithm works on a completion tree. During the run of the algorithm, the completion tree basically represents an incomplete tableau. When the algorithm is finished, either the tableau is completed, and in this case we know that the input concept is satisfiable, due to Theorem 20, or the completion tree cannot be completed in to a tableau, and hence the input concept is unsatisfiable.

Definition 27 (Completion tree)**.** *Let D be a concept in NNF and R_D is the set of all role names that appear in D. An $\mathcal{ALC}$ completion tree for D is a triple $(V, E, \mathcal{L})$, where (V, E) is a tree, with each node $x \in V$ labeled by $\mathcal{L}(x) \subseteq \mathrm{sub}(D)$, each edge $\langle x, y\rangle \in E$ is labeled by $\mathcal{L}(\langle x, y\rangle) \in R_D$.*

Sometimes the algorithm makes nondeterministic decisions. If something went wrong during the run of the algorithm, the algorithm needs to backtrack to the last decision point and try some other options. This inconsistent state, whenever it happens, is called a clash. The completed tableau must always be clash-free.

Definition 28 (Clash). *There is a clash in a completion tree $(V, E, \mathcal{L})$ for a concept D, if for some node $x \in V$ and for some concept C both $C \in \mathcal{L}(x)$ and $\neg C \in \mathcal{L}(x)$. The completion tree is clash-free if there is no clash in any $x \in V$.*

Finally we are ready to introduce the tableaux algorithm. On the input the algorithm expects an $\mathcal{ALC}$ concept in NNF. Recall, that every concept is easily transformed into an equivalent concept in NNF.

Algorithm 1 (Satisfiability in $\mathcal{ALC}$). *Given an input concept C in NNF, the algorithm executes three steps:*

1. initialization: *create a new completion tree $(V, \emptyset, \mathcal{L})$ with a single node $s_0 \in V$ and initialize the label of s_0 to $\mathcal{L}(s_0) = C$;*
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.1. This step is over when none of the rules is applicable;*
3. answer: *if the completion tree is clash-free, answer "C is satisfiable". Otherwise, answer "C is unsatisfiable".*

Let us have a closer look at the tableaux expansion rules used by the algorithm. The $\sqcap$-rule is rather simplistic, it breaks the complex concept descriptions in labels of nodes down and introduces the subconcepts into the labels. The $\sqcup$-rule is applied non-deterministically and in practice, whenever it is applied, both possibilities need to be verified. Hence, $\sqcup$-rule is a source of branching. The $\exists$-rule generates a new node, whenever it is applied: the only way to satisfy the existential restriction within a tree like structure. The $\forall$-rule may be applied several times on a given node, each time after a new successor is introduced by the $\exists$-rule.

We will now formally prove that the tableaux algorithm described above is correct. The proof splits into three lemmata, and the final conclusion is drawn bellow in Theorem 24.

Lemma 21 (Termination). *The tableaux algorithm for deciding the satisfiability of $\mathcal{ALC}$ concepts always terminates.*

Proof. In each step a tableaux rule is applied if a corresponding complex concept C is found in a label of some node x of the completion tree $(V, E, \mathcal{L})$. In order to see that the algorithm indeed terminates after finite number of steps, consider the following three observations:

1. Rule-application is only triggered once by each concept $C \in \mathcal{L}(x)$ for each label $x \in V$ thanks to application conditions included in each rule.
2. Each time a new concept C' is added to the label of some $y \in V$ as a result of rule-application triggered by some concept $C \in \mathcal{L}(x)$ with $x \in V$, then the newly added concept C' is strictly string-shorter than the existing concept C that has triggered the rule application.
3. The algorithm is started with single node $s_0 \in V$ that has single element in $\mathcal{L}(s_0)$ that is represented by a finite string.

And hence the algorithm cannot continue in an endless loop. □

$\sqcap$-rule
If $C_1 \sqcap C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(X)$,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule
If $C_1 \sqcup C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \cap \mathcal{L}(X) = \emptyset$,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\forall$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x, y \in V$, y R-successor x with $C \notin \mathcal{L}(y)$,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.

$\exists$-rule
If $\exists R.C \in \mathcal{L}(x)$, for $x \in V$ with no R-successor y s.t. $C \in \mathcal{L}(y)$,
then add new node z to V with $\mathcal{L}(z) = \{C\}$ and $\mathcal{L}(\langle x, z\rangle) = \{R\}$.

Figure 2.1: $\mathcal{ALC}$ tableaux expansion rules

Lemma 22 (Soundness)**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{ALC}$ concepts is sound (i.e., whenever a complete and clash-free completion tree is constructed for an input concept C, then C is satisfiable.)*

Proof. Given an input concept C, let $T = (V, E, \mathcal{L})$ be the complete and clash-free completion tree constructed by the algorithm. From T we construct a canonical interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ in the following way:

1. $\Delta^{\mathcal{I}} = V$,
2. $A^{\mathcal{I}} = \{x \in V \mid C \in \mathcal{L}(x)\}$, for each atomic concept A,
3. $R^{\mathcal{I}} = \{\langle x, y\rangle \in E \mid R \in \mathcal{L}(\langle x, y\rangle)\}$.

We will show by structural induction that for each $x \in \Delta^{\mathcal{I}}$ and for each $D \in \mathrm{sub}(C)$:

$$D \in \mathcal{L}(x) \implies x \in D^{\mathcal{I}} \ .$$

Given $x \in \Delta^{\mathcal{I}}$ and $D \in \mathcal{L}(x)$, based on the structure of D:

1. If D is atomic, then $x \in D^{\mathcal{I}}$ by construction of $\mathcal{I}$.
2. It $D = \neg A$ for some atomic concept D, then $A \notin \mathcal{L}(x)$ since T is clash-free, by construction of $\mathcal{I}$ this means that $x \in D^{\mathcal{I}}$. (Thanks to NNF-only input concepts we don't have to deal with $D = \neg C$ with C non-atomic.)
3. If $D = C_1 \sqcap C_2$ for some concepts C_1 and C_2 then both C_1 and C_2 belong to $\mathcal{L}(x)$ since the $\sqcap$-rule is not applicable on x. From induction hypothesis we get that $x \in {C_1}^{\mathcal{I}}$ and $x \in {C_2}^{\mathcal{I}}$, and hence also $x \in D^{\mathcal{I}}$.
4. If $D = C_1 \sqcup C_2$ for some concepts C_1 and C_2 then either C_1 or C_2 belong to $\mathcal{L}(x)$ since the $\sqcup$-rule is not applicable on x. By induction either $x \in {C_1}^{\mathcal{I}}$ or $x \in {C_2}^{\mathcal{I}}$, and hence also $x \in D^{\mathcal{I}}$.
5. If $D = \forall R.C$ for some role R and concept C, then for each R-successor y of x we have $C \in \mathcal{L}(y)$ since the $\forall$-rule is not applicable on x. By induction $y \in C^{\mathcal{I}}$ and hence $x \in D^{\mathcal{I}}$.

6. If $D = \exists R.C$ for some role R and concept C, then there is an R-successor y of x such that $C \in \mathcal{L}(y)$ since the $\exists$-rule is not applicable on x. By induction $y \in C^{\mathcal{I}}$ and hence $x \in D^{\mathcal{I}}$.

Finally, since $C \in \mathcal{L}(s_0)$ then $s_0 \in C^{\mathcal{I}}$ and that means that C is satisfiable. □

Lemma 23 (Completeness)**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{ALC}$ concepts is complete (i.e., whenever some $\mathcal{ALC}$-concept C is satisfiable then the algorithm constructs a complete and clash-free completion tree when run with C on input.)*

Proof. Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, possibly infinite and cyclic, and such that $C^{\mathcal{I}} \neq \emptyset$, we shall prove that the algorithm produces a complete and clash-free completion tree $T = (V, E, \mathcal{L})$. We already know that the algorithm terminates, it is clear that it produces a complete tree. It remains to prove that clash is not introduced into T at any computation step. We will inductively construct a mapping $\pi : V \to \Delta^{\mathcal{I}}$ with the property:

$$C \in \mathcal{L}(x) \implies \pi(x) \in C^{\mathcal{I}} \ . \tag{2.1}$$

Base case. When T is initialized to $(\{s_0\}, \emptyset, \mathcal{L})$ with $\mathcal{L}(s_0) = \{C\}$ pick arbitrary $x \in \Delta^{\mathcal{I}}$ such that $x \in C^{\mathcal{I}}$ and assign $\pi(s_0) = x$. Indeed (2.1) is satisfied for C and s_0.

Induction step. Every time a rule is applied and new nodes and/or labels are introduced to V we verify (2.1):

1. If the $\sqcap$-rule is triggered by x with $C_1 \sqcap C_2 \in \mathcal{L}(x)$ and C_1, C_2 are added to $\mathcal{L}(x)$, then we know by induction that $\pi(x) \in C_1 \sqcap C_2{}^{\mathcal{I}}$ but since $\left(C_1 \sqcap C_2{}^{\mathcal{I}}\right) = C_1{}^{\mathcal{I}} \cap C_2{}^{\mathcal{I}}$ we also have $\pi(x) \in C_1{}^{\mathcal{I}}$ and $\pi(x) \in C_2{}^{\mathcal{I}}$.

2. If the $\sqcup$-rule is triggered by x with $C_1 \sqcup C_2 \in \mathcal{L}(x)$, then we know by induction that $\pi(x) \in C_1 \sqcup C_2{}^{\mathcal{I}}$ and that means that at least for one $i \in 1, 2$ we have $\pi(x) \in C_i{}^{\mathcal{I}}$. We know that in the nondeterministic run at least one correct branch is contained (if there is one) and hence we may assume that C_1 has been selected and added to $\mathcal{L}(x)$.

3. If the $\exists$-rule is applied on $x \in V$ a new R-successor z is introduced into V with $C_1 \in \mathcal{L}(z)$, then we know by induction that there is z', an R-successor of $\pi(x)$ in $\mathcal{I}$ such that $z' \in C_1{}^{\mathcal{I}}$. Therefore we assign $\pi(z) = z'$.

4. If the $\forall$-rule is applied on $x \in V$ with $\forall R.C_1 \in \mathcal{L}(x)$ then C_1 is added to all R-successors of x in T. However, we know by induction that $x \in \forall R.C_1{}^{\mathcal{I}}$ and hence for each R-successor y of $\pi(x)$ in $\mathcal{I}$ we have $y \in C_1{}^{\mathcal{I}}$. And indeed, by induction all R-successors of x in T are mapped by π to R-successors of $\pi(x)$ in $\mathcal{I}$.

And so T is clash-free as thanks to (2.1) a clash in some $x \in V$ would counter the assumption that $\mathcal{I}$ is an interpretation. □

Finally, we summarize the three lemmata and derive the correctness of the tableaux decision algorithm for $\mathcal{ALC}$.

Theorem 24 (Baader et al. (2003))**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{ALC}$ concepts described in this section always terminates, and it is sound and complete (i.e., it constructs a complete and clash-free completion tree if and only if the input concept is satisfiable).*

Moreover, we have proven the finite tree model property on the way (see Theorem 19). From Lemma 23 we know that if an $\mathcal{ALC}$ concept is satisfiable then the tableaux algorithm constructs a complete and clash-free completion tree T. In the proof of Lemma 22 we have shown that in such a case we are able to construct the canonical interpretation $\mathcal{I}$ of C from T. Thanks to correspondence to T indeed $\mathcal{I}$ is tree-shaped and finite.

Reasoning with TBoxes

In order to verify whether a concept E is satisfiable with respect to a TBox $\mathcal{T}$, we need to find an interpretation $\mathcal{I}$ such that, on one hand $E^{\mathcal{I}}$ is nonempty, and on the other hand $\mathcal{I}$ is also a model of $\mathcal{T}$. It turns out, that this is possibly achieved by constructing a tableau by the tableaux algorithm for concept satisfiability that we already know, however, during the run of the algorithm, we must verify the validity of the GCI axioms of $\mathcal{T}$ in addition, whenever necessary.

Please observe, that any GCI ϕ of the from $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq \neg C \sqcup D$. In other words, ϕ is satisfied by $\mathcal{I}$ if and only if for every $x \in \Delta^{\mathcal{I}}$ we have that $x \in (\neg C \sqcup D)^{\mathcal{I}}$, that is, x is either an instance of $\neg C$ or it is an instance of D. This observation gives us a clue how we need to alter the tableaux algorithm. Whenever a new node s is introduced into the completion tree, we must add $\mathrm{nnf}(\neg C \sqcup D)$ to the label of s, for every GCI $C \sqsubseteq D \in \mathcal{T}$, and then continue with expanding the completion tree as before. This guarantees, that once a completion tree is complete, it is a model of the input concept E and also of $\mathcal{T}$ in addition. We will introduce a new tableaux rule, called $\mathcal{T}$-rule that will be responsible for this.

However, as the following example shows, finite tree model property no longer holds in the presence of a TBox. Therefore we must be cautious, in order to guarantee the termination of the algorithm.

Example 5. *Consider the TBox $\mathcal{T} = \{C \sqsubseteq \exists R.C\}$. Let us trace the run of the tableaux algorithm, naïvely extended as described above, in order to verify the satisfiability of C with respect to $\mathcal{T}$:*

1. *first, the completion tree $(V, E, \mathcal{L})$ is initialized with $V = \{s_0\}$, $E = \emptyset$, $\mathcal{L}(s_0) = \{C\}$;*

2. *according to our analysis, we extend $\mathcal{L}(s_0)$ to $\{C, \neg C \sqcup \exists R.C\}$, in oder to deal with the GCI;*

3. *applying the $\sqcup$-rule, we first add $\neg C$ to $\mathcal{L}(s_0)$, however, this results into a clash, hence we backtrack and add $\exists R.C$ to $\mathcal{L}(s_0)$ instead. After this step, $\mathcal{L}(s_0) = \{C, \neg C \sqcup \exists R.C, \exists R.C\}$;*

4. *the only possibility now is to apply the $\exists$-rule. We create a new node, say s_1, and set $V = \{s_0, s_1\}$. We also set $E = \{\langle s_0, s_1\rangle\}$, $\mathcal{L}(\langle s_0, s_1\rangle) = \{R\}$, and $\mathcal{L}(s_1) = \{C\}$;*

$\sqcap$-rule
If $C_1 \sqcap C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(X)$,
and x is not blocked,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule
If $C_1 \sqcup C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \cap \mathcal{L}(X) = \emptyset$,
and x is not blocked,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\forall$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x, y \in V$, y R-successor x with $C \notin \mathcal{L}(y)$,
and x is not blocked,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.

$\exists$-rule
If $\exists R.C \in \mathcal{L}(x)$, for $x \in V$ with no R-successor y s.t. $C \in \mathcal{L}(y)$,
and x is not blocked,
then add new node z to V with $\mathcal{L}(z) = \{C\}$ and $\mathcal{L}(\langle x, z\rangle) = \{R\}$.

$\mathcal{T}$-rule
If $C \sqsubseteq D \in \mathcal{T}$ and for some $x \in V$ $\mathrm{nnf}(\neg C \sqcup D) \notin \mathcal{L}(x)$,
and x is not blocked,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{\mathrm{nnf}(\neg C \sqcup D)\}$.

Figure 2.2: $\mathcal{ALC}$ tableaux expansion rules with TBox

5. *according to our analysis, we now have to add $\neg C \sqcup \exists R.C$ to $\mathcal{L}(s_1)$. This would, in the end, result into repeating steps 2, 3 and 4 on the node s_1 and in creation of yet another node, say s_2, with $\mathcal{L}(s_2) = \{C\}$. Even so, the algorithm would not terminate and the whole situation would repeat again and again, generating nodes in an endless loops.*

It turns out that, the algorithm is not completely misbehaved in the previous example. There is no finite tree-shaped model of $\mathcal{T}$ in which $C^{\mathcal{I}}$ is nonempty! The algorithm tries to construct the only kind of tree-shaped model that there is: an infinite tree, in which each node x_i with $C \in \mathcal{L}(x_i)$ has an R-successor x_{i+1} with $C \in \mathcal{L}(x_{i+1})$. Of course, this is not feasible in finite time.

We overcome this slight complication by a workaround called *blocking*, that is also frequently employed in decision techniques for richer DL. Essentially, if a node x in a completion tree has an ancestor[3] y such that $\mathcal{L}(x) \subseteq \mathcal{L}(y)$, then this situation is similar to the example above. Further expansion of the completion tree starting at x would result into periodical construction of an infinite model. At this point, we simply mark x as blocked, and stop the expansion in this node. If the portion of the completion tree between y and x goes fine (i.e., no clash occurs here) we are sure that the infinite model that periodically repeats the part between y and x exists.

[3] In the graph theory, the ancestor relation is obtained as the transitive closure on the predecessor relation and the descendant relation is obtained as the transitive closure of the successor relation.

Definition 29 (Subset blocking)**.** *Let $(V, E, \mathcal{L})$ be a completion tree. A node $x \in V$ is directly blocked, if it has a sole ancestor $y \in V$ such that $\mathcal{L}(x) \subseteq \mathcal{L}(y)$. A node $x \in V$ is indirectly blocked, if it has an ancestor $y \in V$ that is directly blocked. Finally, any node $x \in V$ is blocked if it is directly or indirectly blocked.*

This kind of blocking is called *subset blocking*, since we block whenever a node with the label that is a subset of the label of some of his ancestors is found in the completion tree. We will see other kinds of blocking later on. Now, all the tableaux expansion rules have to be modified, in order to implement blocking into the algorithm. The tableaux algorithm for satisfiability checking with respect to a TBox works exactly the same way, as described before, but instead of the four rules from Fig. 2.1 it uses the newly given rules of Fig. 2.2.

Algorithm 2 (Satisfiability in $\mathcal{ALC}$ w.r.t. a TBox)**.** *Given an input concept C in NNF and an input TBox $\mathcal{T}$, the algorithm executes three steps:*

1. initialization: *create a new completion tree $(V, \emptyset, \mathcal{L})$ with a single node $s_0 \in V$ and initialize the label of s_0 to $\mathcal{L}(s_0) = C$;*
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.2. This step is over when none of the rules is applicable;*
3. answer: *if the completion tree is clash-free, answer "C is satisfiable w.r.t. $\mathcal{T}$". Otherwise, answer "C is unsatisfiable w.r.t. $\mathcal{T}$".*

Theorem 25 (Baader et al. (2003))**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{ALC}$ concepts with respect to a TBox always terminates, and it is sound and complete (i.e., it constructs a complete and clash-free completion tree if and only if the input concept is satisfiable with respect to the input TBox).*

Reasoning with ABoxes

In presence of ABoxes, the main reasoning task is ABox consistency checking. This is because, as we have learned, all other decision problem reduce into ABox consistency checking (see Theorems 13–18).

Let us now have a look at how the tableaux algorithm for concept satisfiability checking is extended for ABox consistency checking with respect to a given TBox. The algorithm works very similarly, and in fact, it only differs in the initialization step.

Algorithm 3 (ABox consistency in $\mathcal{ALC}$ w.r.t. a TBox)**.** *Given an input ABox $\mathcal{A}$ in NNF and an input TBox $\mathcal{T}$, the algorithm executes three steps:*

1. initialization: *create a new completion tree $(V, E, \mathcal{L})$ and initialize V, E and $\mathcal{L}$ as follows:*
   - $V = \{i \in N_{\mathrm{I}} \mid i \text{ appears in } \mathcal{A}\}$,
   - $E = \{\langle i, j\rangle \in N_{\mathrm{I}} \times N_{\mathrm{I}} \mid \langle i, j\rangle : R \in \mathcal{A},\ R \in N_{\mathrm{R}}\}$,
   - *for each* $i \in V$ *assign* $\mathcal{L}(i) = \{C \mid i : C \in \mathcal{A}\}$,
   - *and for each* $\langle i, j\rangle \in E$ *assign* $\mathcal{L}(\langle i, j\rangle) = \{R \mid \langle i, j\rangle : R \in \mathcal{A}\}$.
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.2. This step is over when none of the rules is applicable;*

3. answer: *if the completion tree is clash-free, answer "$\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$". Otherwise, answer "$\mathcal{A}$ is inconsistent w.r.t. $\mathcal{T}$".*

Theorem 26 (Baader et al. (2003))**.** *The tableaux algorithm for ABox consistency checking with respect to a given TBox, as it is described above, always terminates, and it is sound and complete (i.e., it yields a complete and clash-free completion tree if and only if the input ABox is consistent with respect to the input TBox).*

If we reconsider now how the tableaux algorithm is initialized in the case of checking the satisfiability of a concept C, we see, that in this case we essentially check for the consistency of an ABox $\mathcal{A} = \{i : C\}$ for some individual $i \in N_{\mathrm{I}}$. This should be no news for us, since it corresponds to the reduction, as shown before (Theorem 17). Using the very same reduction, the algorithm for checking the satisfiability of a concept C with respect to a given TBox $\mathcal{T}$ and ABox $\mathcal{A}$, is obtained by checking for the consistency of $\mathcal{A}' = \mathcal{A} \cup \{i : C\}$ with respect to $\mathcal{T}$, where i is some new individual that does not appear in $\mathcal{A}$.

2.1.4 Computational complexity

One of the points which makes DL so versatile, is the trade-off between expressivity and computational complexity[4] of the different members of the DL family of logics.[5] We will see, that even for the basic DL $\mathcal{ALC}$, the complexity of reasoning is quite hard (measured by the worst-case complexity). Without taking a TBox into account, the complexity of reasoning in $\mathcal{ALC}$ is in PSpace.

Theorem 27 (Schmidt-Schauß & Smolka (1991))**.** *The problem of satisfiability checking of $\mathcal{ALC}$ concepts is PSpace-complete.*

Theorem 28 (Schmidt-Schauß & Smolka (1991))**.** *The problem of consistency checking for $\mathcal{ALC}$ ABoxes is PSpace-complete.*

Complexity of reasoning with TBoxes is even harder, it rises up to the ExpTime class. This is especially caused by the fact that each GCI has to be verified in every node of the completion tree, on combination with nondeterministic choices introduced by the $\sqcup$-rule.

Theorem 29 (Schild (1991))**.** *The problems of satisfiability checking for concepts and ABox consistency checking with respect to TBox in $\mathcal{ALC}$ are both ExpTime-complete.*

Even if, at the first glance, these worst-case complexity results may suggest very low performance of the algorithm, the empirical results have proven that the tableaux reasoners are usable in practice. The tableaux algorithm described by this section is just a start for an implementation of a practically usable reasoner. A lot of work has been done in optimization and the reasoners have been gradually improving for several years now. Some of the well known optimized DL reasoners include: FaCT (Horrocks 1998), FaCT++ (Tsarkov & Horrocks

[4] The reader unacquainted with the complexity theory and the complexity classes is kindly referred to the work of Papadimitriou (1994).

[5] Description Logic Complexity Navigator, maintained by Evgeny Zolin, is a great online tool to determine the complexity results for any specific DL of interest. Visit the URL: `http://www.cs.man.ac.uk/~ezolin/dl/`.

2006), RACER (Haarslev & Müller 2001), Pellet (Sirin et al. 2007), HermiT, based on so-called hypertableaux calculus (Motik et al. 2009), and KAON2, based on translation into disjunctive datalog and consecutive application of resolution techniques (Hustadt et al. 2004).

It is also worth noting, that there are less expressive DL than $\mathcal{ALC}$, and the main motivation behind them is improved computational complexity. Perhaps the most accurate example of this is DL-Lite (Calvanese et al. 2005), in which the core decision problems are in the complexity class P. Other sub-$\mathcal{ALC}$ DL include $\mathcal{EL}^+$ (Baader et al. 2006), $\mathcal{FL}_0$ (Baader et al. 2003) and multiple languages in the $\mathcal{AL}$ family (Schmidt-Schauß & Smolka 1991).

In this section, we have presented the DL $\mathcal{ALC}$, equipped with a basic family of constructors which are most often used and usually shared by other more expressive DL. We have learned about the tableaux decision algorithm, that is used to reason with $\mathcal{ALC}$. There is a number of more expressive DL, equipped with interesting additional constructors, that have a great deal of use in practice. Throughout the rest of the chapter, several extensions of $\mathcal{ALC}$ will be presented, finally reaching the expressive power of $\mathcal{SROIQ}$, that is the logical base behind the OWL Web Ontology Language (W3C OWL Working Group 2009).

2.2 Transitivity, role hierarchies and inverses

While $\mathcal{ALC}$ offers basic expressivity to work with concepts, it offers almost very little to handle roles. Roles are only any good in value restrictions and existential restrictions, which are both concepts constructors, as we already know. In addition, role assertion axioms are allowed in the ABox. In this section, we will introduce more powerful expressive features for working with roles. We will extend $\mathcal{ALC}$ with transitive and inverse roles and we will introduce so called role inclusion axioms that allow us to assert subsumption between roles, ending up with the DL $\mathcal{SHI}$.

The main motivation behind $\mathcal{SHI}$, is providing some better expressivity for representing so called aggregated objects. These are objects with multiple subparts. Basically we aim at modeling partonomy. We have already attempted to do this, when we modeled about regions in Example 2. In this example, we have used a workaround when we have interleaved taxonomy and partonomy, representing them both by subsumption. We have had both $\mathsf{Port} \sqsubseteq \mathsf{Wine}$ and $\mathsf{OPortoRegion} \sqsubseteq \mathsf{PortugalRegion}$, even if each of this assertions stands for a different kind of relation. Clearly, this is not possible in more complex scenarios.

A more practical modeling approach is to use subsumption for taxonomy, and roles for partonomy. At this point, however, the need for transitive roles arises. For instance, when we wish to model that students are part of a faculty and faculties are in turn part of some university, we may do so by the axioms:

$$\mathsf{Student} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{Faculty} \ , \qquad \mathsf{Faculty} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{University} \ ,$$

With these axioms in the knowledge base, we hope to derive that each student must also be a part of some university. This would follow, if the role $\mathsf{isPartOf}$ was interpreted by a transitive relation.

Role inclusion axioms (RIA) allow us to specify subroles and superroles. Imagine, that we want to distinguish between being a student of a faculty, and being a division of a university with two separate roles. We remodel the two

GCI by $\mathsf{Student} \sqsubseteq \exists \mathsf{isStudentOf}.\mathsf{Faculty}$ and $\mathsf{Faculty} \sqsubseteq \exists \mathsf{isDivisionOf}.\mathsf{University}$. Then we assert two RIA axioms:

$$\mathsf{isStudentOf} \sqsubseteq \mathsf{isPartOf} \ , \qquad \mathsf{isDivisionOf} \sqsubseteq \mathsf{isPartOf} \ .$$

From this knowledge, we are able to infer that $\mathsf{Student} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{Faculty}$, and assuming that $\mathsf{isPartOf}$ is a transitive role, we also infer out that $\mathsf{Student} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{University}$.

The last new feature, is the possibility to use also the role R^- alongside R, which is always interpreted by the inverse relation of $R^{\mathcal{I}}$. Combining this feature with RIA axioms, we may assert:

$$\mathsf{hasPart} \sqsubseteq \mathsf{isPartOf}^- \ , \qquad \mathsf{isPartOf}^- \sqsubseteq \mathsf{hasPart} \ .$$

If wee choose to do this, we effectively define the role $\mathsf{hasPart}$ as an inverse of $\mathsf{isPartOf}^-$, which is very intuitively used in modeling, as whenever $i, j : \mathsf{isPartOf}$ holds, also $j, i : \mathsf{hasPart}$ is implied.

2.2.1 $\mathcal{SHI}$

$\mathcal{SHI}$ may be also called $\mathcal{ALCHI}_{\mathrm{R}+}$.[6] Here, $\mathcal{H}$ stands for role hierarchies (or role inclusion axioms), $\mathcal{I}$ for inverse roles, and $R+$ stands for the possibility to declare transitive roles. Although all its constructors were known prior, $\mathcal{SHI}$ has become a particularly considerable logic after a tableaux algorithm was introduced by Horrocks & Sattler (1999). Upon this work also our survey is based. Let us start with the syntax of $\mathcal{SHI}$.

Definition 30 (Syntax of $\mathcal{SHI}$)**.** *Let N_{I}, N_{C}, and N_{R} be sets of names for individuals, concepts and roles respectively. Let ${N_{\mathrm{R}}}^+ \subseteq N_{\mathrm{R}}$ be a set of names for transitive roles. The set of all $\mathcal{SHI}$ roles is $N_{\mathrm{R}} \cup \{R^- \mid R \in N_{\mathrm{R}}\}$. With a slight abuse of notation, we define a function $\cdot^-$, which for any $\mathcal{SHI}$ role R returns R^- if $R \in N_{\mathrm{R}}$ and it returns S if $R = S^-$, $S \in N_{\mathrm{R}}$. Furthermore, a role R is transitive[7] (denoted by* $\mathrm{Trans}(R)$*) if either $R \in {N_{\mathrm{R}}}^+$ or $R^- \in {N_{\mathrm{R}}}^+$.*

A $\mathcal{SHI}$ concept is

1. *any atomic concept $A \in N_{\mathrm{C}}$,*
2. *the complement $\neg C$,*
3. *the intersection of two concepts $C \sqcap D$,*
4. *the union of two concepts $C \sqcup D$,*
5. *the value restriction $\forall R.C$,*
6. *and the existential restriction $\exists R.C$,*

[6] Adding more and more constructors to $\mathcal{AL}$ and so adding more and more letters to its name started to cause the names to be a bit clumsy. Hence, at some point, $\mathcal{ALC}_{\mathrm{R}+}$ was renamed to $\mathcal{S}$ thanks to its known correspondence with the propositional multi-modal logic $\mathbf{S4}_{(\mathbf{m})}$ (Schild 1991).

[7] Later on, when we introduce the semantics of $\mathcal{SHI}$, the reader may easily verify that the notion of transitive role is properly defined. The semantics implies that a role is transitive if and only if R^- is transitive.

for any two $\mathcal{SHI}$ concepts C and D and for any $\mathcal{SHI}$ role R. A role inclusion axiom is of the form

$$R \sqsubseteq S \ ,$$

where R and S are possibly inverse roles. A set of role inclusion axioms is denoted by $\mathcal{R}$. Finally, the set

$$\mathcal{R}^+ = (\mathcal{R} \cup \{R^- \sqsubseteq S^- \mid R \sqsubseteq S \in \mathcal{R}\}, \sqsubseteq^*) \ ,$$

with $\sqsubseteq^*$ being the transitive-reflexive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{R^- \sqsubseteq S^- \mid R \sqsubseteq S \in \mathcal{R}\}$, is a role hierarchy.

A $\mathcal{SHI}$ TBox is a collection of GCI axioms of the form $C \sqsubseteq D$, for any $\mathcal{SHI}$ concepts C and D. A $\mathcal{SHI}$ ABox is a collection of ABox assertions of the form $i : C$ or $i, j : R$, for any $i, j \in N_{\mathrm{I}}$, any $\mathcal{SHI}$ concept C, and any $\mathcal{SHI}$ role R.

Note that the collection of all RIA axioms in the knowledge base, together with some additional role related axioms in more expressive DL, is sometimes called the RBox.

The semantics of $\mathcal{SHI}$ extends the one of $\mathcal{ALC}$ by introducing further restrictions on interpretations in order to deal with the newly added constructs. Note that, in a sense, any $\mathcal{SHI}$ interpretation is already a "model" of the RBox. This underlines the fact, that we are not interested in reasoning about the role subsumption as a decision problem. Instead, the role hierarchy simply generates additional constraints that must be always satisfied by any interpretation.

Definition 31 (Semantics of $\mathcal{SHI}$)**.** *Given some vocabulary N_{C}, N_{R}, N_{I} and a role hierarchy $\mathcal{R}^+$, a $\mathcal{SHI}$ interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, such that $\Delta^{\mathcal{I}} \neq \emptyset$, and the interpretation function $\cdot^{\mathcal{I}}$ maps every individual $i \in N_{\mathrm{I}}$ to some $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every $\mathcal{SHI}$ concept C to some $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every $\mathcal{SHI}$ role R to a some $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that for any $\mathcal{SHI}$ concepts C, D and for any $\mathcal{SHI}$ roles R, S the conditions listed in Table 2.2 are satisfied.*

A $\mathcal{SHI}$ interpretation satisfies the elements of the language (denoted $\mathcal{I} \models \cdot$) as usual:

1. *$\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;*
2. *$\mathcal{I} \models i : C$ if $i^{\mathcal{I}} \in C^{\mathcal{I}}$;*
3. *$\mathcal{I} \models i, j : R$ if $\langle i^{\mathcal{I}}, j^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$;*
4. *$\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ is a model of $\mathcal{T}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{T}$;*
5. *$\mathcal{I} \models \mathcal{A}$ ($\mathcal{I}$ is a model of $\mathcal{A}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{A}$.*

All decision problems for $\mathcal{SHI}$ are defined as they have been defined for $\mathcal{ALC}$. Also, all the reductions among the decision problems, as presented for $\mathcal{ALC}$, carry over to $\mathcal{SHI}$ directly.

2.2.2 Reasoning with $\mathcal{SHI}$

As we already know, the tableaux algorithm always tries to construct a tableau for some concept D, which is a finite representation of an interpretation of D. For $\mathcal{SHI}$ tableaux, we need to extend the notion of all relevant concepts (Definition 25) that have to be considered in the tableau.

Construct	Condition
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\forall y \in \Delta^{\mathcal{I}})\ \langle x, y\rangle \in R^{\mathcal{I}} \implies y \in C\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\exists y \in \Delta^{\mathcal{I}})\ \langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C\}$
$R \in N_{\mathrm{R}}$	$\langle x, y\rangle \in R^{\mathcal{I}} \iff \langle y, x\rangle \in R^{-\mathcal{I}}$
$R \in {N_{\mathrm{R}}}^{+}$	$\langle x, y\rangle \in R^{\mathcal{I}} \wedge \langle y, z\rangle \in R^{\mathcal{I}} \implies \langle x, z\rangle \in R^{\mathcal{I}}$
$R \sqsubseteq S \in \mathcal{R}^{+}$	$\langle x, y\rangle \in R^{\mathcal{I}} \implies \langle x, y\rangle \in R^{\mathcal{I}}$

Table 2.2: Semantic constraints for $\mathcal{SHI}$ interpretations. Conditions introduced in $\mathcal{SHI}$ shown below the line

Definition 32 (Concept closure)**.** *Given a role hierarchy $\mathcal{R}$, the closure of a concept D with respect to $\mathcal{R}$ is* $\mathrm{cl}(D, \mathcal{R}) = \mathrm{sub}(D) \cup \{\doteq\neg C \mid C \in \mathrm{sub}(D)\} \cup \{\forall R.E \mid \forall S.E \in \mathrm{sub}(D)$ *or* $\doteq\forall S.E \in \mathrm{sub}(D)$ *and R occurs in $\mathcal{R}$ or $D\}$*

This notion is rather technical, its main purpose is to describe and quantify the number of concepts that may appear in the tableau, what is needed in order to prove the termination of the algorithm. According to Horrocks & Sattler (1999), a tableau for $\mathcal{SHI}$ is formally defined as follows.

Definition 33 ($\mathcal{SHI}$ tableau)**.** *Let D be an $\mathcal{SHI}$ concept in NNF and a role hierarchy $\mathcal{R}$. Let R_D be the set of roles occurring in* $\mathrm{cl}(D, \mathcal{R})$ *including all their inverses. A tableau T for D is a triple $(S, \mathcal{L}, \mathcal{E})$, such that S is a set of individuals, $\mathcal{L} : S \to 2^{\mathrm{cl}(D,\mathcal{R})}$ labels each individual of S with a set of concepts under* $\mathrm{cl}(D, \mathcal{R})$*, $\mathcal{E} : R_D \to 2^{S \times S}$ labels each role name in R_D with a set of edges, and there is some individual $s_0 \in S$ such that $D \in \mathcal{L}(s_0)$. Moreover, for each $C, C_1, C_2 \in \mathrm{cl}(D, \mathcal{R})$, for each $R, S \in R_D$, and for each $s, t \in S$ is holds that:*

1. *if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$;*
2. *if $C_1 \sqcap C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ and $C_2 \in \mathcal{L}(s)$;*
3. *if $C_1 \sqcup C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ or $C_2 \in \mathcal{L}(s)$;*
4. *if $\forall R.C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(R)$, then $C \in \mathcal{L}(t)$;*
5. *if $\exists R.C \in \mathcal{L}(s)$, then there also exists some $t \in S$ such that $\langle s, t\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(t)$;*
6. *if $\forall R.C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(S)$ for some $S \sqsubseteq R$ with* $\mathrm{Trans}(S)$*, then $\forall S.C \in \mathcal{L}(t)$;*
7. *$\langle s, t\rangle \in \mathcal{E}(R)$ if and only if $\langle t, s\rangle \in \mathcal{E}(R^{-})$;*
8. *if $\langle s, t\rangle \in \mathcal{E}(R)$ and $R \sqsubseteq S$, then $\langle s, t\rangle \in \mathcal{E}(S)$.*

Tableaux and $\mathcal{SHI}$ interpretations are interrelated, as stated by the following theorem. Hence, it is sufficient to find a tableau in order to prove the satisfiability of a $\mathcal{SHI}$ concept.

Theorem 30 (Horrocks & Sattler (1999)). *A $\mathcal{SHI}$ concept C is satisfiable if and only if a tableau for C exists.*

The tableaux algorithm for concept satisfiability checking works on a completion tree. It tries to construct a completion tree which is complete and clash-free. All these notions are defined exactly the same as for $\mathcal{ALC}$, with a minor change, that edges in the completion tree may now be labeled also with inverse roles.

What is new for $\mathcal{SHI}$ is the notion of R-neighbour. We can no longer rely on tree-successors only, because with inverse roles, the role relationships are now bidirectional in the completion tree. In addition we need to consider the role hierarchy.

Definition 34 (R-neighbour). *In a completion tree $(V, E, \mathcal{L})$ a node $x \in V$ is said to be an R-neighbour of another node $y \in V$ if, for some $S \sqsubseteq R$, either x is a successor of y and $\mathcal{L}(\langle y, x\rangle) = S$, or y is a successor of x and $\mathcal{L}(\langle x, y\rangle) = S^-$.*

$\mathcal{SHI}$ also requires much more sophisticated blocking policy. Due to the presence of inverse roles, subset blocking is insufficient, since role relationships in the tree are bidirectional. Accordingly, we only block if for some node $x \in V$ its ancestor $y \in V$ is found with the labels exactly matching. Moreover, to achieve termination, it is required to impose order on rule application. The (node) nongenerating rules are preferred to the (node) generating $\exists$-rule. This is achieved by so called dynamic blocking: nodes are allowed to become blocked and then become unblocked repeatedly several times during the rule application process. Also certain tableaux rules are allowed to be applied on nodes that are currently blocked and so the effect of precedence between rules is achieved. Blocking is formally defined as follows.

Definition 35 (Dynamic blocking). *Given a completion tree $(V, E, \mathcal{L})$, a node $x \in V$ is directly blocked, if it has a sole ancestor $y \in V$ such that $\mathcal{L}(x) = \mathcal{L}(y)$. A node $x \in V$ is indirectly blocked, if it has an ancestor that is blocked directly. A node $x \in V$ is blocked if it is directly or indirectly blocked.*

We are now ready to present the algorithm, as it was given by Horrocks & Sattler (1999). The tableaux algorithm for checking the satisfiability of $\mathcal{SHI}$ concepts uses a new set of tableaux expansion rules, listed in Fig. 2.3. Otherwise the algorithm works as in the case of $\mathcal{ALC}$.

Algorithm 4 (Satisfiability in $\mathcal{SHI}$). *Given an input concept C in NNF, the algorithm executes three steps:*

1. initialization: *create a new completion tree $(V, \emptyset, \mathcal{L})$ with a single node $s_0 \in V$ and initialize the label of s_0 to $\mathcal{L}(s_0) = C$;*
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.3. This step is over when none of the rules is applicable;*
3. answer: *if the completion tree is clash-free, answer "C is satisfiable". Otherwise, answer "C is unsatisfiable".*

The new $\forall_+$-rule is responsible for correct handling of transitive roles. This works as follows. In case when there are three nodes $s_1, s_2, s_3 \in V$ such that

$\sqcap$-rule
If $C_1 \sqcap C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(X)$,
and x is not indirectly blocked,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule
If $C_1 \sqcup C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \cap \mathcal{L}(X) = \emptyset$,
and x is not indirectly blocked,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\exists$-rule
If $\exists R.C \in \mathcal{L}(x)$, for $x \in V$ with no R-neighbour y s.t. $C \in \mathcal{L}(y)$,
and x is not blocked,
then add new node z to V with $\mathcal{L}(z) = \{C\}$ and $\mathcal{L}(\langle x, z\rangle) = \{R\}$.

$\forall$-rule
If $\forall R.C \in \mathcal{L}(x)$ for $x, y \in V$, y R-neighbour of x with $C \notin \mathcal{L}(y)$,
and x is not indirectly blocked,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.

$\forall_+$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x \in V$, and some $S \sqsubseteq R$ with $\mathrm{Trans}(S)$,
and x has an R-neighbour y s.t. $\forall S.C \notin \mathcal{L}(y)$,
and x is not indirectly blocked,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

Figure 2.3: $\mathcal{SHI}$ tableaux expansion rules

$R \in \mathcal{L}(\langle s_1, s_2\rangle)$, $R \in \mathcal{L}(\langle s_2, s_3\rangle)$, and $\forall R.C \in \mathcal{L}(s_1)$, where R is transitive, the transitivity of R implies that also the edge $\langle s_1, s_3\rangle$ should take part in R. It is not necessary, however, to insert a new edge between s_1 and s_3 into the completion tree, and to add R to $\mathcal{L}(\langle s_1, s_3\rangle)$. In this case, it is important to propagate the concept C into $\mathcal{L}(s_3)$. This is what the $\forall_+$-rule assures by propagating the value restriction $\forall R.C$ from $\mathcal{L}(s_1)$ to $\mathcal{L}(s_2)$.

The point of this workaround is that this way the precondition of the $\forall_+$-rule is local. That is, the rule applicability on $x \in V$ depends only on x and its neighbouring nodes. We do not need to search further in the completion tree in order to determine whether the rule is applicable or not.

Theorem 31 (Horrocks & Sattler (1999))**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{SHI}$ concepts always terminates, and it is sound and complete (i.e., it constructs a complete and clash-free completion tree if and only if the input concept is satisfiable).*

What is interesting, with TBox reasoning in $\mathcal{SHI}$ we get around without the $\mathcal{T}$-rule. Thanks to role transitivity and role inclusion axioms, deciding satisfiability of concepts with respect to a TBox is reducible to the case with no TBox present. And hence, the decision procedure for the basic case, as given above, suffices. This technique, that is formalized below, is called *internalization* of the TBox.

Internalization requires the *universal* role U, a special role that is transitive and is a superrole of any other roles (including inverses). Note, that it is easy to add the universal role to any role hierarchy, simply by adding the $R \sqsubseteq U$ and $R^- \sqsubseteq U$, for every role R already present in the hierarchy. Thanks to clever use of this role and the $\forall_+$-rule, all GCI in the TBox are propagated to each node in the completion tree, thus basically simulating the effect of the $\mathcal{T}$-rule.

Theorem 32 (Horrocks et al. (2000)). *Let $\mathcal{T}$ be a $\mathcal{SHI}$ TBox, C, D be $\mathcal{SHI}$ concepts. Let*

$$C_{\mathcal{T}} = \bigsqcap_{C_i \sqsubseteq D_i \in \mathcal{T}} \neg C_i \sqcup D_i \ .$$

Let U be a transitive role such that $R \sqsubseteq U$ and $R^- \sqsubseteq U$ for every role R occurring in $\mathcal{T}$, C and D. Then C is satisfiable with respect to $\mathcal{T}$ if and only if $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is satisfiable, and $C \sqsubseteq D$ with respect to $\mathcal{T}$ if and only if $C \sqcap \neg D \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is unsatisfiable.

For ABox consistency checking, the algorithm is adjusted exactly the same way as in $\mathcal{ALC}$. When checking ABox consistency with respect to the TBox, the TBox has to be internalized first.

2.2.3 Computational complexity

The complexity of reasoning is increased in $\mathcal{SHI}$, compared to the case of $\mathcal{ALC}$. Both, concept satisfiability and ABox consistency checking. Note that we prefer to reason with TBoxes internalized.

Theorem 33 (Horrocks & Sattler (1999)). *The problem of concept satisfiability checking is ExpTime-complete for the DL $\mathcal{SHI}$.*

Theorem 34 (Horrocks & Sattler (1999)). *The problem of ABox consistency checking is ExpTime-complete for the DL $\mathcal{SHI}$.*

The ExpTime-hardness is presumably caused by the allowance of role hierarchies. For $\mathcal{SI}$, the sub-language of $\mathcal{SHI}$ that does not allow role hierarchies, the concept satisfiability problem retains the same complexity as in the case of $\mathcal{ALC}$.

Theorem 35 (Horrocks et al. (1999)). *The problem of concept satisfiability checking PSpace-complete for the DL $\mathcal{SI}$.*

2.3 Number restrictions

The next family of DL constructors is called number restrictions. There are multiple constructors of this type, or more precisely, there are multiple versions of number restrictions a particular DL may be equipped with. The most basic idea is to allow to make selected roles *functional*, or possibly also conversely, *nonfunctional*. However, we do not want to assert this globally, but instead we wish to have a practical way to control this, based on the particular concept on which the role is being applied. Therefore we introduce two new constructors $\leqslant 1$ and $\geqslant 2$. Applied on a role R, $\leqslant 1\, R$ returns a concept representing

all individuals on which R behaves functionally (i.e., there are never two distinct R-neighbours of any instance of $\leqslant 1\, R$). The other form $\geqslant 2\, R$ is simply a complement of $\leqslant 1\, R$ (i.e., any instance thereof has at least two distinct R-neighbours). This may sound a bit cumbersome, but consider the example GCI axioms:

$$\mathsf{Person} \sqsubseteq \leqslant 1\, \mathsf{hasBilogicalFather} \ , \qquad \mathsf{Person} \sqsubseteq \geqslant 2\, \mathsf{hasBiologicalParent} \ .$$

Clearly, stating that one person has at most one biological father, and that it certainly has more than one biological parent, represents a very natural and intuitive modeling approach. There is no way to fix this kind of modeling in simpler DL, such as $\mathcal{ALC}$ or $\mathcal{SHI}$.

A generalization of these constructors is to allow any natural number $n \geq 0$ in place of 1 and 2. Such constructors are called (unqualified) *number restrictions*. The meaning is obvious, $\leqslant n\, R$ means at most n R-neighbours, and $\geqslant n\, R$ means at least n R-neighbours. With number restrictions we can make our modeling in the previous example more precise, by asserting both:

$$\mathsf{Person} \sqsubseteq \leqslant 2\, \mathsf{hasBiologicalParent} \ , \qquad \mathsf{Person} \sqsubseteq \geqslant 2\, \mathsf{hasBiologicalParent} \ ,$$

and thus fixing that every person has exactly two biological parents. To make things simpler, there is also the form $= n\, R$, which restricts to exactly n R-neighbours. With help of this constructor we may replace the two axioms above with $\mathsf{Person} \sqsubseteq = 2\, \mathsf{hasBiologicalParent}$, and the meaning is just the same. We will see later on, that this constructor is just some syntactic sugar.

Previously, we have been able to restrict the number of R-neighbours of some concept, but there was no possibility to say anything about these R-neighbours. This is finally possible with *qualified number restrictions*, which allow us to require (certain number of) R-neighbours belonging to some particular concept. There are three forms: $\leqslant n\, R.C$, $\geqslant n\, R.C$ and $= n\, R.C$, the last being again syntactic sugar. These constructors are useful, if we want to model with more universal roles, such as for instance $\mathsf{hasPart}$, and we want to *count* a certain number of compounds of particular type. Consider the following modeling about a concept of passenger cars:

$$\mathsf{Car} \sqsubseteq = 1\, \mathsf{hasPart.Engine} \ , \qquad \mathsf{Car} \sqsubseteq = 4\, \mathsf{hasPart.Wheel} \ ,$$
$$\mathsf{Car} \sqsubseteq \leqslant 8\, \mathsf{hasPart.Seat} \ , \qquad \mathsf{Car} \sqsubseteq \geqslant 2\, \mathsf{hasPart.Seat} \ .$$

These axioms will assure that every instance of Car must have exactly one $\mathsf{hasPart}$-neighbour in the concept *Engine*, four $\mathsf{hasPart}$-neighbours of type Wheel, and from two to eight $\mathsf{hasPart}$-neighbours of type Seat. Let us now take a look at these constructors more formally.

2.3.1 $\mathcal{SHIQ}$, $\mathcal{SHIN}$ and $\mathcal{SHIF}$

Different incarnations of number restrictions, as shown above, offer different expressive power. Extending $\mathcal{SHI}$ with some of these constructor thus possibly leads to $\mathcal{SHIF}$ (functionality), $\mathcal{SHIN}$ (unqualified number restrictions) and $\mathcal{SHIQ}$ (qualified number restrictions). We will now have a look at $\mathcal{SHIQ}$, the most general of these languages, the other two being contained in it. $\mathcal{SHIQ}$ has become particularly considerable after the introduction of a tableaux reasoning algorithm by Horrocks et al. (1999, 2000). Our survey is rooted in these works.

Definition 36 (Syntax of $\mathcal{SHIQ}$)**.** *Let N_{I}, N_{C}, and N_{R} be sets of names for individuals, concepts and roles respectively. Let ${N_{\mathrm{R}}}^{+} \subseteq N_{\mathrm{R}}$ be a set of names for transitive roles. The set of all $\mathcal{SHIQ}$ roles is $N_{\mathrm{R}} \cup \{R^{-} \mid R \in N_{\mathrm{R}}\}$. Let $R^{-} = S$, for any $R = S^{-}$, $S \in N_{\mathrm{R}}$. Let $\mathcal{R}$ be a set of RIA axioms of the form $R \sqsubseteq S$, where R and S are possibly inverse roles. Let $\mathcal{R}^{+} = (\mathcal{R} \cup \{R^{-} \sqsubseteq S^{-} \mid R \sqsubseteq S \in \mathcal{R}\}, \circledast)$, with $\circledast$ being the transitive-reflexive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{R^{-} \sqsubseteq S^{-} \mid R \sqsubseteq S \in \mathcal{R}\}$, be the corresponding role hierarchy. Furthermore, a role R is transitive (denoted by* $\mathrm{Trans}(R)$*) either if $R \in {N_{\mathrm{R}}}^{+}$ or if $R^{-} \in {N_{\mathrm{R}}}^{+}$. A role is simple,*[8] *if it neither is transitive nor it has a transitive subrole.*

A $\mathcal{SHIQ}$ concept is:

1. *any atomic concept $A \in N_{\mathrm{C}}$,*
2. *the complement $\neg C$,*
3. *the intersection of two concepts $C \sqcap D$,*
4. *the union of two concepts $C \sqcup D$,*
5. *the value restriction $\forall R.C$,*
6. *and the existential restriction $\exists R.C$,*
7. *a qualified number restriction $\leqslant n\, Q.C$,*
8. *a qualified number restriction $\geqslant n\, Q.C$,*

for any two $\mathcal{SHIQ}$ concepts C and D, any $\mathcal{SHIQ}$ role R, any simple $\mathcal{SHIQ}$ role Q, and a natural number $n \geq 0$.

A $\mathcal{SHIQ}$ TBox is a collection of GCI axiom of the form $C \sqsubseteq D$, for any $\mathcal{SHIQ}$ concepts C and D. A $\mathcal{SHIQ}$ ABox is a collection of ABox assertions of the form $i : C$ or $i, j : R$, for any $i, j \in N_{\mathrm{I}}$, for any $\mathcal{SHIQ}$ concept C, and for any $\mathcal{SHIQ}$ role R.

The two weaker types of number restrictions, the unqualified number restrictions and functionality, are now introduced as special cases.

Definition 37 ($\mathcal{SHIN}$)**.** *A restriction of $\mathcal{SHIQ}$, that only allows for number restrictions of the form $\leqslant n\, Q.\top$ and $\geqslant n\, Q.\top$, with Q a simple role and $n \geq 0$, is called $\mathcal{SHIN}$. We usually write $\leqslant n\, Q$ and $\geqslant n\, Q$ instead of $\leqslant n\, Q.\top$ and $\geqslant n\, Q.\top$ respectively.*

Definition 38 ($\mathcal{SHIF}$)**.** *A restriction of $\mathcal{SHIN}$, that only allows for number restrictions of the form $\leqslant 1\, Q$ and $\geqslant 2\, Q$, with any simple role Q, is called $\mathcal{SHIF}$.*

We also make use of syntactic shorthands such as $= n\, Q.C$, standing for $(\leqslant n\, Q.C) \sqcap (\geqslant n\, Q.C)$, and $= n\, Q$, standing for $(\leqslant n\, Q) \sqcap (\geqslant n\, Q)$. The semantics of $\mathcal{SHIQ}$ extends the $\mathcal{SHI}$ semantics with new constraints that ensure the intended meaning of number restrictions.

[8] The notion of simple roles is needed in order to guarantee decidability, as we will see below. The original notion of a simple role (Horrocks et al. 1999), that also we rely upon, is rather restrictive, no transitive roles or even roles with transitive subroles are allowed. Recent research shows that, to some extent, this restriction can be relaxed and decidability is maintained (Kazakov et al. 2007).

Definition 39 (Semantics of $\mathcal{SHIQ}$)**.** *Given some vocabulary N_{C}, N_{R}, N_{I} and a role hierarchy $\mathcal{R}^+$, a $\mathcal{SHIQ}$ interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, such that $\Delta^{\mathcal{I}} \neq \emptyset$ and the interpretation function $\cdot^{\mathcal{I}}$ maps every $i \in N_{\mathrm{I}}$ to some $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every $\mathcal{SHIQ}$ concept C to some $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every $\mathcal{SHIQ}$ role R to some $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that for any $\mathcal{SHIQ}$ concepts C, D and for any $\mathcal{SHIQ}$ roles R, S the conditions listed in Table 2.3 are satisfied.*

A $\mathcal{SHIQ}$ interpretation satisfies the elements of the language (denoted $\mathcal{I} \models \cdot$) as usual:

1. *$\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;*
2. *$\mathcal{I} \models i : C$ if $i^{\mathcal{I}} \in C^{\mathcal{I}}$;*
3. *$\mathcal{I} \models i, j : R$ if $\langle i^{\mathcal{I}}, j^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$;*
4. *$\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ is a model of $\mathcal{T}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{T}$;*
5. *$\mathcal{I} \models \mathcal{A}$ ($\mathcal{I}$ is a model of $\mathcal{A}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{A}$.*

Construct	Condition
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\forall y \in \Delta^{\mathcal{I}})\ \langle x, y \rangle \in R^{\mathcal{I}} \implies y \in C\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\exists y \in \Delta^{\mathcal{I}})\ \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C\}$
$R \in N_{\mathrm{R}}$	$\langle x, y \rangle \in R^{\mathcal{I}} \iff \langle y, x \rangle \in R^{-\mathcal{I}}$
$R \in {N_{\mathrm{R}}}^+$	$\langle x, y \rangle \in R^{\mathcal{I}} \wedge \langle y, z \rangle \in R^{\mathcal{I}} \implies \langle x, z \rangle \in R^{\mathcal{I}}$
$R \sqsubseteq S \in \mathcal{R}^+$	$\langle x, y \rangle \in R^{\mathcal{I}} \implies \langle x, y \rangle \in S^{\mathcal{I}}$
$\geqslant n\, R.C$	$(\geqslant n\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$
$\leqslant n\, R.C$	$(\leqslant n\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$

Table 2.3: Semantic constraints for $\mathcal{SHIQ}$ interpretations. Conditions, new in $\mathcal{SHIQ}$ listed below the line

The decision problems for $\mathcal{SHIQ}$ are defined exactly as they have been defined for $\mathcal{ALC}$. Also, all the reductions among the decision problems, as presented for $\mathcal{ALC}$, carry over to $\mathcal{SHIQ}$ directly.

Please note also, that existential restriction and value restriction constructors are now just syntactic sugar, and they could have been omitted. This is due to $\exists R.C \equiv \geqslant 1\, R.C$ and $\forall R.C \equiv \leqslant 0\, R.\neg C$.

2.3.2 Reasoning with $\mathcal{SHIQ}$

The tableaux algorithm for $\mathcal{SHIQ}$, given by Horrocks et al. (1999), is similar to the one for $\mathcal{SHI}$. It employs new tableaux rules, that take care of the new constructors, generating and merging nodes in the completion tree when necessary. The algorithm also uses a yet more sophisticated blocking strategy.

After introducing new concept constructors, we need to upgrade the definition of all (syntactic) subconcepts of a concept.

Definition 40. *Let D be a $\mathcal{SHIQ}$ concept. The subconcepts of D, the members of the set* $\mathrm{sub}(D)$*, are recursively defined as follows:*

1. $D \in \mathrm{sub}(D)$,
2. *if* $D = \neg C$, *then* $C \in \mathrm{sub}(D)$,
3. *if* $D = C_1 \sqcap C_2$, *then* $C_1 \in \mathrm{sub}(D)$ *and* $C_2 \in \mathrm{sub}(D)$,
4. *if* $D = C_1 \sqcup C_2$, *then* $C_1 \in \mathrm{sub}(D)$ *and* $C_2 \in \mathrm{sub}(D)$,
5. *if* $D = \forall R.C$, *then* $C \in \mathrm{sub}(D)$,
6. *if* $D = \exists R.C$, *then* $C \in \mathrm{sub}(D)$,
7. *if* $D = \leqslant n\, R.C$, *then* $C \in \mathrm{sub}(D)$,
8. *if* $D = \geqslant n\, R.C$, *then* $C \in \mathrm{sub}(D)$.

The concept closure is not changed, except for the fact that it now uses the redefined $\mathrm{sub}(\cdot)$ operator to compute the subconcepts. To obtain a tableau for a $\mathcal{SHIQ}$ concept, it is necessary to add three new constraints to the definition of a tableau for $\mathcal{SHI}$ concepts. These constraints will take care of number restrictions. According to Horrocks et al. (1999), a $\mathcal{SHIQ}$ tableau is defined as follows.

Definition 41 ($\mathcal{SHIQ}$ tableau)**.** *Let D be a $\mathcal{SHIQ}$ concept in NNF. Let $\mathcal{R}$ be a role hierarchy. Let R_D be a set of roles occurring in* $\mathrm{cl}(D, \mathcal{R})$ *including all their inverses. A tableau T for D with respect to $\mathcal{R}$ is a triple $(S, \mathcal{L}, \mathcal{E})$, such that S is a set of individuals, $\mathcal{L} : S \to 2^{\mathrm{cl}(D,\mathcal{R})}$ labels each individual of S with a set of concepts, $\mathcal{E} : R_D \to 2^{S \times S}$ labels each role name in R_D with a set of edges, and there is some individual $s_0 \in S$ such that $D \in \mathcal{L}(s_0)$. Moreover, for all $C, C_1, C_2 \in \mathrm{cl}(D, \mathcal{R})$, for all $R, S \in R_D$, and for all $s, t \in S$, is holds that:*

1. *if* $C \in \mathcal{L}(s)$, *then* $\neg C \notin \mathcal{L}(s)$;
2. *if* $C_1 \sqcap C_2 \in \mathcal{L}(s)$, *then* $C_1 \in \mathcal{L}(s)$ *and* $C_2 \in \mathcal{L}(s)$;
3. *if* $C_1 \sqcup C_2 \in \mathcal{L}(s)$, *then* $C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$;
4. *if* $\forall R.C \in \mathcal{L}(s)$ *and* $\langle s, t \rangle \in \mathcal{E}(R)$, *then* $C \in \mathcal{L}(t)$;
5. *if* $\exists R.C \in \mathcal{L}(s)$, *then there also exists some* $t \in S$ *such that* $\langle s, t \rangle \in \mathcal{E}(R)$ *and* $C \in \mathcal{L}(t)$;
6. *if* $\forall R.C \in \mathcal{L}(s)$ *and* $\langle s, t \rangle \in \mathcal{E}(S)$ *for some* $S \sqsubseteq R$ *with* $\mathrm{Trans}(S)$, *then* $\forall S.C \in \mathcal{L}(t)$;
7. $\langle s, t \rangle \in \mathcal{E}(R)$ *if and only if* $\langle t, s \rangle \in \mathcal{E}(R^-)$;
8. *if* $\langle s, t \rangle \in \mathcal{E}(R)$ *and* $R \sqsubseteq S$, *then* $\langle s, t \rangle \in \mathcal{E}(S)$;
9. *if* $\geqslant n\, R.C \in \mathcal{L}(s)$, *then* $\sharp\{t \in S \mid \langle s, t \rangle \in \mathcal{E}(R) \wedge C \in \mathcal{L}(t)\} \geq n$;
10. *if* $\leqslant n\, R.C \in \mathcal{L}(s)$, *then* $\sharp\{t \in S \mid \langle s, t \rangle \in \mathcal{E}(R) \wedge C \in \mathcal{L}(t)\} \leq n$;
11. *if* $\geqslant n\, R.C \in \mathcal{L}(s)$ *or* $\leqslant n\, R.C \in \mathcal{L}(s)$ *and* $\langle s, t \rangle \in \mathcal{E}(R)$, *then* $C \in \mathcal{L}(t)$ *or* $\doteq C \in \mathcal{L}(t)$.

Tableaux maintain strict relationship with $\mathcal{SHIQ}$ interpretations, as stated by the following theorem. It is therefore sufficient to find a tableau in order to prove the satisfiability of a $\mathcal{SHIQ}$ concept.

Theorem 36 (Horrocks et al. (1999)). *A $\mathcal{SHIQ}$ concept C is satisfiable if and only if a tableau for C exists.*

In order to deal with number restrictions, the algorithm will merge nodes in the completion tree, if necessary. We need to keep track on whether two nodes in the completion tree can be merged or not. For this reason, we extend the completion tree as follows.

Definition 42 (Completion tree). *Given a role hierarchy $\mathcal{R}$, a completion tree for a $\mathcal{SHIQ}$ concept D is a quadruple $(V, E, \mathcal{L}, \not\doteq)$, where (V, E) is a directed tree, $\not\doteq$ is an inequality relation on V, and $\mathcal{L}$ is a labeling function on $V \cup E$ such that each node $x \in V$ is labeled by $\mathcal{L}(x) \subseteq \mathrm{cl}(D, \mathcal{R})$ and each edge $\langle x, y\rangle \in E$ is labeled by a set of (possibly inverse) roles $\mathcal{L}(\langle x, y\rangle)$ occurring in $\mathrm{cl}(D, \mathcal{R})$.*

We use the same notion of R-neighbour as for $\mathcal{SHI}$. Number restrictions are possibly a new source of clashes.

Definition 43 (Clash). *A clash is said to occur in a node $x \in V$ of a completion tree $(V, E, \mathcal{L}, \not\doteq)$ if:*

- *either for some concept C, $\{C, \neg C\} \subseteq \mathcal{L}(x)$,*
- *or a concept of the form $\leqslant n\, R.C \in \mathcal{L}(x)$ and there are $n+1$ R-neighbours $x_1, \ldots, x_{n+1} \in V$ of x such that $C \in \mathcal{L}(x_i)$ for $1 \leq i \leq n+1$ and pairwise $x_i \not\doteq x_j$ for $1 \leq i < j \leq n+1$.*

Already in $\mathcal{SHIF}$, the finite model property is lost, as there are concepts without any finite model, as the example shows.

Example 6. *Consider the following $\mathcal{SHIF}$ concept:*

$$\neg C \sqcap \exists F^-.(C \sqcap \leqslant 1\, F) \sqcap \forall R^-.\exists F^-.(C \sqcap \leqslant 1\, F) \ ,$$

where R is a transitive superrole of F. Every model of this concept is an infinite F^--sequence, starting in some $\neg C$-instance $s_0 \in V$ followed by infinitely many F^--descendants $s_i \in V$, $i > 0$, all instances of C. It is not possible to terminate the sequence in a cycle: connecting the outgoing F^--edge from any s_i, $i > 2$ to some s_j, $0 < j < i$, violates the $\leqslant 1\, F$ restriction in s_j, since it already has its parent as an F-neighbour; on the other hand, connecting s_j to s_0 does not help too, since s_0 is an instance of $\neg C$ and we have to connect s_j with a C-instance.

Due to the fact, that finite models are not guaranteed, dynamic blocking is not enough (Horrocks & Sattler 1999). $\mathcal{SHIF}$ and more expressive logics require what is called *pairwise blocking*. Here, not just a single node, but a pair of nodes and a linking edge between them has to be found repeatedly, in order to block some node in the completion tree.

Definition 44 (Pairwise blocking). *Given a completion tree $(V, E, \mathcal{L}, \not\doteq)$, a node $x \in V$ is directly blocked, if none of its ancestors is directly blocked, and it has ancestors $x', y, y' \in V$ such that:*

1. *x is a successor of x' and y is a successor of y',*
2. *$\mathcal{L}(x) = \mathcal{L}(x')$ and $\mathcal{L}(y) = \mathcal{L}(y')$,*
3. *$\mathcal{L}((x, x')) = \mathcal{L}((y, y'))$.*

A node in the completion tree is indirectly blocked, if it has an ancestor that is blocked directly. A node is blocked if it is directly or indirectly blocked.

We are now ready to present the tableaux algorithm for $\mathcal{SHIQ}$, as it was given by Horrocks et al. (1999). It works similarly as the algorithm for $\mathcal{ALC}$ or $\mathcal{SHI}$, however, it uses a new set of tableaux expansion rules for $\mathcal{SHI}$, that are listed in Fig. 2.4.

Algorithm 5 (Satisfiability in $\mathcal{SHIQ}$)**.** *Given an input concept C in NNF, the algorithm executes three steps:*

1. initialization: *create a new completion tree $(\{s_0\}, \emptyset, \mathcal{L}, \emptyset)$ and set $\mathcal{L}(s_0) = C$;*
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.4. This step is over when none of the rules is applicable;*
3. answer: *if the completion tree is clash-free, answer "C is satisfiable". Otherwise, answer "C is unsatisfiable".*

There are three new tableaux expansion rules in $\mathcal{SHIQ}$. The $\geqslant$-rule is responsible for generating a sufficient number of R-neighbours in each node $x \in V$ with $\geqslant n\, R.C \in \mathcal{L}(x)$. The rule does not heed the existing R-neighbours, it simply generates n R-successors and puts C into their labels.[9] What is important, the rule also records in the completion tree that these nodes are distinct and they cannot be merged. This is done using the $\neq$ relation.

The $\leqslant$-rule works conversely. Whenever a node $x \in V$ with $\leqslant n\, R.C \in \mathcal{L}(x)$ and with $m > n$ R-successors of type C is found, the rule attempts to merge two of these R-successors. This is only possible, if there are two nodes y and z amongst them such that $y \neq z$ is not recorded by $\neq$. Now, the practical importance of the explicit $\neq$ relation should be clearly seen. It is important to guarantee the termination of the algorithm, otherwise the $\geqslant$-rule would generate R-successors and the $\leqslant$-rule would merge them in an endless loop, for any node with both $\leqslant n\, R.C$ and $\geqslant n+1\, R.C$ in the label.

The choose-rule is needed to detect the unsatisfiability of some concepts, for instance $(\geqslant 3\, R.A) \sqcap (\leqslant 1\, R.B) \sqcap (\leqslant 1\, R.\neg B)$. If this rule was not there, the algorithm would simply generate three R successors x, y and z of s_0, each labeled with A, such that $x \neq y$, $y \neq z$ and $x \neq z$. Afterwards, the algorithm returns "satisfiable", as there is no rule to apply. By guessing either B or $\neg B$ in the labels of x, y and z by the choose-rule, we end up in a clash. As no two of these nodes can be merged, because they were generated by the $\geqslant$-rule, the algorithm halts and returns "unsatisfiable", which is the correct answer.

[9] Such behaviour may be suboptimal, but it suffices in order to prove the soundness and completeness of the algorithm. Of course, for practical implementation we may want to generate only a necessary number of nodes, for sake of optimization.

$\sqcap$-rule
If $C_1 \sqcap C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(X)$,
and x is not indirectly blocked,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule
If $C_1 \sqcup C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \cap \mathcal{L}(X) = \emptyset$,
and x is not indirectly blocked,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\exists$-rule
If $\exists R.C \in \mathcal{L}(x)$, for $x \in V$ with no R-neighbour y s.t. $C \in \mathcal{L}(y)$,
and x is not blocked,
then add new node z to V with $\mathcal{L}(z) = \{C\}$ and $\mathcal{L}(\langle x, z\rangle) = \{R\}$.

$\forall$-rule
If $\forall R.C \in \mathcal{L}(x)$ for $x, y \in V$, y R-neighbour of x with $C \notin \mathcal{L}(y)$,
and x is not indirectly blocked,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.

$\forall_+$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x \in V$, and some $S \sqsubseteq R$ with $\mathrm{Trans}(S)$,
and x has an R-neighbour y s.t. $\forall S.C \notin \mathcal{L}(y)$,
and x is not indirectly blocked,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

choose-rule
If $\geqslant n\, R.C \in \mathcal{L}(x)$ or $\leqslant n\, R.C \in \mathcal{L}(x)$, for some $x \in V$,
and x has an R-neighbour y s.t. $\{C, \dot{\neg}C\} \cap \mathcal{L}(y) = \emptyset$,
and x is not indirectly blocked,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{\dot{\neg}C\}$.

$\geqslant$-rule
If $\geqslant n\, R.C \in \mathcal{L}(x)$, $x \in V$ has no n R-neighbours $y_1, \ldots, y_n$
s.t. $C \in \mathcal{L}(y_i)$, for $1 \leq i \leq n$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$,
and x is not blocked,
then create n new nodes $y_1, \ldots, y_n$ s.t. $C \in \mathcal{L}(y_i)$, $\mathcal{L}(\langle x, y_i\rangle) = \{R\}$
for $1 \leq i \leq n$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$.

$\leqslant$-rule
If $\leqslant n\, R.C \in \mathcal{L}(x)$, for some $x \in V$ not indirectly blocked,
and x has $l > n$ R-neighbours $y_1, \ldots, y_l$ s.t. $C \in \mathcal{L}(y_i)$, $1 \leq i \leq l$,
and $C \in \mathcal{L}(y)$, $C \in \mathcal{L}(z)$, for y, z R-neighbours of x
s.t. $y \neq z$ does not hold and y is not an ancestor of x,
then set $\mathcal{L}(z) = \mathcal{L}(z) \cup \mathcal{L}(y)$,
and if z is an ancestor of x set $\mathcal{L}((z, x)) = \mathcal{L}((z, x)) \cup \mathcal{L}(\langle x, y\rangle)^-$
else set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$,
and set $\mathcal{L}(\langle x, y\rangle) = \emptyset$,
and set $u \neq z$ for all $u \in V$ s.t. $u \neq y$.

Figure 2.4: $\mathcal{SHIQ}$ tableaux expansion rules

Theorem 37 (Horrocks et al. (1999)). *The tableaux algorithm for deciding the satisfiability of $\mathcal{SHIQ}$ concepts always terminates, and it is sound and complete (i.e., it constructs a complete and clash-free completion tree if and only if the input concept is satisfiable).*

The algorithm for ABox consistency checking only differs in the initialization step, and is obtained exactly as in the case of $\mathcal{ALC}$. The same algorithm is used also for decision tasks with respect to a TBox, thanks to internalization (Theorem 32 also holds for $\mathcal{SHIQ}$).

2.3.3 Computational complexity

In terms of worst-case complexity, $\mathcal{SHIQ}$ resides in the same class as $\mathcal{SHI}$. In practical applications, reasoning in $\mathcal{SHIQ}$ may be slower due to the new generating rule ($\geqslant$-rule) and due to the fact that further nondeterminism is introduced by the choose-rule.

Theorem 38 (Tobies (2001)). *The problem of concept satisfiability checking is ExpTime-complete for the DL $\mathcal{SHIQ}$.*

Theorem 39 (Tobies (2001)). *The problem of ABox consistency checking is ExpTime-complete for the DL $\mathcal{SHIQ}$.*

It is worth noting, that already number restrictions constitute a type of DL constructor that is on the verge of undecidability. Only simple roles are allowed in number restrictions, and this requirement is vital in order to guarantee the decidability of the language. Already $\mathcal{SHN}$ is undecidable, if the restriction is left out. For convenience, let us denote the extension of $\mathcal{SHN}$ that allows any roles in number restrictions as $\mathcal{SHN}^+$.

Theorem 40 (Horrocks et al. (2000)). *The concept satisfiability problem for $\mathcal{SHN}^+$ is undecidable.*

This has been proven by Horrocks et al. (2000) by reduction to the so called domino problem that is known to be undecidable.

2.4 Nominals

A nominal is a concept of the from $\{i\}$, where $i \in N_{\mathrm{I}}$. In other words, nominals allow us to bring an ABox individual into the TBox and construct a new concept encapsulating this individual and consecutively use it as any other concept. Semantically, the nominal $\{i\}$ is a concept that contains exactly one instance i. In addition, a more general form of nominals allows us to easily create enumerated concepts of the form $\{i_1, \ldots, i_n\}$, where $i_1, \ldots, i_n \in N_{\mathrm{I}}$, although, we will see that this is only some syntactic sugar. This is the most apparent practical application domain of nominals. For instance, if we have the countries of the world represented as individuals, such as $\mathsf{albania}, \ldots, \mathsf{zimbabwe} \in N_{\mathrm{I}}$, we may want to introduce concepts such as:

$$\mathsf{EuropeanUnion} \equiv \{\mathsf{austria}, \ldots, \mathsf{unitedKingdom}\} \ .$$

This may look fairly simple, but we will learn that adding nominals into $\mathcal{SHIQ}$ makes decision problems harder, and even introduces further modeling

options, for instance the possibility to limit the number of elements in the model to any given finite number, if only one for some reason wishes so.

2.4.1 $\mathcal{SHOIQ}$

In this section, we show how $\mathcal{SHIQ}$ is extended with nominals, thus ending up with the DL $\mathcal{SHOIQ}$.[10] $\mathcal{SHOIQ}$ has become especially considerable since a practical tableaux reasoning algorithm was introduced by Horrocks & Sattler (2005, 2007). Our review is based on these works.

Definition 45 (Syntax of $\mathcal{SHOIQ}$)**.** *Let N_{I}, N_{C}, and N_{R} be sets of names for individuals, concepts and roles respectively. Let ${N_{\mathrm{R}}}^{+} \subseteq N_{\mathrm{R}}$ be a set of names for transitive roles. The set of all $\mathcal{SHOIQ}$ roles is $N_{\mathrm{R}} \cup \{R^{-} \mid R \in N_{\mathrm{R}}\}$. Let $R^{-} = S$, for any $R = S^{-}$, $S \in N_{\mathrm{R}}$. Let $\mathcal{R}$ be a set of RIA axioms of the form $R \sqsubseteq S$, where R and S are possibly inverse roles. Let $\mathcal{R}^{+} = (\mathcal{R} \cup \{R^{-} \sqsubseteq S^{-} \mid R \sqsubseteq S \in \mathcal{R}\}, \sqsubseteq^{*})$, with $\sqsubseteq^{*}$ being the transitive-reflexive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{R^{-} \sqsubseteq S^{-} \mid R \sqsubseteq S \in \mathcal{R}\}$, be the corresponding role hierarchy. Furthermore, a role R is transitive (denoted by* $\mathrm{Trans}(R)$*) if either $R \in {N_{\mathrm{R}}}^{+}$ or $R^{-} \in {N_{\mathrm{R}}}^{+}$. A role is simple, if it neither is transitive nor it has a transitive subrole.*

A $\mathcal{SHOIQ}$ concept is:

1. *any atomic concept $A \in N_{\mathrm{C}}$,*
2. *the complement $\neg C$,*
3. *the intersection of two concepts $C \sqcap D$,*
4. *the union of two concepts $C \sqcup D$,*
5. *the value restriction $\forall R.C$,*
6. *and the existential restriction $\exists R.C$,*
7. *a qualified number restriction $\leqslant n\, Q.C$,*
8. *a qualified number restriction $\geqslant n\, Q.C$,*
9. *a nominal $\{i\}$,*

for any $i \in N_{\mathrm{I}}$, any two $\mathcal{SHOIQ}$ concepts C and D, any $\mathcal{SHOIQ}$ role R, any simple $\mathcal{SHOIQ}$ role Q, and any natural number $n \geq 0$.

A $\mathcal{SHOIQ}$ TBox is a collection of GCI axioms of the form $C \sqsubseteq D$, for any $\mathcal{SHOIQ}$ concepts C and D. A $\mathcal{SHOIQ}$ ABox is a collection of ABox assertions of the form $i : C$ or $i, j : R$, for any $i, j \in N_{\mathrm{I}}$, for any $\mathcal{SHOIQ}$ concept C, and for any $\mathcal{SHOIQ}$ role R.

The enumerated concept $\{i_1, \ldots, i_n\}$, where $i_1, \ldots, i_n \in N_{\mathrm{I}}$, is a syntactic shorthand standing for $\{i_1\} \sqcup \cdots \sqcup \{i_n\}$. Hence, this concept has exactly n instances. The semantics of $\mathcal{SHOIQ}$ extends the $\mathcal{SHIQ}$ semantics with the corresponding constraint for nominals.

[10] Also $\mathcal{SHOIN}$ and $\mathcal{SHOIF}$ are derived from $\mathcal{SHOIQ}$, by allowing only unqualified number restriction or functional restriction, in the respective order.

Definition 46 (Semantics of $\mathcal{SHOIQ}$)**.** *A $\mathcal{SHOIQ}$ interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is nonempty and the interpretation function $\cdot^{\mathcal{I}}$ maps every individual $i \in N_{\mathrm{I}}$ to some $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every $\mathcal{SHOIQ}$ concept C to some $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every $\mathcal{SHOIQ}$ role R to some $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. In addition, for any $\mathcal{SHOIQ}$ concepts C, D and any $\mathcal{SHOIQ}$ roles R, S the conditions listed in Table 2.4 are satisfied.*

A $\mathcal{SHOIQ}$ interpretation satisfies the elements of the language (denoted $\mathcal{I} \models \cdot$) as usual:

1. *$\mathcal{I} \models C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;*
2. *$\mathcal{I} \models i : C$ if $i^{\mathcal{I}} \in C^{\mathcal{I}}$;*
3. *$\mathcal{I} \models i, j : R$ if $\langle i^{\mathcal{I}}, j^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$;*
4. *$\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ is a model of $\mathcal{T}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{T}$;*
5. *$\mathcal{I} \models \mathcal{A}$ ($\mathcal{I}$ is a model of $\mathcal{A}$) if $\mathcal{I} \models \phi$, for all $\phi \in \mathcal{A}$.*

Construct	Condition
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\forall y \in \Delta^{\mathcal{I}})\ \langle x, y \rangle \in R^{\mathcal{I}} \implies y \in C\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (\exists y \in \Delta^{\mathcal{I}})\ \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C\}$
$R \in N_{\mathrm{R}}$	$\langle x, y \rangle \in R^{\mathcal{I}} \iff \langle y, x \rangle \in R^{-\mathcal{I}}$
$R \in {N_{\mathrm{R}}}^{+}$	$\langle x, y \rangle \in R^{\mathcal{I}} \wedge \langle y, z \rangle \in R^{\mathcal{I}} \implies \langle x, z \rangle \in R^{\mathcal{I}}$
$R \sqsubseteq S \in \mathcal{R}^{+}$	$\langle x, y \rangle \in R^{\mathcal{I}} \implies \langle x, y \rangle \in S^{\mathcal{I}}$
$\geqslant n\, R.C$	$(\geqslant n\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$
$\leqslant n\, R.C$	$(\leqslant n\, R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$
$\{i\}$	$\{i\}^{\mathcal{I}} = \{i^{\mathcal{I}}\}$

Table 2.4: Semantic constraints for $\mathcal{SHOIQ}$ interpretations. The single condition new in $\mathcal{SHOIQ}$ listed below the line

Again, all decision problems are formally defined exactly as it has been introduced before for the weaker DL. Also, all the corresponding reductions among the decision problems apply for $\mathcal{SHOIQ}$.

2.4.2 Reasoning with $\mathcal{SHOIQ}$.

In $\mathcal{SHOIQ}$ tableaux, the notions of subconcepts $\mathrm{sub}(D)$ and closure $\mathrm{cl}(D, \mathcal{R})$ are exactly as in $\mathcal{SHIQ}$. We did introduce a new concept constructor for nominals, however, the recursive definition of $\mathrm{sub}(D)$ is not affected, since this constructor does not work on top of concepts. According to (Horrocks & Sattler 2005), a $\mathcal{SHOIQ}$ tableau is defined as follows.

Definition 47 ($\mathcal{SHOIQ}$ tableau)**.** *Let D be an $\mathcal{SHOIQ}$ concept in NNF and $\mathcal{R}$ be role hierarchy. Let R_D be the set containing all roles occurring in* $\mathrm{cl}(D, \mathcal{R})$ *together with their inverses. A tableau T for D is a triple $(S, \mathcal{L}, \mathcal{E})$, such that S*

is a set of individuals, $\mathcal{L}: S \to 2^{\mathrm{cl}(D,\mathcal{R})}$ *labels each individual of* S *with a set of concepts under* $\mathrm{cl}(D,\mathcal{R})$*,* $\mathcal{E}: R_D \to 2^{S\times S}$ *labels each role name in* R_D *with a set of edges, and there is some individual* $s_0 \in S$ *such that* $D \in \mathcal{L}(s_0)$*. Moreover, for each* $s,t \in S$*,* $i \in N_{\mathrm{I}}$*,* $C, C_1, C_2 \in \mathrm{cl}(D,\mathcal{R})$ *and* $R, S \in R_D$ *is holds that:*

1. *if* $C \in \mathcal{L}(s)$*, then* $\neg C \notin \mathcal{L}(s)$*;*
2. *if* $C_1 \sqcap C_2 \in \mathcal{L}(s)$*, then* $C_1 \in \mathcal{L}(s)$ *and* $C_2 \in \mathcal{L}(s)$*;*
3. *if* $C_1 \sqcup C_2 \in \mathcal{L}(s)$*, then* $C_1 \in \mathcal{L}(s)$ *or* $C_2 \in \mathcal{L}(s)$*;*
4. *if* $\forall R.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(R)$*, then* $C \in \mathcal{L}(t)$*;*
5. *if* $\exists R.C \in \mathcal{L}(s)$*, then there also exists some* $t \in S$ *such that* $\langle s,t\rangle \in \mathcal{E}(R)$ *and* $C \in \mathcal{L}(t)$*;*
6. *if* $\forall R.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(S)$ *for some* $S \sqsubseteq R$ *with* $\mathrm{Trans}(S)$*, then* $\forall S.C \in \mathcal{L}(t)$*;*
7. $\langle s,t\rangle \in \mathcal{E}(R)$ *if and only if* $\langle t,s\rangle \in \mathcal{E}(R^-)$*;*
8. *if* $\langle s,t\rangle \in \mathcal{E}(R)$ *and* $R \sqsubseteq S$*, then* $\langle s,t\rangle \in \mathcal{E}(S)$*;*
9. *if* $\geqslant n\, R.C \in \mathcal{L}(s)$*, then* $\sharp\{t \in S \mid \langle s,t\rangle \in \mathcal{E}(R) \wedge C \in \mathcal{L}(t)\} \geq n$*;*
10. *if* $\leqslant n\, R.C \in \mathcal{L}(s)$*, then* $\sharp\{t \in S \mid \langle s,t\rangle \in \mathcal{E}(R) \wedge C \in \mathcal{L}(t)\} \leq n$*;*
11. *if* $\leqslant n\, R.C \in \mathcal{L}(s)$ *and* $\langle s,t\rangle \in \mathcal{E}(R)$*, then* $C \in \mathcal{L}(t)$ *or* $\dot{\neg} C \in \mathcal{L}(t)$*;*
12. *if* $\{i\} \in \mathcal{L}(s) \cap \mathcal{L}(t)$*, then* $s = t$*.*

Theorem 41 (Horrocks & Sattler (2005))**.** *Given a role hierarchy* $\mathcal{R}$*, a* $\mathcal{SHOIQ}$ *concept* D *in NNF is satisfiable if and only if there exists a tableau for* D*.*

As explained by (Horrocks & Sattler 2005, 2007), building a tableaux algorithm for $\mathcal{SHOIQ}$ is more complex than is it for $\mathcal{SHIQ}$, and one has to be careful in order to achieve termination. Some of the notions that we previously have used need to be specified more precisely. First, due to the loss of the finite model property, it no longer suffices to work with completion trees, we move to completion graphs. The main difference between completion trees previously used with $\mathcal{SHIQ}$ and the new completion graphs is that nodes can now be arbitrarily interconnected, including cycles. Plus, also the set of concepts that may possibly appear in the node labels is extended.

Definition 48 (Completion graph)**.** *Given a role hierarchy* $\mathcal{R}$*, a completion graph for a* $\mathcal{SHOIQ}$ *concept* D *is a quadruple* $(V, E, \mathcal{L}, \neq)$*, such that* (V, E) *is a directed graph,* $\neq$ *is an inequality relation on* V*, and* $\mathcal{L}$ *is a labeling function that labels each node* $x \in V$ *with a set* $\mathcal{L}(x) \subseteq \mathrm{cl}(D,\mathcal{R}) \cup \{\{i\} \mid i \in N_{\mathrm{I}}\} \cup \{\leqslant m\, R.C \mid \leqslant n\, R.C \in \mathrm{cl}(D,\mathcal{R}) \wedge m \leq n\}$*, and each edge* $\langle x,y\rangle$ *with a set* $\mathcal{L}(\langle x,y\rangle)$ *that is a subset of the set of all roles occurring in* $\mathrm{cl}(D,\mathcal{R})$ *and their inverses.*

There are now more possibilities to run into a clash, since each nominal may only be assigned to a single individual.

Definition 49 (Clash)**.** *A completion graph* $(V, E, \mathcal{L}, \neq)$ *contains a clash if:*

- *for some concept C, there is $x \in V$ such that $\{C, \neg C\} \subseteq \mathcal{L}(x)$;*
- *or for some $x \in V$, $\leqslant n\, R.C \in \mathcal{L}(x)$ and there are $n+1$ R-neighbours $x_1, \ldots, x_{n+1} \in V$ of x such that $C \in \mathcal{L}(x_i)$, for $1 \leq i \leq n+1$, and pairwise $x_i \neq\!\!\!\!\cdot\; x_j$, for $1 \leq i < j \leq n+1$;*
- *or there are two nodes $x, y \in V$ and $i \in N_{\mathrm{I}}$ such that $\{i\} \in \mathcal{L}(x) \cap \mathcal{L}(y)$ and $x \neq\!\!\!\!\cdot\; y$.*

Before we get to blocking, let us define some auxiliary terminology. The notions of nominal nodes and blockable nodes will be used by the blocking strategy. In addition, we distinguish different level of nominal nodes. This will be needed later, in order to implement a successful rule application strategy that terminates and it is correct.

Definition 50 (Blockable and nominal nodes)**.** *Let $(V, E, \mathcal{L}, \neq\!\!\!\!\cdot\;)$ be a completion graph for an input concept D, any node $x \in V$ is blockable, if there is no nominal in $\mathcal{L}(x)$. Each node $x \in V$ that contains at least one nominal in $\mathcal{L}(x)$ is a nominal node.*

If $\{i_1\}, \ldots, \{i_n\}$ are nominals appearing in the input concept D, then each node $x \in V$ with $\{i_i\} \in \mathcal{L}(x)$, for some $1 \leq i \leq n$, is of level 0. Otherwise, a nominal node is of level j, if it is not of level $k < j$ and it has a nominal neighbour that is of level $j-1$.

We will use pairwise blocking, but we will not be able to block in the presence of nominals. Recall that each completion graph, once completed and clash-free, is a finite representation of a possibly infinite interpretation. The parts between the blocking node and the blocked node are repeated in the interpretation infinitely many times. For this reason nominals are not allowed here.

Definition 51 (Pairwise blocking in presence of nominals)**.** *Given a completion graph $(V, E, \mathcal{L}, \neq\!\!\!\!\cdot\;)$, a node $x \in V$ is directly blocked, if none of its ancestors is directly blocked, and it has ancestors $x', y, y' \in V$ such that:*

1. *x is a successor of x' and y is a successor of y',*
2. *x, y and all the nodes on the path from x to y are blockable,*
3. *$\mathcal{L}(x) = \mathcal{L}(x')$ and $\mathcal{L}(y) = \mathcal{L}(y')$,*
4. *$\mathcal{L}((x, x')) = \mathcal{L}((y, y'))$.*

In this case we say that y blocks x. A node in the completion graph is indirectly blocked, if it has an ancestor that is blocked directly. A node in the completion graph is blocked if it is directly or indirectly blocked.

In addition, an R-neighbour y of a node x is called safe either if x is blockable or if x is a nominal node and y is not blocked.

Some of the tableaux rules merge two nodes into one. We need to specify merging more precisely for $\mathcal{SHOIQ}$. Merging also uses an auxiliary operation called pruning.

Definition 52 (Pruning)**.** *Pruning a node y (invoked by* Prune(y)*) yields a completion graph obtained from the source completion graph $(V, E, \mathcal{L}, \neq\!\!\!\!\cdot\;)$ as follows:*

1. *For all successors z of y, remove the edge $\langle y, z\rangle$ from E and, if z is blockable, then* Prune(z).
2. *Remove y from V.*

Definition 53 (Merging)**.** *Merging a node y into a node x (invoked by calling* Merge(y, x)*) yields a completion graph obtained from the source completion graph $(V, E, \mathcal{L}, \neq)$ as follows:*

1. *For all nodes z such that $(z, y) \in E$*
   (a) *if $\{\langle x, z\rangle, (z, x)\} \cap E = \emptyset$, then add (z, x) to E and set $\mathcal{L}((z, x)) = \mathcal{L}((z, y))$,*
   (b) *if $(z, x) \in E$, then set $\mathcal{L}((z, x)) = \mathcal{L}((z, x)) \cup \mathcal{L}((z, y))$,*
   (c) *if $\langle x, z\rangle \in E$, then set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}((z, y))^{-}$, and*
   (d) *remove (z, y) from E.*
2. *For all nominal nodes z such that $\langle y, z\rangle \in E$*
   (a) *if $\{\langle x, z\rangle, (z, x)\} \cap E = \emptyset$, then add $\langle x, z\rangle$ to E and set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle y, z\rangle)$,*
   (b) *if $\langle x, z\rangle \in E$, then set $\mathcal{L}(\langle x, z\rangle) = \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle y, z\rangle)$,*
   (c) *if $(z, x) \in E$, then set $\mathcal{L}((z, x)) = \mathcal{L}((z, x)) \cup \mathcal{L}(\langle y, z\rangle)^{-}$, and*
   (d) *remove $\langle y, z\rangle$ from E.*
3. *Set $\mathcal{L}(x) = \mathcal{L}(x) \cup \mathcal{L}(y)$.*
4. *Add $x \neq z$ for all z such that $y \neq z$.*
5. *Invoke* Prune(y).

The tableaux expansion rules for $\mathcal{SHOIQ}$ are listed in Fig. 2.5 and 2.6. The $\geqslant$-rule, the $\exists$-rule and the NN-rule are generating rules (they add new nodes to the completion graph). The $\leqslant$-rule, the $\leqslant_o$-rule and the o-rule are shrinking (they merge nodes). Correct order of rule application is vital, in order to achieve termination.

Definition 54 (Rule application strategy)**.** *The expansion rules are applied on the completion graph according the following strategy:*

1. *the o-rule is applied with the highest priority;*
2. *next, the $\leqslant_o$-rule and the NN-rule are applied, and they are first applied to nominal nodes of lower level, before they are applied to nodes with higher levels. In case when these rules are both applicable to the same node, the NN-rule is applied first;*
3. *all other expansion rules are applied with lower priority;*

This strategy is necessary in order to guarantee termination. We are now ready to present the algorithm, as it was given by Horrocks & Sattler (2005, 2007). We will see that besides for the initialization step, in which we have to initialize some additional nodes for all nominals present in the input concept, and the new set of tableaux rules, the algorithm is essentially the same as in case of $\mathcal{ALC}$, $\mathcal{SHI}$ or $\mathcal{SHIQ}$.

$\sqcap$-rule
If $C_1 \sqcap C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(X)$,
then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule
If $C_1 \sqcup C_2 \in \mathcal{L}(x)$ for some $x \in V$ and $\{C_1, C_2\} \cap \mathcal{L}(X) = \emptyset$,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_1\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\exists$-rule
If $\exists R.C \in \mathcal{L}(x)$, for some $x \in V$, x is not blocked,
and there is no safe R-neighbour y s.t. $C \in \mathcal{L}(y)$,
then add new node z to V with $\mathcal{L}(z) = \{C\}$ and $\mathcal{L}(\langle x, z\rangle) = \{R\}$.

$\forall$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x \in V$,
and x has an R-neighbour y with $C \notin \mathcal{L}(y)$,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.

$\forall_+$-rule
If $\forall R.C \in \mathcal{L}(x)$ for some $x \in V$,
and there is S s.t. $S \boxplus R$ and $\mathrm{Trans}(S)$,
and x has an R-neighbour y with $\forall S.C \notin \mathcal{L}(y)$,
then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

choose-rule
If $\leqslant n\, R.C \in \mathcal{L}(x)$, for some $x \in V$,
and x has an R-neighbour y s.t. $\{C, \dot{\neg}C\} \cap \mathcal{L}(y) = \emptyset$,
then either set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C\}$ or set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{\dot{\neg}C\}$.

$\geqslant$-rule
If $\geqslant n\, R.C \in \mathcal{L}(x)$ for some $x \in V$, x is not blocked,
and there are not n safe R-neighbours $y_1, \ldots, y_n$ of x
s.t. $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$,
then create n new nodes $y_1, \ldots, y_n$ s.t. $C \in \mathcal{L}(y_i)$, $\mathcal{L}(\langle x, y_i\rangle) = R$
and set $y_i \neq y_j$ for $1 \leq i < j \leq n$.

$\leqslant$-rule
If $\leqslant n\, R.C \in \mathcal{L}(x)$, for some $x \in V$,
and x has $l > n$ R-neighbours $y_1, \ldots, y_l$ s.t. $C \in \mathcal{L}(i)$, $1 \leq i \leq l$,
and for any i, j, $1 \leq i < j \leq l$, $y_i \neq y_j$ is not true,
then if y_i is a nominal node then $\mathrm{Merge}(y_j, y_i)$,
else if y_j is nominal or an ancestor of y_i, $\mathrm{Merge}(y_i, y_j)$,
else $\mathrm{Merge}(y_j, y_i)$.

Figure 2.5: $\mathcal{SHOIQ}$ tableaux expansion rules (part 1)

o-rule
If $\{i\} \in \mathcal{L}(x) \cap \mathcal{L}(y)$, for $i \in N_{\mathrm{I}}$, $x, y \in V$ s.t. not $x \neq y$,
then if y is an initial node, $\mathrm{Merge}(x, y)$,
else $\mathrm{Merge}(y, x)$.

NN-rule
If $\leqslant n\, R.C \in \mathcal{L}(x)$, x is a nominal node with a blockable
R-predecessor y s.t. $C \in \mathcal{L}(y)$,
and there is no $\leqslant m\, R.C \in \mathcal{L}(x)$ with $m \leq n$,
s.t. x has m nominal R-neighbours $z_1, \ldots, z_m$
with $C \in \mathcal{L}(z_i)$ and $z_i \neq z_j$ for all $1 \leq i < j \leq m$,
then guess m with $1 \leq m \leq n$ and set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{\leqslant m\, R.C\}$,
and create m new nodes $y_1, \ldots, y_m$ with $\mathcal{L}(\langle x, y_i \rangle) = \{R\}$,
$\mathcal{L}(y_i) = \{C, \{i_i\}\}$, each $i_i \in N_{\mathrm{I}}$ new for $1 \leq i \leq m$,
and set $y_i \neq y_j$ for $1 \leq i < j \leq m$.

$\leqslant_o$-rule
If $\leqslant m\, R.C \in \mathcal{L}(x)$, x is a nominal node with a blockable
R-neighbour y s.t. $C \in \mathcal{L}(y)$,
and x has m nominal R-neighbours $z_1, \ldots, z_m$
with $C \in \mathcal{L}(z_i)$ and $z_i \neq z_j$ for all $1 \leq i < j \leq m$,
and x has a nominal R-neighbour z with $C \in \mathcal{L}(z)$ and not $y \neq z$,
then $\mathrm{Merge}(y, z)$.

Figure 2.6: $\mathcal{SHOIQ}$ tableaux expansion rules (part 2)

Algorithm 6 (Satisfiability in $\mathcal{SHOIQ}$)**.** *Let C be the input in NNF and let $\{i_1\}$, ..., $\{i_n\}$ be the nominals occurring in C. The algorithm executes three steps:*

1. initialization: *create a new completion graph $(\{s_0, s_1, \ldots, s_n\}, \emptyset, \mathcal{L}, \emptyset)$ with $\mathcal{L}(s_0) = \{D\}$, and set $\mathcal{L}(s_j) = \{\{i_j\}\}$ for $1 \leq j \leq n$ (nodes $s_0, \ldots, s_n$ are called initial nodes);*

2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.5 and 2.6 in accordance with the rule application strategy (Definition 54). This step is over when none of the rules is applicable;*

3. answer: *if the completion graph is clash-free, answer "C is satisfiable". Otherwise, answer "C is unsatisfiable".*

The first eight tableaux rules, as listed in Fig. 2.5, are very similar to the tableaux rules used by the $\mathcal{SHIQ}$ algorithm (Fig. 2.4). We now call $\mathrm{Merge}(x, y)$ and $\mathrm{Prune}(x)$ to implement the merging. A notable change is in the $\exists$-rule, which is now applicable if no *safe* neighbour of the desired type is found. This is due to the fact that unsafe neighbours (which are possibly blocked nodes) may not correspond to individuals of the infinite interpretation corresponding to the completion graph.

In Fig. 2.6 three new tableaux rules that deal with nominals. The o-rule assures that whenever two nodes with the same nominal in their labels are generated, they are immediately merged. Recall, that each nominal may only have

one instance, hence this is necessary. The remaining two rules can be viewed as auxiliary and they are needed to guarantee correctness and termination. Please refer to the work of Horrocks & Sattler (2007) for details.

Theorem 42 (Horrocks & Sattler (2005, 2007))**.** *The tableaux algorithm for deciding the satisfiability of $\mathcal{SHOIQ}$ concepts always terminates, and it is sound and complete (i.e., it constructs a complete and clash-free completion graph if and only if the input concept is satisfiable).*

For TBox reasoning, internalization is used precisely as in the case of $\mathcal{SHI}$ and $\mathcal{SHIQ}$. For ABox reasoning, to take an ABox into account, the completion graph is initialized to reflect all assertions in the ABox, similarly as in the case of $\mathcal{ALC}$, before the expansion rules are applied.

2.4.3 Computational complexity

The computational complexity of $\mathcal{SHOIQ}$ jumps to NExpTime. This is likely to be due to the loss of the tree model property, and due to interaction between nominals, number restrictions and inverse roles, as explained by Horrocks & Sattler (2007).

Theorem 43 (Tobies (2000))**.** *The problems of concept satisfiability checking and ABox consistency checking are both NExpTime-hard for $\mathcal{SHOIQ}$.*

2.5 More Expressive DL

Even if we have got to know a fairly rich family of DL, more are known in the literature. In this section, we briefly review $\mathcal{SROIQ}$ and $\mathcal{ALCQIb}$, two additional DL which add further expressive features. Later on in this thesis, we will work with multiple distributed ontology frameworks, that all feature a knowledge base build from multiple DL knowledge bases. Occasionally, some of these frameworks make use of richer DL constructors not contained in $\mathcal{SHOIQ}$. However, we will not go too much into detail, as these logics are marginally related to the results presented in this thesis.

2.5.1 $\mathcal{SROIQ}$

Introduced by Horrocks et al. (2006), $\mathcal{SROIQ}$ is a DL with rising significance. This is due to the fact, that the recently standardized newest version of the OWL Web Ontology Language, OWL 2 (W3C OWL Working Group 2009), derives its semantics from this DL. Therefore, $\mathcal{SROIQ}$ is of some importance for the distributed ontology formalisms, at least for those which find important motivation in the Semantic Web. However, $\mathcal{SROIQ}$ is a fairly complex DL, and thus its practical application in the distributed setting is yet a completely open issue.

With $\mathcal{SROIQ}$, Horrocks et al. aim at improving the options that the language provides for modeling with roles. This is done by enriching $\mathcal{SHOIQ}$ with the following constructs:

1. *reflexive*, *irreflexive* and *symmetric* roles may be declared. This is done with new role axioms $\mathrm{Ref}(R)$, $\mathrm{Irr}(R)$ and $\mathrm{Sym}(R)$, in the respective order.

In addition, transitive roles are now also declared by an axiom $\mathrm{Tra}(R)$. The semantics of these axioms is obvious, $R^{\mathcal{I}}$ constraint to be a reflexive, irreflexive, symmetric, or transitive relation;

2. two roles R and S may now also be declared *disjoint* with an axiom of the form $\mathrm{Dis}(R, S)$. As a consequence $R^{\mathcal{I}}$ and $s^{\mathcal{I}}$ are made disjoint in every model. Consider roles such as $\mathsf{isPartOf}$ and $\mathsf{hasPart}$. It makes sense to declare them disjoint. According to Horrocks et al. (2006), this may possibly yield new derivations between concepts in the TBox;

3. *universal role* U, a role counterpart of the top concept $\top$ is now part of the language. This "mother of all roles" subsumes every other role. Its interpretation is always equal to $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, the set of all pairs over $\Delta^{\mathcal{I}}$.

4. restricted form of *role composition* is now allowed in so called *complex role inclusion axioms*. The basic form of these axioms is $R \circ S \sqsubseteq R$ or $S \circ R \sqsubseteq R$, for two roles R and S. More complex forms allow arbitrarily long composition chains, always on the left hand side only, for instance $S_1 \circ \cdots \circ S_n \sqsubseteq R$. However, the composition chains must comply to further restrictions, as it is known that unrestricted role composition leads to undecidability (Baader & Sattler 1996). Role composition $R \circ S$ is interpreted by the composition of binary relations $R^{\mathcal{I}} \circ S^{\mathcal{I}}$. Complex role inclusions are a particularly handy feature, allowing us to express an "extended reachability" of selected roles. Consider, for instance, the simple observation: "Owning an object implies owning any part thereof." This is naturally modeled by $\mathsf{owns} \circ \mathsf{hasPart} \sqsubseteq \mathsf{owns}$.

5. *negated role assertions* allow us to express in the ABox that a particular pair of individuals is not related by a role. For instance, the assertion $\mathsf{rickDecard}, \mathsf{royBatty} : \neg\mathsf{likes}$ implies that $\left\langle \mathsf{rickDecard}^{\mathcal{I}}, \mathsf{royBatty}^{\mathcal{I}} \right\rangle \notin \mathsf{likes}^{\mathcal{I}}$.

6. finally, a new concept constructor of the form $\exists R.\mathrm{Self}$, allowing to express *local reflexivity*. Semantically, $\exists R.\mathrm{Self}^{\mathcal{I}}$ is a set of objects having itself as an R-neighbour (i.e., $\exists R.\mathrm{Self}^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \langle x, x \rangle \in R^{\mathcal{I}}\}$.) This might not seem a very useful constructor at the first glance, however, consider that having a transitive role $\mathsf{hasPart}$, we want to forbid anything becoming a part of itself, because this is almost certainly a modeling error. We just have to add the axiom $\exists\mathsf{hasPart}.\mathrm{Self} \sqsubseteq \bot$ and we are done.

A tableaux algorithm for $\mathcal{SROIQ}$ was given by Horrocks et al. (2006). Complexity of reasoning is, however, harder than in $\mathcal{SHOIQ}$. As shown by Kazakov (2008), $\mathcal{SROIQ}$ is 2NExpTime-hard, and even its sublanguage $\mathcal{RIQ}$ (Horrocks & Sattler 2004) is harder than $\mathcal{SHOIQ}$, being 2ExpTime-hard.

2.5.2 $\mathcal{ALCQIb}$

Introduced by Tobies (2001), $\mathcal{ALCQIb}$ extends $\mathcal{ALCQI}$ with selected role constructors. These include:

1. *role complement* ($\neg$), applied on a role, $\neg R$ is a complement of $R^{\mathcal{I}}$ with respect to $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. It can be seen as a more powerful variant of the role disjointness axiom of $\mathcal{SROIQ}$;

2. *role union* ($\sqcup$). Semantically, $(R \cup S)^{\mathcal{I}}$ equals to $R^{\mathcal{I}} \sqcup S^{\mathcal{I}}$. In any language that offers role hierarchies, this can be very approximatively emulated by a common superrole of R and S;

3. *role intersection* ($\sqcap$), with semantics analogous to concept intersection, that is $(R \sqcap S)^{\mathcal{I}}$ always equals to $R^{\mathcal{I}} \cap S^{\mathcal{I}}$.

With these constructors plus the inverse role constructor, complex roles may be constructed, which are then possibly used in existential restrictions, value restrictions and number restrictions. Note, that $\mathcal{ALCQI}b$ does not feature role hierarchies which would allow us to create subroles and defined roles. In addition, the complex roles are restricted to so called safe expressions, due to complexity issues (please refer to Tobies (2001)).

Satisfiability of concepts in PSpace-hard in $\mathcal{ALCQI}b$, however deciding satisfiability with respect to a knowledge base jumps into ExpTime-hard problems. A tableaux algorithm concept satisfiability is also given by Tobies (2001). TBoxes internalization is not known for this, and Tobies (2001) offers an automata-based decision procedure for deciding satisfiability with respect to a knowledge base. To our best knowledge, a practical implementation is not known.

2.6 Summary

We have surveyed a family of DL, starting with $\mathcal{ALC}$, and consecutively adding more expressive features, reaching $\mathcal{SHI}$, $\mathcal{SHIQ}$ and $\mathcal{SHOIQ}$. We have discussed their syntax, semantics and also the tableaux reasoning algorithms which constitute the key to the practical applicability of these languages. Later on, we will discuss reasoning with various distributed ontology formalisms, which build on top of reasoning algorithms of these DL. We have also briefly reviewed two further expressive DL $\mathcal{SROIQ}$ and $\mathcal{ALCQI}b$, which provide additional expressive power that is also sometimes used in the context of distributed ontologies.

DL are particularly well suited for representation of ontologies, and they are relevant for the Semantic Web, as the semantics of OWL is derived from them. DL possibly offer rich expressive features, and different logics in the family provide a trade-off between expressivity and complexity of reasoning. Although for more expressive DL, the core decision problems are in ExpTime, or even NExpTime, practical implementations show that they are fairly usable even for moderately large datasets. Considerable research efforts are currently directed towards the optimization of the reasoners, and hence the practical applicability of these languages will hopefully further improve in the near future (Horrocks et al. 2000, Hudek & Weddell 2006, Fokoue et al. 2006, Ding & Haarslev 2006, Möller et al. 2006, Zuo & Haarslev 2006, Sirin & Parsia 2006).

Chapter 3

Distributed Description Logics

In the previous chapter we have encountered ontologies formally represented by logical theories in DL. We have learned that the DL family is wide, each language offering distinct features with particular purpose. One feature which all these languages share is that they always treat ontologies as monolithic structures. A DL theory is a set of axioms without any further structure. While this is a common assumption in mainstream ontology research, there are many reasons, why it should be loosened. In Sect. 1.2 we have cited two main motivating factors: introduction of modularity into the ontology engineering process and knowledge reuse by combination of existing ontologies. While there are more formalisms aimed towards overcoming this limitation of DL, the main attention of this thesis lies with Distribution Description Logics, on which also this chapter is focused. The other approaches are reviewed later on in Chap. 6.

Distributed Description Logics (DDL) constitute a formal representation framework that builds on top of DL, that has been introduced by Borgida & Serafini (2002). DDL follow the mapping approach. Multiple ontology units are allowed, which are combined by means of semantic mapping. The mapping is expressed with novel axioms called bridge rules. In the original proposal (Borgida & Serafini 2002, 2003) bridge rules allow to express semantic association between concepts from different ontology units. In addition, bridge rules are directed: the mapping is always between a source and a target ontology unit. The target ontology unit reuses knowledge from the source one. This is particularly important, as we do not wish the source ontology unit to be affected only because its knowledge is being reused somewhere elsewhere. Of course, if we wish so, we may express the mapping also in the opposite direction.

DDL allow to reuse knowledge of existing ontologies and they also open the doors to modular ontology engineering. This is also true for the other related frameworks, however we are fond of DDL because they are particularly well aligned with certain assumptions of the Semantic Web vision. DDL allow us to deal with partially overlapping domains of the ontology units that are to be combined. What is more important, sometimes these ontology units may also contain certain amount of contradictory information, or particular aspects of knowledge may be modeled in a heterogeneous way. While some other dis-

tributed and modular ontology frameworks simply rule out such scenarios at the beginning, DDL allow certain level of knowledge reuse also in such cases.

To illustrate an example of heterogeneous modeling we borrow a scenario from (Borgida & Serafini 2003). In this case we deal with similar knowledge that we wish to integrate and reuse, but things are modeled on different level of abstraction in distinct ontology units. Consider two information sources, one contains information about credit card purchases related to persons, and the other one contains census data that only records information about households. It is not apparent how to associate persons with households as they have different properties, yet one may wish to integrate these two information sources in some way. For instance we may wish to enrich the census data with approximative records of consumption per household.

In this chapter, we concentrate on the basic DDL framework, as it has been developed by Borgida & Serafini (2002, 2003), Serafini & Tamilin (2004), Serafini et al. (2005), and Tamilin (2007). This version of DDL allow bridge rules between concepts. In the following two chapters, we address some open issues related to DDL, and build on this version of the framework. DDL have been further enriched with bridge rules between individuals, bridge rules between roles, and even with so called heterogeneous bridge rules that represent mapping between concepts and roles (Serafini & Tamilin 2006, 2007, Tamilin 2007, Ghidini & Serafini 2006*a*,*b*, Ghidini et al. 2007). We only briefly mention these extensions in the final part of this chapter. We revisit more formally later on in Chap. 6.

3.1 Formalization

A DDL knowledge base is formed by an indexed set of ontology units, often called local ontologies. Each local ontology may use its own local language, however, we assume that this language is always a sublanguage of $\mathcal{SHIQ}$. In addition, there is a set of bridge rules which express semantic mapping between concepts from different ontologies. Bridge rules are directed and there are two basic types:

into-bridge rules (denoted using the symbol $\xrightarrow{\sqsubseteq}$): stating that the source concept corresponds to a subconcept of the target concept;

onto-bridge rules (denoted with the symbol $\xrightarrow{\sqsupseteq}$): stating that the source concept is a superconcept of the target concept.

By combination of an into- and an onto-bridge rule it is possible to state that the source concept corresponds to the target concept exactly. Hence the combined equivalence bridge rules (using the symbol $\xrightarrow{\equiv}$) are only introduced as syntactic sugar.

The local ontologies possibly contain GCI axioms, ABox expressions and also RIA axioms, and hence are in fact composed of a TBox, ABox and an RBox. We will however stick to the terminology introduced by (Borgida & Serafini 2002) and we will often refer to a local ontology as *local TBox* and to a DDL knowledge base as *distributed TBox*. It is to be stressed however, and the reader is kindly asked to take this into account, that ABox expressions and RIA axioms are possibly present. This abuse of terminology is justified by the fact

that we mostly concentrate on relations between concepts, since bridge rules are only expressed between concepts. Notation is simplified this way. Later on in Chap. 6 we will introduce more fine grained notation in order to compare DDL with other distributed ontology frameworks.

Definition 55 (Syntax of DDL)**.** *Assume a non-empty index set I, a family of mutually disjoint sets of atomic concept names $N_{\mathrm{C}} = \{N_{\mathrm{C}i}\}_{i\in I}$, a family of mutually disjoint sets of atomic role names $N_{\mathrm{R}} = \{N_{\mathrm{R}i}\}_{i\in I}$ and a family of mutually disjoint sets of individual names $N_{\mathrm{I}} = \{N_{\mathrm{I}i}\}_{i\in I}$. A DDL knowledge base is constructed as follows:*

1. *a local TBox $\mathcal{T}_i$, $i \in I$, is a DL knowledge base in some sublanguage of $\mathcal{SHIQ}$ (i.e., it possibly contains GCI, RIA, transitivity and ABox axioms);*

2. *a bridge rule from i to j is an expression of one of the two forms*

$$i : C \xrightarrow{\sqsubseteq} j : G \ , \qquad\qquad i : C \xrightarrow{\sqsupseteq} j : G \ ,$$

   where $i, j \in I$, $i \neq j$, and C, G are concepts in the local language used by the local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ respectively (the form on the left called into-bridge rule, the form on the right called onto-bridge rule);

3. *$\mathfrak{B}$ denotes a set of bridge rules which decomposes into $\mathfrak{B} = \bigcup_{i,j\in I, i\neq j} \mathfrak{B}_{ij}$, such that each $\mathfrak{B}_{ij}$ contains only bridge rules from i to j;*

4. *a distributed TBox over I is a pair of the form $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$.*

Given a DDL knowledge base $\mathfrak{T}$, if a concept C is constructed in the language of some local TBox $\mathcal{T}_i$ of $\mathfrak{T}$, using the atomic symbols from $N_{\mathrm{C}i}$, $N_{\mathrm{R}i}$ and $N_{\mathrm{I}i}$, we call it an i-concept of $\mathfrak{T}$. Note, that some particular concept C is possibly an i-concept and a j-concept for distinct indices $i, j \in I$, since we do not assume the two alphabets represented by $N_{\mathrm{C}i}$, $N_{\mathrm{R}i}$, $N_{\mathrm{I}i}$ and $N_{\mathrm{C}j}$, $N_{\mathrm{R}j}$, $N_{\mathrm{I}j}$ to be disjoint (see also the disjoint names normal form, Definition 64). Also, in order to state that certain axiom ϕ belongs to $\mathcal{T}_i$ we write $i : \phi$.

As outlined above, equivalence bridge rules are mere syntactic sugar: a combination of two bridge rules $i : C \xrightarrow{\sqsubseteq} j : G \in \mathfrak{B}$ and $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$ is denoted by $i : C \xrightarrow{\equiv} j : G \in \mathfrak{B}$. Note that since bridge rules are directed, the sets $\mathfrak{B}_{ij}$ and $\mathfrak{B}_{ji}$ are possibly and expectedly distinct, for a given pair of indices $i, j \in I$, $i \neq j$. Let us now have a look at the following example in order to illustrate the syntax.

Example 7. *Consider a distributed TBox with two local TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$. The local TBox $\mathcal{T}_1$ is concerned with geography and it contains the following GCI axiom:*

$$1 : \mathsf{DouroValley} \sqsubseteq \mathsf{Portugal} \ .$$

On the other hand, $\mathcal{T}_2$ is concerned with wine and it contains the following axiom:

$$2 : \mathsf{Port} \sqsubseteq \forall \mathsf{producedIn}.\mathsf{OPortoRegion} \ .$$

Now, we do not want to repeat all the geography information in $\mathcal{T}_2$, we would like to reuse the knowledge already recorded in $\mathcal{T}_1$. We do it by introducing bridge rules directed from $\mathcal{T}_1$ to $\mathcal{T}_2$, such as for instance:

$$1 : \mathsf{DouroValley} \xrightarrow{\sqsupseteq} 2 : \mathsf{OPortoRegion} \ ,$$

$$1 : \mathsf{Portugal} \xrightarrow{\sqsubseteq} 2 : \mathsf{PortugalRegion} \ .$$

Below we will see that in such a distributed TBox, the concept Port *is subsumed by the concept* $\forall$producedIn.PortugalRegion*. This subsumption relation is derived in $\mathcal{T}_2$ using the bridge rules and the knowledge in $\mathcal{T}_1$. It is not directly recorded in $\mathcal{T}_2$.*

The semantics of DDL builds on top of the semantics of $\mathcal{SHIQ}$, as introduced in previous chapter. Each local TBox $\mathcal{T}_i$ is assigned a local interpretation $\mathcal{I}_i = \left\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i} \right\rangle$, consisting of the local domain $\Delta^{\mathcal{I}_i}$ and the local interpretation function $\cdot^{\mathcal{I}_i}$.

The aim of DDL is to relate elements from different ontology units. In the semantics this is implemented using so called domain relations. Given $i, j \in I$, $i \neq j$, a domain relation r_{ij} is a relation between $\Delta^{\mathcal{I}_i}$ and $\Delta^{\mathcal{I}_j}$. The semantic mapping expressed using bridge rules constraints this relation and thus semantic relations are propagated from one ontology unit to another. In accordance with the intuition that things are possibly modeled differently in different local ontologies, any relation $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ is permitted. Thus, one element from the domain $\Delta^{\mathcal{I}_i}$ is possibly mapped to multiple elements of the domain $\Delta^{\mathcal{I}_j}$ and vice versa.

A structure that connects local interpretations for each ontology units and domain relations is called distributed interpretation. Distributed interpretations are formally introduced as follows. There are two flavours of distributed interpretation. First is the d-interpretation, which was used in first papers on DDL.

Definition 56 (d-interpretation)**.** *Given a distributed TBox $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ over an index set I, a d-interpretation of $\mathfrak{T}$ is a pair $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i,j \in I, i \neq j} \rangle$, consisting of a set of local interpretations $\{\mathcal{I}_i\}_{i \in I}$ and a set of domain relations $\{r_{ij}\}_{i,j \in I, i \neq j}$. Each $\mathcal{I}_i = \left\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i} \right\rangle$ is an interpretation in the respective sublanguage of $\mathcal{SHIQ}$, with nonempty domain $\Delta^{\mathcal{I}_i}$. In addition, the local domains are mutually disjoint: for each $i, j \in I$, $i \neq j$, we have $\Delta^{\mathcal{I}_i} \cap \Delta^{\mathcal{I}_j} = \emptyset$. Each domain relation r_{ij} is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$.*

Later on the notion of a distributed interpretation has been altered in order to prevent the whole DDL knowledge base to become necessarily inconsistent if one of its local ontology units is inconsistent. For this reason a special kind of local interpretations called *holes* has been introduced. First introduced by Serafini & Tamilin (2004), however the notion has been consecutively developed until finally stabilized by Serafini et al. (2005).

Definition 57 (Hole)**.** *A hole is a novel kind of local interpretation $\mathcal{I}^\epsilon = \langle \emptyset, \cdot^\epsilon \rangle$ such that its domain is the empty set $\emptyset$ and its interpretation function assigns $X^\epsilon = \emptyset$ to every concept, role or individual X.*

Distributed interpretations that allow also holes in place of local interpretations are called ϵ-interpretations. It is to be stressed, that ϵ-interpretations is the

main notion of interpretation used in DDL, hence when we just say "distributed interpretation" we always mean an ϵ-interpretation.

Definition 58 (ϵ-interpretation)**.** *Given a distributed TBox $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$ over an index set I, an ϵ-interpretation of $\mathfrak{T}$ is a pair $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i\in I}, \{r_{ij}\}_{i,j\in I, i\neq j}\rangle$, consisting of a set of local interpretations $\{\mathcal{I}_i\}_{i\in I}$ and a set of domain relations $\{r_{ij}\}_{i,j\in I, i\neq j}$. Each $\mathcal{I}_i = \langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}\rangle$ is an interpretation in the respective sublanguage of $\mathcal{SHIQ}$, or it is a hole (i.e., in the second case $\mathcal{I}_i = \mathcal{I}^\epsilon$). The local domains are mutually disjoint except for the holes: for each $i, j \in I$, $i \neq j$, either $\Delta^{\mathcal{I}_i} \cap \Delta^{\mathcal{I}_j} = \emptyset$ or $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_j} = \emptyset$. Each domain relation r_{ij} is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$.*

The following auxiliary notation is frequently used when dealing with domain relations: $r_{ij}(d)$ denotes the set $\{d' \mid \langle d, d'\rangle \in r_{ij}\}$; $r_{ij}(D)$ denotes the set $\bigcup_{d\in D} r_{ij}(d)$; r denotes the union $\bigcup_{i,j\in I} r_{ij}$.

A distributed interpretation is a model of a DDL knowledge base, if each of its local interpretation is a model of the respective local ontology unit and in addition the domain relations satisfy the constraints generated by bridge rules. Formal definition follows.

Definition 59 (Distributed model)**.** *For every i and j, a distributed interpretation $\mathfrak{I}$ satisfies the elements of a distributed TBox $\mathfrak{T}$ (denoted by $\mathfrak{I} \models_\epsilon \cdot$) according to the following clauses:*

1. $\mathfrak{I} \models_\epsilon i : C \sqsubseteq D$ *if* $\mathcal{I}_i \models C \sqsubseteq D$;
2. $\mathfrak{I} \models_\epsilon \mathcal{T}_i$ *if* $\mathcal{I}_i$ *is a model of* $\mathcal{T}_i$;
3. $\mathfrak{I} \models_\epsilon i : C \xrightarrow{\sqsubseteq} j : G$ *if* $r_{ij}\big(C^{\mathcal{I}_i}\big) \subseteq G^{\mathcal{I}_j}$;
4. $\mathfrak{I} \models_\epsilon i : C \xrightarrow{\sqsupseteq} j : G$ *if* $r_{ij}\big(C^{\mathcal{I}_i}\big) \supseteq G^{\mathcal{I}_j}$;
5. $\mathfrak{I} \models_\epsilon \mathfrak{B}$ *if* $\mathfrak{I}$ *satisfies all bridge rules in* $\mathfrak{B}$;
6. *finally,* $\mathfrak{I}$ *is said to be a distributed model of* $\mathfrak{T}$ *(denoted by* $\mathfrak{I} \models_\epsilon \mathfrak{T}$*) if* $\mathfrak{I} \models_\epsilon \mathfrak{B}$ *and* $\mathfrak{I} \models_\epsilon \mathcal{T}_i$ *for each* i.

If $\mathfrak{I} \models_\epsilon \mathfrak{T}$, we also sometimes call $\mathfrak{I}$ an ϵ-model or just a model of $\mathfrak{T}$. If $\mathfrak{I}$ is a d-interpretation and $\mathfrak{I} \models_\epsilon \mathfrak{T}$, we sometimes say that it is a d-model of $\mathfrak{T}$ and we denote it by $\mathfrak{I} \models_{\mathrm{d}} \mathfrak{T}$.

The two main decision problems remain to be satisfiability of concepts and entailment of subsumption. Also in DDL, as usual in DL, these two problems are interreducible (see Sect. 3.3). Please note that both these decision problems are always defined with respect to some particular local knowledge base, identified by an index $i \in I$.

Definition 60 (Concept satisfiability)**.** *Given a DDL knowledge base $\mathfrak{T}$ over an index set I, some index $i \in I$, we say that an i-concept C is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ if there exists a distributed model $\mathfrak{I}$ of $\mathfrak{T}$ such that $C^{\mathcal{I}_i} \neq \emptyset$.*

If we do not say otherwise, we also admit ϵ-models when judging satisfiability. Please note that a hole is not allowed in place of $\mathcal{I}_i$ in the definition above, since there exists some $x \in C^{\mathcal{I}_i} \subseteq \Delta^{\mathcal{I}_i}$. If we want to stress that d-model is required, we use the term d-satisfiability (i.e., an i-concept C is d-satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ if there exists a d-model $\mathfrak{I}$ of $\mathfrak{T}$ such that $C^{\mathcal{I}_i} \neq \emptyset$). Let us now look at the problem of subsumption entailment.

Definition 61 (Subsumption entailment)**.** *Given a DDL knowledge base $\mathfrak{T}$ over some index set I, an index $i \in I$, and two i-concepts C and D, we say that C is subsumed by D with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ (denoted by $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$) whenever in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ we have $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.*

As much as with satisfiability, we also assume that ϵ-models in the definition above, if we do not say otherwise. We will use also the notion of d-entailment, in which a formula $i : C \sqsubseteq D$ is d-entailed if it is satisfied by each d-model of $\mathfrak{T}$. Again, we will clearly refer to d-entailment with any such usage, and denote it with the symbol $\models_{\mathrm{d}}$.

In order to proceed with a more formal example of entailment, we need the following lemma which we will frequently use when working with distributed interpretation and models.

Lemma 44. *Assume a DDL knowledge base $\mathfrak{T}$ over an index set I, a distributed interpretation $\mathfrak{I}$ and any two distinct indices $i, j \in I$. Given any two subsets $X, Y \subseteq \Delta^{\mathcal{I}_i}$, the following always holds:*

$$X \subseteq Y \implies r_{ij}(X) \subseteq r_{ij}(Y) \ .$$

Proof. Assume $x \in r_{ij}(X)$. This implies that there is $y \in X$, such that $x \in r_{ij}(y)$. Since $X \subseteq Y$ then also $y \in Y$ and hence $r_{ij}(y) \subseteq r_{ij}(Y)$. This finally implies that $x \in r_{ij}(Y)$. □

We will use this simple observation multiple times when arguing about images of the domain relation. In fact, similar property also holds for subsets of Cartesian product $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ of any local domain (i.e., the interpretation of roles), on which the mapping $r_{ij}(\cdot)$ can also be applied. Finally, let us now reinspect Example 7 in a more formal fashion.

Example 8. *Consider the distributed TBox $\mathfrak{T}$ introduced in Example 7. We shortly recapitulate the axioms for convenience:*

$$\begin{aligned} 1 : \mathsf{DouroValley} &\sqsubseteq \mathsf{Portugal} \ , \\ 2 : \mathsf{Port} &\sqsubseteq \forall \mathsf{producedIn}.\mathsf{OPortoRegion} \ , \\ 1 : \mathsf{DouroValley} &\xrightarrow{\sqsupseteq} 2 : \mathsf{OPortoRegion} \ . \\ 1 : \mathsf{Portugal} &\xrightarrow{\sqsubseteq} 2 : \mathsf{PortugalRegion} \ . \end{aligned}$$

We will show that $\mathfrak{T} \models_\epsilon 2 : \mathsf{Port} \sqsubseteq \forall \mathsf{producedIn}.\mathsf{PortugalRegion}$. Let $\mathfrak{I}$ be a distributed model of $\mathfrak{T}$. We have $\mathsf{DouroValley}^{\mathcal{I}_1} \subseteq \mathsf{Portugal}^{\mathcal{I}_1}$ from the first axiom. From Lemma 44 we get $r_{12}\Big(\mathsf{DouroValley}^{\mathcal{I}_1}\Big) \subseteq r_{12}\Big(\mathsf{Portugal}^{\mathcal{I}_1}\Big)$. Using last two axioms we get:

$$\mathsf{OPortoRegion}^{\mathcal{I}_2} \subseteq r_{12}\Big(\mathsf{DouroValley}^{\mathcal{I}_1}\Big) \subseteq r_{12}\Big(\mathsf{Portugal}^{\mathcal{I}_1}\Big) \subseteq \mathsf{PortugalRegion}^{\mathcal{I}_2} \ ,$$

and hence $\mathfrak{T} \models_\epsilon 2 : \mathsf{OPortoRegion} \sqsubseteq \mathsf{PortugalRegion}$.

However, we ought to prove that $\mathfrak{T} \models_\epsilon 2 : \mathsf{Port} \sqsubseteq \forall \mathsf{producedIn}.\mathsf{PortugalRegion}$, in other words, every producedIn*-successor y of any element x of $\mathsf{Port}^{\mathcal{I}_2}$ belongs to $\mathsf{PortugalRegion}^{\mathcal{I}_2}$. From the second axiom above we get that any such $y \in \mathsf{OPortoRegion}^{\mathcal{I}_2}$. We have shown above that $\mathsf{OPortoRegion}^{\mathcal{I}_2} \subseteq \mathsf{PortugalRegion}^{\mathcal{I}_2}$ and hence $y \in \mathsf{PortugalRegion}^{\mathcal{I}_2}$.*

Please notice, that in the last paragraph of the example above, we have effectively proven that $\mathfrak{T} \models_\epsilon i : C \sqsubseteq \forall R.D$ and $\mathfrak{T} \models_\epsilon i : D \sqsubseteq E$ implies $\mathfrak{T} \models_\epsilon i : C \sqsubseteq \forall R.E$ – a proposition which also holds in any $\mathcal{SHOIQ}$. We will see in the following section that there is no need to repeat the proof of this or any other proposition that holds in the local knowledge base (see the Monotonicity property below and Theorem 47).

There are two options when talking about the consistence of knowledge bases in DDL. We either demand for global consistence, a state when every single local ontology is consistent, or we might be interested in consistence within a particular local ontology unit $\mathcal{T}_i$, not deeming the rest of the system. The first, stronger notion is called global consistence and is defined as follows.

Definition 62 (Global consistence)**.** *A DDL knowledge base $\mathfrak{T}$ over an index set I is said to be globally consistent, if there exists a d-model $\mathfrak{I}$ of $\mathfrak{T}$.*

The second, weaker notion of consistence is called i-consistence, reflecting the fact that it is the local knowledge $\mathcal{K}_i$ which is required to be consistent. Formal definition follows.

Definition 63 (i-consistence)**.** *A DDL knowledge base $\mathfrak{T}$ over an index set I is said to be i-consistent for some $i \in I$, if there exists an ϵ-model $\mathfrak{I}$ of $\mathfrak{T}$ such that $\Delta^{\mathcal{I}_i} \neq \emptyset$.*

As we observe from the definitions, there is a notable correspondence between the global consistence and d-models and also between the i-consistence and ϵ-models.

We will conclude this section with some auxiliary definitions and two normal forms for DDL knowledge bases that will simplify the notation later on. Let us start with the notion of bridge graph. Given a DDL knowledge base $\mathfrak{T}$ over an index set I, the bridge graph of $\mathfrak{T}$ (denoted by $\mathrm{G}_{\mathfrak{T}}$) is a graph $\mathrm{G}_{\mathfrak{T}} = \langle V, E\rangle$ such that $V = I$ and $E = \{\langle i, j\rangle \mid i, j \in I \wedge \mathfrak{B}_{ij} \neq \emptyset\}$.

Another notion we will need is the one of a reduced DDL knowledge base. Given a DDL knowledge base $\mathfrak{T}$ defined over some index set I and given $J \subseteq I$, we denote by $\mathfrak{T} \setminus\setminus J$ the DDL knowledge base obtained from $\mathfrak{T}$ by removing all local knowledge bases $\mathcal{T}_j$ such that $j \in J$ including all directly adjacent bridge rules. More formally, $\mathfrak{T} \setminus\setminus J$ is a knowledge base indexed by $I \setminus J$ such that $\mathfrak{T} \setminus\setminus J = \left\langle \{\mathcal{T}_i\}_{i \in I\setminus J}, \bigcup_{i,j \in I\setminus J} \mathfrak{B}_{ij} \right\rangle$. We say that $\mathfrak{T} \setminus\setminus J$ is $\mathfrak{T}$ reduced by J.

While, in the DDL setting, it is considered very natural that two distinct local ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$ of some distributed ontology $\mathfrak{T}$ possibly share some names of concepts, roles and individuals, there is in fact no difference in the semantics if we take away this liberty. This is formally established as the disjoint names normal form.

Definition 64 (Disjoint names normal form)**.** *Assume a DDL knowledge base $\mathfrak{T}$ over an index set I, that is build using the atomic concept names $\{N_{\mathrm{C}i}\}_{i\in I}$, atomic role names $\{N_{\mathrm{R}i}\}_{i\in I}$ and individual names $\{N_{\mathrm{I}i}\}_{i\in I}$. We say that $\mathfrak{T}$ is in the disjoint names normal form whenever for any $i, j \in I$, such that $i \neq j$, we have $N_{\mathrm{C}i} \cap N_{\mathrm{C}j} = \emptyset$, $N_{\mathrm{R}i} \cap N_{\mathrm{R}j} = \emptyset$ and $N_{\mathrm{I}i} \cap N_{\mathrm{I}j} = \emptyset$.*

Theorem 45. *For any DDL knowledge base $\mathfrak{T}$ there exists a DDL knowledge base $\mathfrak{T}'$ in the disjoint names normal form that is equivalent to $\mathfrak{T}$ disregarding the renaming.*

Proof. Let $\mathfrak{T}$ be an arbitrary DDL knowledge base over some index set I. And let the vocabulary of $\mathfrak{T}$ be denoted $\{N_{\mathrm{C}i}\}_{i\in I}$, $\{N_{\mathrm{R}i}\}_{i\in I}$ and $\{N_{\mathrm{I}i}\}_{i\in I}$, as usual. Let for each $i \in I$, $N'_{\mathrm{C}i} = \{A_i \mid A \in N_{\mathrm{C}i}\}$, $N'_{\mathrm{R}i} = \{R_i \mid R \in N_{\mathrm{C}i}\}$ and $N'_{\mathrm{I}i} = \{a_i \mid a \in N_{\mathrm{I}i}\}$. Clearly, $\mathfrak{T}'$, obtained by replacing in $\mathcal{T}_i$ every occurrence of X by X_i for every $X \in N_{\mathrm{C}i} \cup N_{\mathrm{R}i} \cup N_{\mathrm{I}i}$ and for every $i \in I$, is a DDL knowledge base in the disjoint names normal form.

It remains to show that $\mathfrak{T}'$ is equivalent with $\mathfrak{T}$ disregarding the renaming. This is shown by proving, for every $i \in I$, that for any i-concept C it holds that C is satisfiable with respect to $\mathfrak{T}$ if and only if C' is satisfiable with respect to $\mathfrak{T}'$, where C' is obtained by replacing each $X \in N_{\mathrm{C}i} \cup N_{\mathrm{R}i} \cup N_{\mathrm{I}i}$ that appears in C by X_i. It is straightforward to prove (by structural induction on C) that for every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that $C^{\mathcal{I}_i} = C'^{\mathcal{I}'_i}$, where $\mathfrak{I}'$ is obtained from $\mathfrak{I}$ by the very same renaming as $\mathfrak{T}'$ from $\mathfrak{T}$. □

And so indeed we see that the coincidence of local names has nothing to do with semantics of DDL. Instead local concepts are exclusively related in DDL by means of bridge rules.

In many works on DDL we find the assumption that only atomic concepts are allowed in bridge rules. We did not introduce this limitation in this thesis. Neither our approach yields a different formalism, neither this assumption does limit the expressive power of DDL. Both approaches are equivalent, as noted also in the previous work.

Definition 65 (Atomic bridge rules normal form)**.** *A DDL knowledge base $\mathfrak{T} = \langle\{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$ over an index set I is in the atomic bridge rules normal form if all bridge rules of $\mathfrak{B}$ are of one of the following two forms:*

$$i : A \xrightarrow{\sqsubseteq} j : B \ , \qquad\qquad i : A \xrightarrow{\sqsupseteq} j : B \ ,$$

where i, j are any two distinct indices of I and $A, B \in N_{\mathrm{C}i}$ are atomic concepts.

Theorem 46. *For every DDL knowledge base $\mathfrak{T}$ there exists a DDL knowledge base $\mathfrak{T}'$ in the atomic bridge rules normal form that is equivalent to $\mathfrak{T}$.*

Proof. Let $\mathfrak{T} = \langle\{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$ by an arbitrary DDL knowledge base over an index set I. Assume, without loss of generality, that the number of bridge rules in $\mathfrak{B}$ is n. Let us denote these bridge rules by $\phi_1, \ldots, \phi_n$, and for each ϕ_k, $1 \leq k \leq n$ let us denote the concept on the left hand side of ϕ_k by C_k and the concept on the right hand side of ϕ_k by D_k. Let $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$ be all new atomic concept names that do not appear in $N_{\mathrm{C}i}$ for any $i \in I$. Let $\mathfrak{T}' = \langle\{\mathcal{T}'_i\}_{i\in I}, \mathfrak{B}'\rangle$ be a new knowledge base such that:

1. $\mathcal{T}'_i = \mathcal{T}_i \cup \{A_k \equiv C_k \mid i : C_k \xrightarrow{\sqsubseteq} j : D_k \in \mathfrak{B} \vee i : C_k \xrightarrow{\sqsupseteq} j : D_k \in \mathfrak{B}\} \cup$
   $\{B_k \equiv D_k \mid j : C_k \xrightarrow{\sqsubseteq} i : D_k \in \mathfrak{B} \vee j : C_k \xrightarrow{\sqsupseteq} i : D_k \in \mathfrak{B}\}$;
2. $\mathfrak{B}' = \{i : A_k \xrightarrow{\sqsubseteq} j : B_k \mid \phi_k = i : C_k \xrightarrow{\sqsubseteq} j : D_k \in \mathfrak{B}\} \cup$
   $\{i : A_k \xrightarrow{\sqsupseteq} j : B_k \mid \phi_k = i : C_k \xrightarrow{\sqsupseteq} j : D_k \in \mathfrak{B}\}$.

We shall prove that any i-concept C, that appears in $\mathfrak{T}$, is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ if and only if C is satisfiable with respect to $\mathcal{T}'_i$ of $\mathfrak{T}'$. Let us start with the *only-if part*. Assume that C is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ and therefore there is a model of $\mathfrak{I}$ of $\mathfrak{T}$ such that $C^{\mathcal{I}_i} \neq \emptyset$. Let us construct a

model $\mathfrak{I}'$ of $\mathfrak{T}'$ by extending $\mathfrak{I}$: for each bridge rule ϕ_k of $\mathfrak{B}$ that is either of the form $\phi_k = i : C_k \xrightarrow{\sqsubseteq} j : D_k$ or of the form $i : C_k \xrightarrow{\sqsupseteq} j : D_k$, for some $i, j \in I$ we set ${A_k}^{\mathcal{I}'_i} = {C_k}^{\mathcal{I}_i}$ and ${B_k}^{\mathcal{I}'_j} = {D_k}^{\mathcal{I}_j}$. Everything else is copied into $\mathfrak{I}'$ from $\mathfrak{I}$. It is now straightforward to verify that $\mathfrak{I}'$ is a distributed model of $\mathfrak{T}'$. Since $C^{\mathcal{I}'_i} = C^{\mathcal{I}_i} \neq \emptyset$ it follows that C is satisfiable with respect to $\mathcal{T}'_i$ of $\mathfrak{T}'$.

The *if part* is trivial. Let us assume that C is satisfiable with respect to $\mathcal{T}'_i$ of $\mathfrak{T}'$ and therefore there exists as model $\mathfrak{I}$ of $\mathfrak{T}'$ such that $C^{\mathcal{I}_i} \neq \emptyset$. However, $\mathfrak{I}$ is also a model of $\mathfrak{T}$, hence C is also satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$. □

3.2 Properties of DDL

In the previous section we have formalized the syntax and the semantics of DDL, one of the formal frameworks for representing distributed and modular ontologies. Once we have the semantics, the important research question is how this semantics behaves and how it should behave. This question has been studied in literature and various desired properties of DDL have been suggested and the behaviour of DDL with respect to these properties has been investigated by Borgida & Serafini (2003), Serafini & Tamilin (2004), Serafini et al. (2005). In this section, we give an overview of these properties, characterizing the behaviour of DDL. Mind also an interesting paper by Loebe (2006) which brings this investigation on a more abstract level, concentrating on the desired behaviour of a distributed ontology in general, and not only within one particular framework.

Our very first concern is about the monotonicity of the framework. DDL aim at reusing knowledge: by combining multiple local ontologies we will be able to derive more entailed conclusions. We do not want to lose information in this process. Hence the monotonicity property requires that bridge rules never delete local subsumptions that are implied by the local semantics (Borgida & Serafini 2003, Serafini & Tamilin 2004).

Property 1 (Monotonicity)**.** *The monotonicity property is satisfied whenever in every distributed TBox $\mathfrak{T}$ it holds that*

$$\mathcal{T}_i \models C \sqsubseteq D \implies \mathfrak{T} \models_\epsilon i : C \sqsubseteq D \ .$$

Bridge rules are directed and hence it is desired that information is reused only according to their direction. If the mapping is expressed only in the direction from $\mathcal{T}_i$ to $\mathcal{T}_j$, the first local ontology unit should not be affected by the fact that its knowledge is being reused somewhere elsewhere. It is also sometimes said that there should be no *backflow* in DDL knowledge bases. The formalization of this desideratum has undergone some development; here we present the most recent version introduced by Serafini et al. (2005).

Property 2 (Directionality)**.** *The directionality property is satisfied whenever in every distributed TBox $\mathfrak{T}$ with an index set I and bridge graph $\mathrm{G}_\mathfrak{T}$, for every two distinct indices $i, j \in I$, it holds that if there is no directed path between i to j in $\mathrm{G}_\mathfrak{T}$, then*

$$\mathfrak{T} \models_\epsilon j : C \sqsubseteq D \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D \ .$$

The exact mechanism of knowledge reuse has been an important research question ever since the first introduction of DDL. For instance, Serafini & Tamilin (2004) propose a hypothesis, that in order for knowledge to be transfered, a combination of into- and onto- bridge rules is required. This hypothesis has been formalized into the strong directionality property.

Property 3 (Strong directionality)**.** *The strong directionality property is satisfied whenever for each distributed TBox $\mathfrak{T}$ with an index set I the following condition holds: given any $i \in I$, if either for all $k \in I$, $k \neq i$, $\mathfrak{B}_{ki}$ contains no into-bridge rules, or for all $k \in I$, $k \neq i$, $\mathfrak{B}_{ki}$ contains no onto-bridge rules, then*

$$\mathfrak{T} \models_\epsilon i : A \sqsubseteq B \implies \mathcal{T}_i \models A \sqsubseteq B \ .$$

This property, however, does not verify in DDL. Consider a very simple distributed TBox $\mathfrak{T}$ over $I = \{1, 2\}$: $\mathfrak{T} = \langle \{\emptyset, \emptyset\}, \{1 : \bot \xrightarrow{\sqsupseteq} 2 : E\} \rangle$. It is easy to verify, that $\mathfrak{T} \models_\epsilon 2 : E \sqsubseteq \bot$. This should not be true according to the strong directionality, as $T_2 \not\models E \sqsubseteq \bot$.

As we have learned, the DDL semantics allows holes (i.e., local interpretation with empty domain) in order to restrict propagation of accidental inconsistency that possibly appears in one or more of the ontology units. If this happens, we do not want the whole distributed ontology to become inconsistent, as it would be the case without holes or some other mechanism. However, if local inconsistency appears, the distributed ontology is affected. The exact amount of acceptable inconsistency propagation is characterized by the local inconsistency property. Notice that the characterization is based on d-entailment.

Property 4 (Local inconsistency)**.** *The local inconsistency property is satisfied whenever in each distributed TBox $\mathfrak{T}$ with an index set I the following holds:*

$$\mathfrak{T} \models_\epsilon i : C \sqsubseteq D \iff ((\forall J)\ J \subseteq I \wedge i \notin J \implies \mathfrak{T}(\epsilon_J) \models_{\mathrm{d}} i : C \sqsubseteq D) \ ,$$

where $\mathfrak{T}(\epsilon_J) = \mathfrak{T} \setminus\setminus J \cup \{D \sqsubseteq \bot \mid j : C \xrightarrow{\sqsupseteq} i : D \in \mathfrak{B} \wedge j \in J\}$.

With last two properties we shift our attention towards propagation of concept subsumption along bridge rules. Subsumption propagation is the basic mechanism of knowledge reuse in DDL, and it is given particular attention by Borgida & Serafini (2003), Serafini & Tamilin (2004), Serafini et al. (2005). In the most basic setting, subsumption propagation is achieved using the combination of a single into- and a single onto-bridge rule.

Property 5 (Simple subsumption propagation)**.** *The simple subsumption propagation property is satisfied whenever for each distributed TBox $\mathfrak{T}$ with an index set I the following holds: if $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$ and $i : D \xrightarrow{\sqsubseteq} j : H \in \mathfrak{B}$ then*

$$\mathfrak{T} \models_\epsilon i : C \sqsubseteq D \implies \mathfrak{T} \models_\epsilon j : G \sqsubseteq H \ .$$

In fact, subsumption does propagate in DDL even in more general setting, as established by the following property. This property is of particular importance, later it serves as a base to establish a distributed reasoning algorithm.

Property 6 (Generalized subsumption propagation)**.** *The generalized subsumption propagation property is satisfied whenever, for each distributed TBox $\mathfrak{T}$ with*

an index set I, the following holds: if $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$ and $i : D_k \xrightarrow{\sqsubseteq} j : H_k \in \mathfrak{B}$, for $1 \leq k \leq n$ then

$$\mathfrak{T} \models_\epsilon i : C \sqsubseteq \bigsqcup_{k=1}^{n} D_k \implies \mathfrak{T} \models_\epsilon j : G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \ .$$

All of the properties presented above except for the strong directionality property are actually satisfied by DDL.

Theorem 47 (Borgida & Serafini (2003), Serafini & Tamilin (2004), Serafini et al. (2005))**.** *The monotonicity, directionality, local inconsistency, simple and generalized subsumption propagation properties are satisfied by DDL.*

The properties discussed in this section provide an important characterization of the semantics of DDL. They show that DDL stand out as a distributed ontology representation formalism that enables for knowledge reuse, it is a very reasonable extension of DL, the mechanism of knowledge reuse is practical and well understood and in addition DDL are robust enough to deal with distributed environments where occasional local inconsistency may be present.

3.3 Reasoning with DDL

As in many other similar frameworks, the two main decision problems are concept satisfiability and entailment of subsumption, which we have formally introduced in Sect. 3.1. Typically, entailment and (un)satisfiability are interreducible, and DDL are no exception to this.

Theorem 48. *Given a DDL knowledge base $\mathfrak{T}$ with an index set I, and given some $i \in I$ and any i-concept C, C is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ if and only if C is not subsumed by $\bot$ with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ (i.e., if and only if $\mathfrak{T} \models_\epsilon i : C \sqsubseteq \bot$ does not hold).*

Proof. The theorem is in fact a trivial observation: C is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ $\iff$ there exists a model $\mathfrak{I}$ of $\mathfrak{T}$ that contains $x \in C^{\mathcal{I}_i}$ $\iff$ $\mathfrak{T} \models_\epsilon i : C \sqsubseteq \bot$ does not hold. □

Theorem 49. *Given a DDL knowledge base $\mathfrak{T}$ with an index set I, and given some $i \in I$ and any two i-concepts C and D, C is subsumed by D with respect to $\mathcal{T}_i$ of $\mathfrak{T}$ (i.e., $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$) if and only if $C \sqcap \neg D$ is unsatisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$.*

Proof. Also this theorem is rather straightforward: $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ holds $\iff$ in every model $\mathfrak{I}$ of $\mathfrak{T}$, each $x \in C^{\mathcal{I}_i}$ also belongs to $D^{\mathcal{I}_i}$ $\iff$ there is no model $\mathfrak{I}'$ of $\mathfrak{T}$ with $x \in C^{\mathcal{I}'_i}$ such that $x \in \neg D^{\mathcal{I}'_i}$ $\iff$ $C \sqcap \neg D$ is unsatisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$. □

A distributed tableaux reasoning algorithm that decides satisfiability of concepts with respect to a DDL knowledge base, and hence also subsumption entailment, has been developed by Serafini & Tamilin (2004), Serafini et al. (2005), Tamilin (2007).

The algorithm is based on a fix-point characterization of the semantics. The basic idea is to apply the generalized subsumption propagation property (Property 6) in an exhaustive way in order to reduce bridge rules into local GCI axioms. An operator $\mathfrak{B}_{ij}(\cdot)$ is introduced, that propagates as much knowledge from $\mathcal{T}_i$ to $\mathcal{T}_j$ applying rules from $\mathfrak{B}_{ij}$ as can be computed in one step.

Definition 66 ($\mathfrak{B}_{ij}(\cdot)$ operator)**.** *Given a distributed TBox $\mathfrak{T}$ with two local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ each in respective language $\mathcal{L}_i$ and $\mathcal{L}_j$, $\mathfrak{B}_{ij}(\cdot)$ takes a TBox in $\mathcal{L}_i$ and produces a TBox in $\mathcal{L}_j$ such that*

$$\mathfrak{B}_{ij}(\mathcal{T}_i) = \left\{ G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \;\middle|\; \begin{array}{l} \mathcal{T}_i \models A \sqsubseteq \sqcup_{k=1}^{n} B_k, \\ i : A \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}_{ij}, \\ i : B_k \xrightarrow{\sqsubseteq} j : H_k \in \mathfrak{B}_{ij}, \\ 1 \leq k \leq n \end{array} \right\} .$$

Consequently, the operator $\mathfrak{B}(\cdot)$ is introduced, that takes a whole DDL knowledge base and caries out as much computation as can be done in one step on all local TBoxes contained therein.

Definition 67 ($\mathfrak{B}(\cdot)$ operator)**.** *Given a distributed TBox $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ indexed with I, the operator $\mathfrak{B}(\cdot)$ is defined as follows:*

$$\mathfrak{B}(\mathfrak{T}) = \left\langle \left\{ \mathcal{T}_i \cup \bigcup_{j \neq i} \mathfrak{B}_{ij}(\mathcal{T}_j) \right\}_{i \in I}, \mathfrak{B} \right\rangle .$$

As showed by Serafini et al. (2005) the $\mathfrak{B}(\cdot)$ operator has a fix-point, if repeatedly applied to some distributed TBox.

Theorem 50 (Serafini et al. (2005))**.** *Given any DDL knowledge base $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$, there always exists an integer b such that $\mathfrak{B}^b(\mathfrak{T}) = \mathfrak{B}^{b+1}(\mathfrak{T})$.*

For any DDL knowledge base $\mathfrak{T}$, we denote this fix-point, that is, $\mathfrak{B}^b(\mathfrak{T})$ in the previous theorem, by $\mathfrak{B}^*(\mathfrak{T})$. Finally, the following theorem shows that, by computing $\mathfrak{B}^*(\mathfrak{T})$, deciding subsumption entailment with respect to a distributed TBox is reduced to deciding subsumption in a local TBox.

Theorem 51 (Serafini et al. (2005))**.** *Let $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$, be a DDL knowledge base with some index set I. Let $\mathfrak{B}^*(\mathfrak{T}) = \langle \{\mathcal{T}_i^*\}_{i \in I}, \mathfrak{B}^* \rangle$ be the fix-point computed according to Theorem 50. Then, $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ if and only if $\mathcal{T}_i^* \models C \sqsubseteq D$.*

From the theorem we see, that after a preprocessing phase in which $\mathfrak{B}^*(\mathfrak{T})$ is computed, only local reasoning is needed. Of course it would be inefficient in general to always apply all bridge rules exhaustingly like this. Therefore the algorithm takes a different approach.

The algorithm is based on the tableaux algorithm for $\mathcal{SHIQ}$, established by (Horrocks et al. 1999). Formally, the algorithm is implemented as a function DTab_j which takes a j-concept as an input and launches a satisfiability check of C with respect to $\mathcal{T}_j$ of $\mathfrak{T}$. The algorithm uses $\mathcal{SHIQ}$ tableaux rules (see Fig. 2.4) and in addition one new rule, which is called $\mathfrak{B}_{ij}$-rule (listed in Fig. 3.1).

$\mathfrak{B}_{ij}$-**rule:**
If $G \in \mathcal{L}_j(x)$,
$H \notin \mathcal{L}_j(x)$,
$i : A \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}_{ij}$,
$\mathrm{DTab}_i(A \sqcap \neg \bigsqcup B) = \mathrm{false}$,
where $BH \subseteq \{\langle B_k, H_k\rangle \mid i : B_k \xrightarrow{\sqsubseteq} j : H_k \in \mathfrak{B}_{ij}\}$,
$B = \{B \mid \langle B, X\rangle \in BH\}$,
$H = \{H \mid \langle Y, H\rangle \in BH\}$,
then set $\mathcal{L}_j(x) = \mathcal{L}_j(x) \cup \{\bigsqcup H\}$.

Figure 3.1: The $\mathfrak{B}_{ij}$ tableaux rule used by original DDL

Upon calling $\mathrm{DTab}_j(C)$, the algorithm tries to compute as much of the tableaux as possible locally within $\mathcal{T}_j$, using the $\mathcal{SHIQ}$ tableaux rules. Whenever a node is found within the local tableaux that triggers a particular combination of bridge rules, the $\mathfrak{B}_{ij}$-rule is fired. From this rule, a recursive (un)satisfiability check is launched by calling DTab_i, for some $i \neq j$. Let us put this more formally.

Algorithm 7 (Satisfiability in DDL (DTab_j)**.** *Given a j-concept C in NNF and a distributed TBox $\mathfrak{T}$ on the input, the algorithm executes three steps:*

1. initialization: *create a new completion tree $T_j = (\{s_0\}, \emptyset, \mathcal{L}_j, \emptyset)$ and set $\mathcal{L}(s_0) = C$;*
2. tableau expansion: *exhaustingly apply the tableaux expansion rules listed in Fig. 2.4 together the $\mathfrak{B}_{ij}$-rule listed in Fig. 3.1. This step is over when none of the rules is applicable, and all recursive calls of* DTab_i *for any $i \in I$ terminated.*
3. answer: *if the completion tree is clash-free and all recursive calls of* DTab_i *for any $i \in I$ answered "unsatisfiable", then answer "satisfiable". Otherwise, answer "unsatisfiable".*

The algorithm represented by DTab_i is correct: it always terminates and it is sound and complete.

Theorem 52 (Serafini et al. (2005))**.** *Given some DDL knowledge base $\mathfrak{T} = \langle\{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$ with an index set I and a i-concept C, the algorithm* DTab_i *always terminates and the answer* $\mathrm{DTab}_i(C) = \mathrm{true}$ *if and only if C is satisfiable with respect to $\mathcal{T}_i$ of $\mathfrak{T}$.*

According to (Ghidini et al. 2008), the computational complexity of this algorithm is in ExpTime. A further optimized version of DTab_i has been also introduced that employs caching of the values of DTab_i, so that they are not recomputed several times during the run of the algorithm (Serafini et al. 2005). In order to deal with cyclic dependencies in within a DDL knowledge base (whenever the bridge graph $\mathrm{G}_{\mathfrak{T}}$ is cyclic) a blocking strategy is employed, as described by Tamilin (2007, Sect. 5.3.4, p. 91).

The algorithm has been implemented into a distributed reasoner DRAGO[1] (Serafini & Tamilin 2005). DRAGO uses the reasoner Pellet for local tableaux.

[1] The DRAGO reasoner is available through its home page: `http://drago.itc.it/`.

While it can be run also on a single machine, it permits a fully distributed reasoning strategy where the reasoning task is divided between multiple autonomous local reasoning agents, one for each local ontology unit. For further implementation and optimization details we kindly refer the reader to the work of Serafini et al. (2005), Serafini & Tamilin (2005), Tamilin (2007).

Finally, it is worth noting, that this is not the only approach to reasoning in DDL that is known from the literature. A completely different approach to this problem is undertaken by Schlicht & Stuckenschmidt (2008), where distributed reasoning algorithm based on resolution techniques has been introduced.

3.4 Extensions of DDL

We have deliberately chosen to present a rather basic version of DDL in this chapter, the one that only permits bridge rules between concepts, as we built our research in the following two chapters on this particular version. However, DDL have been further developed, in order to achieve a formalism that is capable to deal also with mapping between other types of ontology entities. In these richer versions of DDL it is possible to express, and reason with, mapping between ABox individuals, mapping between roles, and also so called heterogeneous mapping between concepts on one side and roles on the other one. In this section we briefly discuss these extensions of DDL for sake of completeness. Later on, in Chap. 6, where we compare DDL with other distributed and modular ontology frameworks, we will revisit some of them formally.

Additional axioms aimed to express correspondence between individuals have been suggested already in the original paper by Borgida & Serafini (2002). Later on, individual correspondences have been studied by Serafini & Tamilin (2006, 2007), Tamilin (2007). The form on the left is called partial individual correspondence, and the form on the right is called total individual correspondence:

$$i : a \longmapsto j : b \ , \qquad\qquad i : a \overset{=}{\longmapsto} j : \{b_1, \ldots, b_n\} \ ,$$

where $a \in N_{\mathrm{I}i}$ and $b, b_1, \ldots, b_n \in N_{\mathrm{I}j}$ are individual names.

In the intuitive reading, by partial individual correspondence one associates the individual on the left, with the other one on the right. This is reflected in the semantics by requiring $b^{\mathcal{I}_j} \in r_{ij}\big(a^{\mathcal{I}_i}\big)$. However, DDL permit arbitrary domain relations, and so, possibly, there are other elements in $r_{ij}\big(a^{\mathcal{I}_i}\big)$ besides for $b^{\mathcal{I}_j}$. If we wish to associate a certain individual with another one exactly and exclusively, we may do this using the total individual correspondence axiom, whose semantics requires $a^{\mathcal{I}_i} = \{{b_1}^{\mathcal{I}_j}, \ldots, {b_n}^{\mathcal{I}_j}\}$. Also individual correspondences possibly enable for knowledge reuse in the distributed ontology, as we demonstrate by an example.

Example 9. *Consider the DDL knowledge base $\mathfrak{T} =$ with two local ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. In both of these ontologies a particular person, John Smith, is represented: in $\mathcal{T}_1$ by the individual* e234780 *and in $\mathcal{T}_2$ by the individual* johnSmith. *The correspondence is recorded by an individual correspondence axiom:*

$$1 : \mathsf{e234780} \longmapsto 2 : \mathsf{johnSmith} \ .$$

The local ontology $\mathcal{T}_1$ is concerned with professions. In the data, it is recorded that John Smith is a junior executive:

$$1 : \mathsf{JuniorExecutive}(\mathsf{e234780}) \ .$$

The other local ontology is concerned with income and tax data. Individuals are sorted into three income classes: LowIncomeClass, MiddleIncomeClass *and* HighIncomeClass. *Based on the professions from $\mathcal{T}_2$, we are able to track these income classes. For instance, we have the following bridge rule:*

$$1 : \mathsf{JuniorExecutive} \xrightarrow{\sqsubseteq} 2 : \mathsf{MiddleIncomeClass} \ .$$

Based solely on the three axioms above, it is implied that the individual johnSmith *is amongst the instances of* MiddleIncomeClass.

The development of the framework has continued by introduction of bridge rules between roles (Ghidini & Serafini 2006*a*). The motivation of this extension is clear, with these new bridge rules also the role hierarchy is possibly reused between distinct ontology units. These are possibly into- or onto-bridge rules, very similar to those between concepts:

$$i : R \xrightarrow{\sqsubseteq} j : S \ , \qquad\qquad i : R \xrightarrow{\sqsupseteq} j : S \ ,$$

where R is an i-role and S is a j-role. Semantically, these behave just as bridge rules between concepts (i.e., the into-form requires $r_{ij}\big(R^{\mathcal{I}_i}\big) \subseteq S^{\mathcal{I}_j}$ and the onto-form requires $r_{ij}\big(R^{\mathcal{I}_i}\big) \supseteq S^{\mathcal{I}_j}$). The only difference is that the mapping $r_{ij}(\cdot)$ now projects a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ to a subset of $\Delta^{\mathcal{I}_j} \times \Delta^{\mathcal{I}_j}$.

One of the intuitions and intended uses of DDL is also resolving the modeling heterogeneity between ontology units and facilitating knowledge reuse even in cases in which each of the ontology units takes a completely different modeling approach. At the beginning of this chapter we have discussed a scenario in which an entity – a household – is represented by a single instance in one ontology but as multiple instances in another ontology. There are more complex scenarios to consider, in which knowledge is possibly reused. Ghidini & Serafini (2006*b*), Ghidini et al. (2007) suggest, that in certain cases the same or a very similar subsumption hierarchy is possibly modeled with subsumption of concepts in one ontology and with subsumption of roles in some other. In such a case, it makes sense to map between concepts from the first ontology and roles from the second. To cope with this scenario, so called heterogeneous bridge rules have been introduced. They are of four possible forms:

$$i : C \xrightarrow{\sqsubseteq} j : S \ , \qquad\qquad i : C \xrightarrow{\sqsupseteq} j : S \ ,$$
$$i : R \xrightarrow{\sqsubseteq} j : D \ , \qquad\qquad i : R \xrightarrow{\sqsupseteq} j : D \ ,$$

given an i-concept C, a j-concept D, an i-role R and a j-role S. Consider the following example, that we borrow from Ghidini et al. (2007).

Example 10. *Assume a DDL knowledge base $\mathfrak{T}$ which connects two local ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. Both these ontologies are concerned with social relationships of people. In $\mathcal{T}_1$, the notion of marriage is modeled with the concept* Marriage.

There are also other related concepts, such as civilUnion*, which is broader then* Marriage *and hence this ontology features the axiom:*

$$1 : \mathsf{Marriage} \sqsubseteq \mathsf{civilUnion} \ .$$

In the local ontology $\mathcal{T}_2$, the notion of marriage and related partnerships is modeled using roles. There are two roles marriedInChurchWith *and* partnerOf*. These are possibly both used in $\mathcal{T}_2$, there is no local record of any direct relation between them, however.*

With heterogeneous bridge rules we are able to relate these concepts and roles. In our modeling scenario, we have decided that the relationship represented with the role partnerOf *in $\mathcal{T}_2$ is equally represented by the concept* civilUnion *in $\mathcal{T}_1$. Therefore we add the two bridge rules:*

$$1 : \mathsf{civilUnion} \xrightarrow{\sqsubseteq} 2 : \mathsf{partnerOf} \ , \qquad 1 : \mathsf{civilUnion} \xrightarrow{\sqsupseteq} 2 : \mathsf{partnerOf} \ .$$

In addition, we understand that the 2-role marriedInChurchWith *is covered by the 1-concept* Marriage*, however, we are not sure, whether there is not also some other type of marriages possible, hence we just add the following onto-bridge rule:*

$$1 : \mathsf{Marriage} \xrightarrow{\sqsupseteq} 2 : \mathsf{marriedInChurchWith} \ .$$

If we take a closer look at the scenario which we have just modeled, we realize that it resembles the very basic subsumption propagation scenario of DDL (see Property 5), the only difference is that on one side we have concepts and on the other one we have roles. Indeed, the semantics of heterogeneous bridge rules is powerful enough and it propagates the subsumption between the two 1-concepts Marriage *and* civilUnion *onto the two 2-roles* marriedInChurchWith *and* partnerOf*.*

The semantics of heterogeneous bridge rules is more complex, and the extension of the framework is not straightforward. Especially, it is not sufficient any more to rely on simple domain relations $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$, and, in order to achieve the desired results, more complex domain relations are introduced. In addition, the local language on which DDL are built is changed from $\mathcal{SHIQ}$ to $\mathcal{ALCQI}b$ (Tobies 2001), a rich DL that allows complex role constructors such as role complement, role union, and role intersection. The point here is to compensate the imbalance between concepts and roles in more traditional DL that allow many operations with concepts but only very little operations with roles. This is desired since the concept hierarchy and the role hierarchy are now possibly densely interconnected. Since we do not research heterogeneous bridge rules in this thesis we refer the curious reader to the work of (Ghidini et al. 2008) for more details.

3.5 Summary

In this chapter we have reviewed the state of the art in Distributed Description Logics, a formalism for distributed ontological knowledge representation, as introduced in (Borgida & Serafini 2002) and consequently developed by further research.

DDL follow the ontology mapping paradigm and they formalize a distributed knowledge base, which allows local ontology units, expressed in DL, to be combined with new axioms called bridge rules. In the original proposal, bridge rules express semantic mapping between concepts (Borgida & Serafini 2002, 2003, Serafini & Tamilin 2004, Serafini et al. 2005). In more recent work, the framework has been extended and it allows to express also mapping between individuals (Serafini & Tamilin 2006, 2007), and mapping between roles (Ghidini & Serafini 2006*a*). Also, so called heterogeneous mappings between concepts on one hand and roles on the other one have been investigated (Ghidini & Serafini 2006*b*, Ghidini et al. 2007).

DDL continually prove to be an interesting formalism with semantics that is naturally built on top of the semantics of underlaying DL. DDL enable for knowledge reuse and the open the door for modular ontology development. The semantics of DDL satisfies a number of interesting properties, that are considered for distributed ontologies: DDL monotonically extend DL, knowledge reuse is directional, and hence one ontology may reuse another without affecting its knowledge. In addition, DDL are capable of dealing with accidental inconsistency, if it appears in one of the ontology units that are being combined. Also the mechanism of knowledge reuse is intuitive and well understood.

A distributed tableaux decision algorithm for DDL has been developed (Serafini & Tamilin 2004, Serafini et al. 2005) by extending the one known for $\mathcal{SHIQ}$. This algorithm has been implemented into the DRAGO reasoner (Serafini & Tamilin 2005). The reasoner also permits a fully distributed mode, when the reasoning task is distributed between multiple local reasoners, that communicate by using queries.

While an outstanding number of research reports has been published on DDL since their inception, there are still many interesting challenges and open issues. Some of these we address later on in this thesis. We address the problem of subsumption propagation on complex concepts in Chap. 4 and the problem of subsumption propagation between remote ontologies in Chap. 5. In these two chapters, we build on the basic DDL framework, as introduced in the current chapter. Later on, in Chap. 6, we formalize a richer version of DDL, including partial individual correspondences and bridge rules between roles, and consequently we compare the expressive power of DDL with other distributed and modular ontology representation frameworks.

Other interesting open issues connected to DDL, that we do not address in this thesis, include: lifting the local language to ever more expressive flavours of DL such as $\mathcal{SHOIQ}$ (Horrocks & Sattler 2005) and $\mathcal{SROIQ}$ (Horrocks et al. 2006), that include nominals and other constructs; deeper investigation of heterogeneous bridge rules and development of a reasoning algorithm for DDL with heterogeneous bridge rules; investigation of computational complexity of the reasoning tasks in the DDL settings; optimization; and others.

Chapter 4

Subsumption Propagation and Complex Concepts in DDL

Our attention is now diverted towards subsumption propagation – the basic mechanism of knowledge reuse within DDL knowledge bases. In general, subsumption propagation is characterized by the following scenario: given a DDL knowledge base $\mathfrak{T}$ with two local ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$, $i \neq j$, let there be concepts C and D within $\mathcal{T}_i$ which are not related in terms of local subsumption, and let there be concepts G and H within $\mathcal{T}_j$ which are subsumed, one by another; if there are relevant bridge rules between G and H, on one side, and C and D, on the other side, that result into subsumption being entailed between C and D when reasoning with the whole knowledge base $\mathfrak{T}$, we say that subsumption propagation has occurred.

Of course, it is desired that subsumption propagation must respect the direction of bridge rules that are involved. This is formally stated by the directionality property (Property 2, see Sect. 3.2). Hence, we have indeed talked about bridge rules directed from $\mathcal{T}_j$ to T_i in the scenario above.

A typical example of subsumption propagation occurs in Fig. 4.2. In our distributed knowledge base $\mathfrak{T}$ there are two ontologies. $\mathcal{T}_1$ is the generic ontology of wines (depicted on the left), the other ontology $\mathcal{T}_2$ is concerned with a specific kind of wines from the Tokaj region in central Europe (depicted on the right). In $\mathcal{T}_1$, the subsumption between Tokaji_1 and $\mathsf{DesertWine}_1$ is locally entailed. Due to the bridge rules, this subsumption propagates into $\mathcal{T}_2$, and the distributed knowledge base, as a whole, entails also the subsumption between Tokaji_2 and $\mathsf{SweetWine}_2$ within $\mathcal{T}_2$, which is not entailed locally.

Subsumption propagation, as observed above, has been described as desired and one of the main features of DDL and it has been studied by Borgida & Serafini (2003), Serafini & Tamilin (2004) and Serafini et al. (2005). It has been shown in these works that the scenario depicted in Fig. 4.2 holds in general, as we have learned in Sect. 3.2 (Properties 5 and 6).

In this thesis, we will learn, that in certain cases the amount of subsumption propagation under the original DDL semantics is not sufficient. In this chapter, we study subsumption propagation between complex concepts, such that

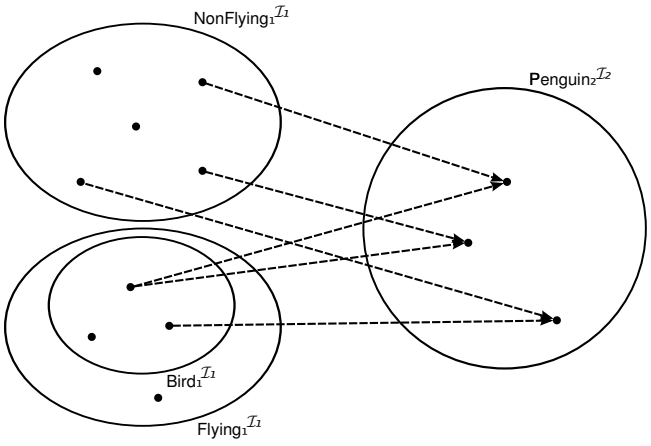

Figure 4.1: Depiction of a distributed model of the distributed TBox $\mathfrak{T}$ from Example 11 in which $\mathsf{Penguin}_2^{\mathcal{I}_2}$ is nonempty. Elements of the local domain $\Delta^{\mathcal{I}_i}$ are on the left hand side, elements of $\Delta^{\mathcal{I}_j}$ on the right hand side. The domain relation r_{ij} is depicted by dashed arrows

bridge rules are expressed between concepts from which these complex concepts are constructed. As an example recall the generalized subsumption property (Property 6) from Sect. 3.2. In this case, subsumption is entailed between G and the complex concept $H_1 \sqcup \cdots \sqcup H_n$, even if the bridge rules involved do not map to the concept $H_1 \sqcup \cdots \sqcup H_n$ directly, but instead there are multiple bridge rules involving the concepts $H_1, \ldots, H_n$. Such subsumption propagation is intuitively expected. However, we will see in this chapter, that while the concept union constructor allows subsumption propagation to the desired extent, this is not always the case for the concept intersection constructor, as we intuitively expect.

In order to cope with this issue, we propose and study two different semantic extensions of DDL. In the first one we introduce a new kind of bridge rules, called conjunctive bridge rules, with amended semantics. The second semantics does not introduce new syntactic constructs, but instead it places further restrictions on the domain relation within each distributed model. Both of these solutions deal with the problem and increase subsumption propagation in situations involving concept intersection.

4.1 Analysis

Our investigation is sparked by the work of Cuenca Grau et al. (2004*b*), where a modeling scenario is outlined in which DDL behave counterintuitively. Let us recall this scenario in the following example.

Example 11 (Cuenca Grau et al. (2004*b*))**.** *Consider the ontology represented by the following TBox $\mathcal{T}$:*

$$\mathsf{NonFlying} \equiv \neg\mathsf{Flying}\ , \qquad \mathsf{Penguin} \sqsubseteq \mathsf{Bird}\ ,$$
$$\mathsf{Bird} \sqsubseteq \mathsf{Flying}\ , \qquad \mathsf{Penguin} \sqsubseteq \mathsf{NonFlying}\ .$$

Consider also $\mathfrak{T}$, the distributed counterpart of $\mathcal{T}$, divided into two local TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$. The local TBox $\mathcal{T}_1$ is listed on the left below. The local ontology $\mathcal{T}_2$ contains only a single concept $\mathsf{Penguin}_2$ and it does not contain any axioms. Finally, there are also two bridge rules, listed on the right:

$$1 : \mathsf{NonFlying}_1 \equiv \neg \mathsf{Flying}_1 \ , \qquad 1 : \mathsf{Bird}_1 \xrightarrow{\sqsupseteq} 2 : \mathsf{Penguin}_2 \ ,$$

$$1 : \mathsf{Bird}_1 \sqsubseteq \mathsf{Flying}_1 \ . \qquad 1 : \mathsf{NonFlying}_1 \xrightarrow{\sqsupseteq} 2 : \mathsf{Penguin}_2 \ .$$

As it has been argued by Cuenca Grau et al. (2004b), while the concept Penguin *of $\mathcal{T}$ is not satisfiable, the corresponding concept $\mathsf{Penguin}_2$ of $\mathcal{T}_2$ is. The problem is that, in a perfectly sane distributed model of $\mathfrak{T}$, it is possible that ${\mathsf{Penguin}_2}^{\mathcal{I}_2}$ is nonempty. One such a model is depicted in Fig. 4.1. In such a model, each instance $x \in {\mathsf{Penguin}_2}^{\mathcal{I}_2}$ has been assigned by r to two distinct elements of $\Delta^{\mathcal{I}_1}$, say y_1 and y_2, one of them instance of Bird_1 and the other one of $\mathsf{NonFlying}_1$. Note that this is possible even if ${\mathsf{Bird}_1}^{\mathcal{I}_1}$ and ${\mathsf{NonFlying}_1}^{\mathcal{I}_1}$ are disjoint as required by the TBox $\mathcal{T}_1$. Cuenca Grau et al. (2004b) point out, that it is intuitive to expect that bridge rules retain certain basic properties that GCI axioms have, as we may think of bridge rules as of inter-ontology subsumption axioms. So, we would expect $\mathsf{Penguin}_2$ to be unsatisfiable, as we have made it a "subconcept of two imported concepts"* Bird_1 *and* $\mathsf{NonFlying}_1$ *which in their source ontology $\mathcal{T}_1$ are disjoint.*

Let us now proceed by generalizing this simple scenario a bit further. We will see that the problem has something to do with the concept intersection constructor. Consider a distributed TBox $\mathfrak{T}$ with two local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$. There are two onto-bridge rules from i to j, $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$ and $i : D \xrightarrow{\sqsupseteq} j : H \in \mathfrak{B}$. This situation generalizes the one of Example 11, where G and H are both equal to $\mathsf{Penguin}_2$. The problem is marked by the observation that the inclusion $(G \sqcap H)^{\mathcal{I}_j} \subseteq r_{ij}\Big((C \sqcap D)^{\mathcal{I}_i}\Big)$ does not necessarily hold in every model of $\mathfrak{T}$, as we would have expected. The source of our intuition here is indeed the fact that the respective inclusion $(G \sqcap H)^{\mathcal{I}} \subseteq (C \sqcap D)^{\mathcal{I}}$ holds in every model $\mathcal{I}$ in the case when C, D, G and H are all concepts of some monolithic knowledge base $\mathcal{T}$, and instead of the bridge rules we have two GCI axioms $G \sqsubseteq C \in \mathcal{T}$ and $H \sqsubseteq D \in \mathcal{T}$.

In order to motivate this generalization by some less abstract reasoning scenario, we extend the example depicted in Fig. 4.2. In this example there are two ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. The first one is a more general wine ontology dealing with abstract concepts such as Wine_1, $\mathsf{DryWine}_1$, $\mathsf{DessertWine}_1$, $\mathsf{Selection}_1$ and also with classification of wines based on their origin, such as $\mathsf{Beaujolais}_1$, Port_1, Tokaji_1 and others. The second one deals exclusively with wines from the Tokaj region, hence it contains concepts for the specific kinds of Tokaji wine: $\mathsf{FivePuttony}_2$, $\mathsf{SixPuttony}_2$, $\mathsf{Frumint}_2$, $\mathsf{Szamorodni}_2$, etc.; but also a few more generic concepts such as Wine_2, Tokaji_2, etc. In Fig. 4.2 we have asserted the axiom $\mathsf{Tokaji}_1 \sqsubseteq \mathsf{DessertWine}_1$ in $\mathcal{T}_1$ because most of the Tokaji wines are from this category. Now we would like to amend this modeling, as we have learned that apart from most Tokaji wines which are sweet such as the famous Tokaji selections, there are also some Tokaji wines which are not necessarily sweet, such as the Szamorodni wine. We will use the key: if the wine is a selection wine from Tokaj region then it is for sure a dessert wine. Let us remodel this example more formally.

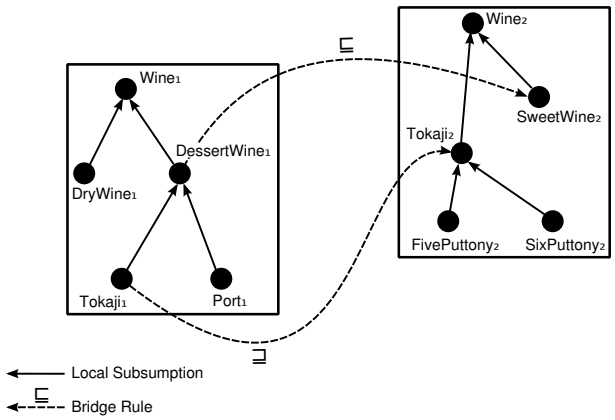

Figure 4.2: Simple subsumption propagation example. The subsumption between Tokaji_1 and $\mathsf{DesertWine}_1$ propagates, and it is entailed also between Tokaji_2 and $\mathsf{SweetWine}_2$

Example 12. *Consider the distributed TBox $\mathfrak{T}$ depicted in Fig. 4.3. $\mathfrak{T}$ consists of two TBoxes $\mathcal{T}_1$ (general wine ontology) and $\mathcal{T}_2$ (Tokaji wine ontology). In $\mathcal{T}_1$, we have axioms as follows:*

$$1 : \mathsf{DryWine}_1 \sqsubseteq \mathsf{Wine}_1 \ , \quad 1 : \mathsf{DessertWine}_1 \sqsubseteq \mathsf{Wine}_1 \ ,$$
$$1 : \mathsf{Tokaji}_1 \sqsubseteq \mathsf{Wine}_1 \ , \quad 1 : \mathsf{Tokaji}_1 \sqcap \mathsf{Selection}_1 \sqsubseteq \mathsf{DessertWine}_1 \ .$$

In $\mathcal{T}_2$, we have axioms such as:

$$2 : \mathsf{Tokaji}_2 \sqsubseteq \mathsf{Wine}_2 \ , \quad 2 : \mathsf{TokajiSelection}_2 \sqsubseteq \mathsf{Tokaji}_2 \sqcap \mathsf{Selection}_2 \ ,$$
$$2 : \mathsf{FivePuttony}_2 \sqsubseteq \mathsf{TokajiSelection}_2 \ , \quad 2 : \mathsf{SixPuttony}_2 \sqsubseteq \mathsf{TokajiSelection}_2 \ ,$$
$$2 : \mathsf{Frumint}_2 \sqsubseteq \mathsf{Tokaji}_2 \ , \quad 2 : \mathsf{Szamorodni}_2 \sqsubseteq \mathsf{Tokaji}_2 \ .$$

With this modeling we capture the fact that not all, but only specific wines from Tokaj are dessert wines. This holds inside $\mathcal{T}_1$. Now we want to introduce the concept of sweet wine into $\mathcal{T}_2$, since most Tokaji wines are indeed sweet. Since similar concept already exists in $\mathcal{T}_1$ ($\mathsf{DessertWine}_1$), we would like to map between these concepts and reuse the knowledge in $\mathcal{T}_1$ also in $\mathcal{T}_2$, as it worked in the simpler case represented by Fig. 4.2. Hence we add these bridge rules:

$$1 : \mathsf{Tokaji}_1 \xrightarrow{\sqsupseteq} 2 : \mathsf{Tokaji}_2 \ , \quad 1 : \mathsf{DessertWine}_1 \xrightarrow{\sqsubseteq} 2 : \mathsf{SweetWine}_2 \ ,$$
$$1 : \mathsf{Selection}_1 \xrightarrow{\sqsupseteq} 2 : \mathsf{Selection}_2 \ .$$

Intuitively, we would now expect that $\mathfrak{T} \models_\epsilon 2 : \mathsf{SixPuttony} \sqsubseteq \mathsf{SweetWine}$. Unfortunately, this is not the caste. The problem that prevents from such an entailment to be proven, is the same as already described above: there are distributed models of $\mathfrak{T}$ in which $(\mathsf{Tokaji}_2 \sqcap \mathsf{Selection}_2)^{\mathcal{I}_2} \not\subseteq r_{12}\left((\mathsf{Tokaji}_1 \sqcap \mathsf{Selection}_1)^{\mathcal{I}_1}\right)$, and hence the entailment cannot be proven.

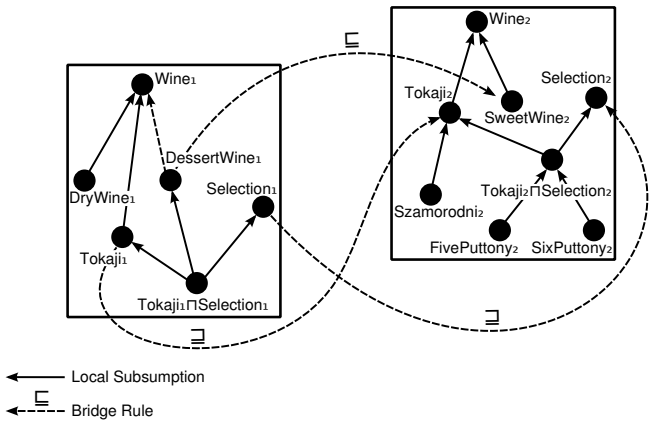

Figure 4.3: Distributed TBox $\mathfrak{T}$ from Example 12. The general wine TBox $\mathcal{T}_1$ is depicted on the left hand side; $\mathcal{T}_2$, that deals with Tokaji wines on the right hand side

Pushing the generalization further, we realize that such complex subsumption propagation patterns are intuitive also in cases when more than two onto-bridge rules are involved. If there are several concepts in one local TBox $\mathcal{T}_i$, say $C_1, \ldots, C_n$, and another bunch of concepts in $\mathcal{T}_j$, say $D_1, \ldots, D_n$, which are mapped between by bridge rules, C_i to D_i for all i, $1 \leq i \leq n$, it is intuitive to expect that this should have impact on complex concepts formed from $C_1 \ldots, C_n$ and from $D_1, \ldots, D_n$. Of course, this is only justified in cases when the bridge rules between C_i and D_i are all of the same direction and of the same kind.

We will soon learn, that this issue is related to onto-bridge rules only, and the problem does not arise in case of into-bridge rules (see Theorem 55 below). In the following, we introduce an alternative kind of onto-bridge rules with slightly modified semantics. We consecutively show, that this semantics follows the intuitions outlined above.

4.2 Complex Concepts and Subsumption Propagation in Original DDL

Let us now verify, how much of the desired properties related on subsumption propagation on complex concepts, as we have described and motivated in the previous section, are in fact retained by the original DDL semantics. That is, we will be interested in cases, when there are $n > 1$ bridge rules in the knowledge base, for each $1 \leq k \leq n$ the respective bridge rule maps between C_k from $\mathcal{T}_i$ and D_k from $\mathcal{T}_j$, and each of them is of the same direction and of the same type, given $i, j \in I$.

This breaks down into four cases. There are two types of bridge rules, and for each type we are interested in propagation of subsumption over concept union and over concept intersection predicate. Out of this cases, the one involving into-bridge rules and concept union has been studied. This we have already

learned in Sect. 3.2. Recall the generalized subsumption propagation property (Property 6), which exactly describes this kind of propagation pattern. More precisely, in Property 6 we deal with a knowledge base $\mathfrak{T}$ with a bridge rule $i : D \xrightarrow{\sqsupseteq} j : H$, a subsumption $i : D \sqsubseteq C_1 \sqcup \cdots \sqcup C_n$ and series of bridge rules $i : C_k \xrightarrow{\sqsubseteq} j : G_k$, for each $1 \leq k \leq n$. In such a case, we are always able to derive that $\mathfrak{T} \models_\epsilon j : H \sqsubseteq G_1 \sqcup \cdots \sqcup G_n$.

An interesting point to observe is that given the n bridge rules $i : C_k \xrightarrow{\sqsubseteq} j : G_k$, for $1 \leq k \leq n$, the semantics behaves as if a bridge rule of the same kind was *implied*[1] between the two unions of concepts $\bigsqcup_{k=1}^{n} C_n$ and $\bigsqcup_{k=1}^{n} D_n$. This is the reason behind this particular subsumption propagation mechanism. This phenomenon is formally described by the following theorem.

Theorem 53. *Given a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$, if for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsubseteq} j : G_n$ are all part of $\mathfrak{B}$, then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that:*

$$r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big) \subseteq (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j} \quad .$$

Proof. Let $\mathfrak{T}$ be a distributed TBox with an index set I and set of bridge rules $\mathfrak{B}$. Let $\mathcal{T}_i$ and $\mathcal{T}_j$, $i \neq j$, be two local TBoxes of $\mathfrak{T}$ and let there be bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsubseteq} j : G_n$ in $\mathfrak{B}$. Assume that $x \in r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big)$. We shall prove that $x \in (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j}$.

If $x \in r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big)$ that there is some $y \in (C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}$ such that $x \in r_{ij}(y)$. Since $(C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i} = {C_1}^{\mathcal{I}_i} \cup \cdots \cup {C_n}^{\mathcal{I}_i}$ then there is k, $1 \leq k \leq n$, such that $y \in {C_k}^{\mathcal{I}_i}$. But we have assumed the existence of the bridge rule $i : C_k \xrightarrow{\sqsubseteq} j : G_k$ in $\mathfrak{B}$. This implies that $r_{ij}\big({C_k}^{\mathcal{I}_i}\big) \subseteq {G_k}^{\mathcal{I}_j}$. And so, $x \in {G_k}^{\mathcal{I}_j}$, which in turn means that $x \in {G_1}^{\mathcal{I}_j} \cup \ldots \cup {G_n}^{\mathcal{I}_j} = (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j}$. □

And hence we conclude, that the case when we consider into-bridge rules and the concept union constructor poses no problem for subsumption propagation. Indeed, this is a known result, as Property 6 has been already proven by Serafini & Tamilin (2004). We will now show that similar characterization holds also in the second case, when onto-bridge rules and the concept union constructor are involved.

Theorem 54. *Given a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$, if for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsupseteq} j : G_n$ are all part of $\mathfrak{B}$, then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it follows that:*

$$r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big) \supseteq (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j} \quad .$$

Proof. Let $\mathfrak{T}$ be a distributed TBox with an index set I and set of bridge rules $\mathfrak{B}$. Let $\mathcal{T}_i$ and $\mathcal{T}_j$, $i \neq j$, be two local TBoxes of $\mathfrak{T}$ and let there be bridge

[1] Here we touch the notion of mapping entailment, that is, introducing a new decision problem of a bridge rule being entailed by a knowledge base. We deliberately choose not to introduce this notion, as in this thesis we don't intend to explore this decision task further. To our best knowledge, this issue is also vacant in the available literature and it opens an interesting direction of research in DDL.

rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1, \ldots, i : C_n \xrightarrow{\sqsupseteq} j : G_n$ in $\mathfrak{B}$. Suppose that $y \in (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j}$. We ought to prove that $y \in r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big)$.

Since $(G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j} = {G_1}^{\mathcal{I}_j} \cup \ldots \cup {G_n}^{\mathcal{I}_j}$ then there is some k, $1 \leq k \leq n$, such that $y \in {G_k}^{\mathcal{I}_j}$. From the assumptions we know that there is a bridge rule $i : C_k \xrightarrow{\sqsupseteq} j : G_k \in \mathfrak{B}$. The existence of this bridge rule implies ${G_k}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_k}^{\mathcal{I}_i}\big)$, and hence there is $x \in {C_k}^{\mathcal{I}_i}$ such that $y \in r_{ij}(x)$. However, it also holds that $x \in {C_1}^{\mathcal{I}_i} \cup \ldots \cup {C_n}^{\mathcal{I}_i} = (C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}$, and hence $y \in r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big)$, what we wanted to prove. □

The very next question that appears in context of the above findings is, whether some corresponding properties also hold for the concept conjunction constructor. We first show that indeed for into-bridge rules an analogous characterization holds.

Theorem 55. *Given a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$, if for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : G_1, \ldots, i : C_n \xrightarrow{\sqsubseteq} j : G_n$ are all part of $\mathfrak{B}$, then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that:*

$$r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big) \subseteq (G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \ .$$

Proof. Let $\mathfrak{T}$ be some distributed TBox with an index set I and a set of bridge rules $\mathfrak{B}$, and with two distinct local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$. Let there be bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : G_1, \ldots, i : C_n \xrightarrow{\sqsubseteq} j : G_n$ in $\mathfrak{B}$. Let $\mathfrak{I}$ be some distributed model of $\mathfrak{T}$.

From the definition of the mapping r_{ij} and from the basic properties of intersection we get $r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_n}^{\mathcal{I}_i}\big)$. Also $r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_n}^{\mathcal{I}_i}\big) \subseteq {G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j}$ holds, as we have $r_{ij}\big({C_k}^{\mathcal{I}_i}\big) \subseteq {G_k}^{\mathcal{I}_j}$, for each $1 \leq k \leq n$, which follows from the assumption that rule $i : C_k \xrightarrow{\sqsubseteq} j : G_k$ is part of $\mathfrak{B}$. Together this amounts to $r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big) \subseteq {G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j}$. Directly from the definition of distributed interpretation, the last statement equals to $r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big) \subseteq (G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j}$, and hence the theorem. □

And so we have proven that in three of the four cases, the original DDL semantics performs satisfactory with respect to propagation of subsumption towards more complex concepts. As we have already learned from Examples 11 and 12, this does not hold in the final fourth case that involves onto-bridge rules and the concept intersection constructor. The reason for this seems to be the fact, that allowing simply any relation on the local domains as domain relation in distributed models is perhaps overly permissive. Later on in this chapter, we will propose and study two possible solutions for this problem.

So far we have analysed the four basic cases. Particularly, we have limited our consideration on complex concepts constructed by repeated application of the same constructor, either concept union, or concept intersection. In fact, as we will show, the semantic implications of bridge rules propagate on complex concepts constructed by arbitrary combination of the concept intersection and the concept union constructor.

Theorem 56. *Assume a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$. Let for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsubseteq} j : G_n$ be all part of $\mathfrak{B}$. Let there be two complex concepts, an i-concept E and a j-concept F, constructed as follows:*

1. *E is constructed from the n concepts $C_1, \ldots, C_n$ using any combination of concept intersection and concept union constructors, and no other constructors;*

2. *F is obtained from E by replacing each occurrence of C_i by G_i, for each i, such that $1 \leq i \leq n$.*

Then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that $r_{ij}\big(E^{\mathcal{I}_i}\big) \subseteq F^{\mathcal{I}_j}$.

Proof. By structural induction on the structure of E. In the base case there are two options. Either $E = C_i \sqcup C_j$ and $F = G_i \sqcup G_j$, for some $i, j \in \{1, \ldots, n\}$ and then the base case follows from Theorem 53, or $E = C_i \sqcap C_j$ and $F = G_i \sqcap G_j$, for some $i, j \in \{1, \ldots, n\}$ and then the base case follows from Theorem 55.

In the induction step, there are also two cases. In the first case, $E = E_1 \sqcup E_2$ and $F = F_1 \sqcup F_2$ and we know from the induction hypothesis, that $r_{ij}\big({E_1}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j}$ and $r_{ij}\big({E_2}^{\mathcal{I}_i}\big) \subseteq {F_2}^{\mathcal{I}_j}$. This implies $r_{ij}\big({E_1}^{\mathcal{I}_i}\big) \cup r_{ij}\big({E_2}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j} \cup {F_2}^{\mathcal{I}_j}$. The projection of the union of two sets is always equal to the union of their projections and hence $r_{ij}\big({E_1}^{\mathcal{I}_i} \cup {E_2}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j} \cup {F_2}^{\mathcal{I}_j}$. Finally, from the definition of the interpretation of complex concepts we have $r_{ij}\big(E^{\mathcal{I}_i}\big) \subseteq r_{ij}\Big((E_1 \sqcup E_2)^{\mathcal{I}_i}\Big) \subseteq (F_1 \sqcup F_2)^{\mathcal{I}_j} \subseteq F^{\mathcal{I}_j}$.

The second case is very similar. We have $E = E_1 \sqcap E_2$ and $F = F_1 \sqcap F_2$ and we know from the induction hypothesis, that $r_{ij}\big({E_1}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j}$ and $r_{ij}\big({E_2}^{\mathcal{I}_i}\big) \subseteq {F_2}^{\mathcal{I}_j}$. This implies $r_{ij}\big({E_1}^{\mathcal{I}_i}\big) \cap r_{ij}\big({E_2}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j} \cap {F_2}^{\mathcal{I}_j}$. The projection of the intersection of two sets is always a subset of the intersection of their projections, and hence $r_{ij}\big({E_1}^{\mathcal{I}_i} \cap {E_2}^{\mathcal{I}_i}\big) \subseteq {F_1}^{\mathcal{I}_j} \cap {F_2}^{\mathcal{I}_j}$. Finally, from the definition of the interpretation of complex concepts we have $r_{ij}\big(E^{\mathcal{I}_i}\big) \subseteq r_{ij}\Big((E_1 \sqcap E_2)^{\mathcal{I}_i}\Big) \subseteq (F_1 \sqcap F_2)^{\mathcal{I}_j} \subseteq F^{\mathcal{I}_j}$. □

A counterpart of this proposition for onto-bridge rules cannot be proven under the original DDL semantics, since the concept intersection constructor is involved here.

4.3 Conjunctive Bridge Rules

We have learned, that if multiple bridge rules between two local ontologies are present in a DDL knowledge base, they possibly interact and this interaction may have an impact on complex concepts build from the concepts that take part in these bridge rules. We have also learned that this behaviour is not exhibited by all kinds of bridge rules and all kinds of eligible concept constructors evenly. We have shown that onto-bridge rules do not interact with concept intersection to the same extent as they do interact with concept union. Into-bridge rules do interact with both, concept intersection and concept union.

We agree with Cuenca Grau et al. (2004*b*) that such an uneven behaviour can be considered a drawback of the original DDL semantics. In our point of

view, in the modeling scenarios presented in Examples 11 and 12, it is very intuitive to expect the entailment which, as we have pointed out, the original DDL semantics does not support.

In a sense, the original semantics of bridge rules can be seen as too weak, as we have seen in Example 11, it permits too many models. If we are to overcome this issue, the semantics has to be strengthened. We will now propose a new kind of onto-bridge rules, which we call *conjunctive*. We will also call the original form of bridge rules *normal* in this chapter, in order to distinguish between these two forms.

Definition 68 (Conjunctive bridge rules)**.** *Given a DDL knowledge base $\mathfrak{T}$ with an index set I and a set of bridge rules $\mathfrak{B}$, and given $i, j \in I$, $i \neq j$, an i-concept C and a j-concept G, a conjunctive onto-bridge rule is an axiom of the following form:*

$$i : D \xrightarrow{\sqsupseteq} j : H \quad .$$

In the following we sometimes state propositions that are valid for both kind of bridge rules equally. For this reason, we also use $i : D \overset{\sqsupseteq}{\rightsquigarrow} j : H$ to denote onto-bridge rules that are possibly of any of the two kinds, either conjunctive or normal.

Our amendment to the DDL framework is performed by allowing also conjunctive onto-bridge rules in the distributed TBox.

Definition 69 (DDL knowledge base with conjunctive bridge rules)**.** *Assuming a nonempty index set I, and some indexed vocabulary of atomic concept names $N_{\mathrm{C}} = \{N_{\mathrm{C}i}\}_{i \in I}$, role names $N_{\mathrm{R}} = \{N_{\mathrm{R}i}\}_{i \in I}$, and individual names $N_{\mathrm{I}} = \{N_{\mathrm{I}i}\}_{i \in I}$, a DDL knowledge base with conjunctive bridge rules as a pair $\mathfrak{T} = \langle \{\mathcal{T}_i\}, \mathfrak{B} \rangle$ where:*

1. *$\mathcal{T}_i$, $i \in I$, is a knowledge base in some sublanguage of $\mathcal{SHIQ}$, which possibly contains GCI and RIA axioms, transitivity axioms and ABox assertions;*
2. *$\mathfrak{B} = \bigcup_{i,j \in I, i \neq j} \mathfrak{B}_{ij}$, such that for each $i, j \in I$, $\mathfrak{B}_{ij}$ contains any number of into-bridge rules, normal onto-bridge rules and conjunctive onto-bridge rules from i to j.*

Our adjusted DDL framework will use the same notion of interpretation, as the original one (Definition 58), however we need to define satisfiability of conjunctive onto-bridge rules. This is done by adding a new clause into Definition 59. For of sake convenience we copy the definition here, note the added clause number 5. As we will see, the semantics reflects the fact that interaction between bridge rules must be captured by additional constraints, and it directly corresponds to our findings from Sect. 4.1.

Definition 70 (Distributed model)**.** *For every i and j, a distributed interpretation $\mathfrak{I}$ satisfies the elements of a distributed TBox $\mathfrak{T}$ (denoted by $\mathfrak{I} \models_\epsilon \cdot$) according to the following clauses:*

1. *$\mathfrak{I} \models_\epsilon i : C \sqsubseteq D$ if $\mathcal{I}_i \models C \sqsubseteq D$;*
2. *$\mathfrak{I} \models_\epsilon \mathcal{T}_i$ if $\mathcal{I}_i$ is a model of $\mathcal{T}_i$;*
3. *$\mathfrak{I} \models_\epsilon i : C \xrightarrow{\sqsubseteq} j : G$ if $r_{ij}\big(C^{\mathcal{I}_i}\big) \subseteq G^{\mathcal{I}_j}$;*

4. $\mathfrak{I} \models_\epsilon i : C \xrightarrow{\sqsupseteq} j : G$ *if* $r_{ij}\big(C^{\mathcal{I}_i}\big) \supseteq G^{\mathcal{I}_j}$;
5. $\mathfrak{I} \models_\epsilon i : C \xrightarrow[\Rightarrow]{\sqsupseteq} j : G$ *if, given any other* $n \geq 0$ *conjunctive onto-bridge rules* $i : D_1 \xrightarrow[\Rightarrow]{\sqsupseteq} j : H_1$, ..., $i : D_n \xrightarrow[\Rightarrow]{\sqsupseteq} j : H_n$ *that also belong to* $\mathfrak{B}_{ij}$, *we have* $r_{ij}\big(C^{\mathcal{I}_i} \cap {D_1}^{\mathcal{I}_i} \cap \cdots \cap {D_n}^{\mathcal{I}_i}\big) \supseteq G^{\mathcal{I}_j} \cap {H_1}^{\mathcal{I}_j} \cap \cdots \cap {H_n}^{\mathcal{I}_j}$;
6. $\mathfrak{I} \models_\epsilon \mathfrak{B}$, *if* $\mathfrak{I}$ *satisfies all bridge rules in* $\mathfrak{B}$;
7. *finally,* $\mathfrak{I}$ *is said to be a distributed model of* $\mathfrak{T}$ *(denoted by* $\mathfrak{I} \models_\epsilon \mathfrak{T}$*), if* $\mathfrak{I} \models_\epsilon \mathfrak{B}$ *and* $\mathfrak{I} \models_\epsilon \mathcal{T}_i$ *for each* i.

Our choice of adding new kind of onto-bridge rules instead of simply replacing the old semantics is to underline the fact that both kinds can coexist and be used according to the modeling scenario and the intentions of the ontology editor. Also, it is not yet clear, how the use of conjunctive bridge rules affects the computational complexity of the framework. It surely introduces a significant number of additional conditions to verify. Hence, it might be desirable to be allowed to choose the exact kind of a bridge rule according to the modeling scenario.

We now proceed with exploring basic properties of conjunctive bridge rules. Our first observation is that conjunctive bridge rules are strictly stronger, than normal bridge rules. That is, all the semantic implications caused by normal bridge rules are also in effect if conjunctive bridge rules are used instead. With conjunctive bridge rules, we have some more implications in addition. This follows rather straightforward from the definition, as it was indeed our intention and the definition reflects it.

Theorem 57. *Assume a DDL knowledge base* $\mathfrak{T}$ *with an index set* I *and a set of bridge rules* $\mathfrak{B}$. *If there is* $i : C \xrightarrow[\Rightarrow]{\sqsupseteq} j : G \in \mathfrak{B}$ *between two local TBoxes* $\mathcal{T}_i$ *and* $\mathcal{T}_j$, $i, j \in I$, $i \neq j$, *then in each distributed model* $\mathfrak{I}$ *of* $\mathfrak{T}$ *it holds that* $G^{\mathcal{I}_j} \subseteq r_{ij}\big(C^{\mathcal{I}_i}\big)$.

Proof. Let $\mathfrak{T}$ and $i : C \xrightarrow[\Rightarrow]{\sqsupseteq} j : G \in \mathfrak{B}$ be as in the assumptions of the theorem. Let $\mathfrak{I}$ be any distributed model of $\mathfrak{T}$. Since $\mathfrak{I} \models_{\mathrm{d}} i : C \xrightarrow[\Rightarrow]{\sqsupseteq} j : G$, we know from Definition 70, that for any $n \geq 0$ other conjunctive bridge rules $i : D_1 \xrightarrow[\Rightarrow]{\sqsupseteq} j : H_1$, ..., $i : D_n \xrightarrow[\Rightarrow]{\sqsupseteq} j : H_n$ that belong to $\mathfrak{B}_{ij}$ it holds that $G^{\mathcal{I}_j} \cap {H_1}^{\mathcal{I}_j} \cap \cdots \cap {H_n}^{\mathcal{I}_j} \subseteq r_{ij}\big(C^{\mathcal{I}_i} \cap {D_1}^{\mathcal{I}_i} \cap \cdots \cap {D_n}^{\mathcal{I}_i}\big)$. In case when $n = 0$ this gives us $G^{\mathcal{I}_j} \subseteq r_{ij}\big(C^{\mathcal{I}_i}\big)$. □

Our main interest with conjunctive bridge rules is the desired effect on complex concepts formed by concept intersection. More specifically, if we bridge between several pairs of concepts with conjunctive onto-bridge rules, say $i : C_1 \xrightarrow[\Rightarrow]{\sqsupseteq} j : G_1$, ..., $i : C_n \xrightarrow[\Rightarrow]{\sqsupseteq} j : G_n$, then the semantic implications posed by bridge rules do propagate on the two intersections of these concepts $C_1 \sqcap \cdots \sqcap C_n$ and $G_1 \sqcap \cdots \sqcap G_n$. This does not hold for normal bridge rules however, as demonstrated by Examples 11 and 12. For conjunctive onto-bridge rules this property is a straightforward consequence of the definition.

Theorem 58. *Given a distributed TBox* $\mathfrak{T}$ *with a set of bridge rules* $\mathfrak{B}$ *and some local TBoxes* $\mathcal{T}_i$ *and* $\mathcal{T}_j$ *such that* $i \neq j$, *if for some* $n > 0$, *the bridge rules* $i : C_1 \xrightarrow[\Rightarrow]{\sqsupseteq} j : G_1$, ..., $i : C_n \xrightarrow[\Rightarrow]{\sqsupseteq} j : G_n$ *are all part of* $\mathfrak{B}$, *then in every distributed model* $\mathfrak{I}$ *of* $\mathfrak{T}$ *it holds that:*

$$r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big) \supseteq (G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \quad .$$

Proof. Let us have a distributed TBox $\mathfrak{T}$ with n bridge rules $i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1$, $\ldots$, $i : C_n \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_n$ in $\mathfrak{B}$, as assumed by the theorem. Given any model $\mathfrak{I}$ of $\mathfrak{T}$, since $\mathfrak{I} \models_\epsilon i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1$, then from Definition 70 trivially we get ${G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$. Which by the semantics of complex concepts is equivalent to $(G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \subseteq r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big)$. □

Of course, conjunctive bridge rules perform equally well as normal bridge rules considering the interaction of bridge rules and the concept union constructor. This is formally stated by the following corollary, a straightforward consequence of Theorems 54 and 57.

Corollary 59. *Given a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$, if for some $n > 0$, the bridge rules $i : C_1 \stackrel{\sqsupseteq}{\rightsquigarrow} j : G_1, \ldots, i : C_n \stackrel{\sqsupseteq}{\rightsquigarrow} j : G_n$ are all part of $\mathfrak{B}$, then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that*

$$r_{ij}\Big((C_1 \sqcup \cdots \sqcup C_n)^{\mathcal{I}_i}\Big) \supseteq (G_1 \sqcup \cdots \sqcup G_n)^{\mathcal{I}_j} \quad .$$

In Sect. 4.2 we have learned, that with into-bridge rules a sufficient extent of semantic implications propagates also on complex concepts constructed by arbitrary combination of concept union and concept intersection constructors (Theorem 56). However, under the original semantics, this simply does not hold for onto-bridge rules. With conjunctive onto-bridge rules, we are finally able to establish this property.

Theorem 60. *Assume a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$. Let for some $n > 0$ the bridge rules $i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1$, $\ldots$, $i : C_n \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_n$ be all part of $\mathfrak{B}$. Let there be two complex concepts, an i-concept E and a j-concept F, constructed as follows:*

1. *E is constructed from the n concepts $C_1, \ldots, C_n$ using any combination of concept intersection and concept union constructors, and no other constructors;*

2. *F is obtained from E by replacing each occurrence of C_i by G_i, for each i, such that $1 \leq i \leq n$ (let us denote this transformation by $(\cdot)^\star$).*

Then in every distributed model $\mathfrak{I}$ of $\mathfrak{T}$ it holds that $r_{ij}\big(E^{\mathcal{I}_i}\big) \supseteq F^{\mathcal{I}_j}$.

Proof. Let $\mathfrak{T}$, the bridge rules $i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1$, $\ldots$, $i : C_n \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_n$, and the concepts E and F be as in the assumptions of the theorem. We first observe, that the atomic bridge rules normal form (ABRNF) is also valid for DDL knowledge bases with conjunctive bridge rules. This is easily verified by inspection of the proof of Theorem 46. Hence, without loss of generality, we will assume that the knowledge base $\mathfrak{T}$ is in ABRNF.

There exists a concept E' in DNF, that is equivalent to E, since there exists one for every concept (Theorem 8). From Definition 9 we know that $E' = E_1 \sqcup \cdots \sqcup E_m$, $m > 0$, such that for each $k \in \{1, \ldots, m\}$, E_k is constructed by concept intersection of one or more concepts from the set $\{C_1, \ldots, C_n\}$. Observe, that the concept $F' = F_1 \sqcup \ldots \sqcup F_m$, obtained by applying the transformation $(\cdot)^\star$ of E' (i.e., $F' = (E')^\star$), is also in DNF and it is equivalent to F.

The proof proceeds by mathematical induction on m. In the base case $m = 1$. In this case we have $E \equiv E_1 = C_{1_1} \sqcap \cdots \sqcap C_{1_{m_1}}$ and $F \equiv F_1 = G_{1_1} \sqcap \cdots \sqcap G_{1_{m_1}}$. Since we have assumed a conjunctive onto-bridge rule between each pair of concepts C_{1_l} and G_{1_l}, $1 \leq l \leq m_1$, it follows from Theorem 58 that ${G_{1_1}}^{\mathcal{I}_j} \cap \cdots \cap {G_{1_{m_1}}}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_{1_1}}^{\mathcal{I}_i} \cap \cdots \cap {C_{1_{m_1}}}^{\mathcal{I}_i}\big)$, which implies $F^{\mathcal{I}_j} \subseteq r_{ij}\big(E^{\mathcal{I}_i}\big)$.

In the induction step $m = k > 1$, and we have $E \equiv E_1 \sqcup \cdots \sqcup E_k$ and $F \equiv F_1 \sqcup \cdots \sqcup F_k$. From the induction hypothesis we have that ${F_1}^{\mathcal{I}_j} \cup \cdots \cup {F_{k-1}}^{\mathcal{I}_j} \subseteq r_{ij}\big({E_1}^{\mathcal{I}_i} \cup \cdots \cup {E_{k-1}}^{\mathcal{I}_i}\big)$. Analogously to the base case, we get from Theorem 58 that ${F_k}^{\mathcal{I}_j} \subseteq r_{ij}\big({E_k}^{\mathcal{I}_i}\big)$. This gives us:

$$ {F_1}^{\mathcal{I}_j} \cup \cdots \cup {F_k}^{\mathcal{I}_j} \subseteq r_{ij}\big({E_1}^{\mathcal{I}_i} \cup \cdots \cup {E_{k-1}}^{\mathcal{I}_i}\big) \cup r_{ij}\big({E_k}^{\mathcal{I}_i}\big) \ , $$

however, since the projection of the union of two sets is always equal to the union of their projections, it follows that:

$$ {F_1}^{\mathcal{I}_j} \cup \cdots \cup {F_k}^{\mathcal{I}_j} \subseteq r_{ij}\big({E_1}^{\mathcal{I}_i} \cup \cdots \cup {E_{k-1}}^{\mathcal{I}_i}\big) \cup r_{ij}\big({E_k}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({E_1}^{\mathcal{I}_i} \cup \cdots \cup {E_k}^{\mathcal{I}_i}\big) \ , $$

which by the semantics of complex concepts is equivalent to $F^{\mathcal{I}_j} \subseteq r_{ij}\big(E^{\mathcal{I}_i}\big)$. □

And so we have seen that conjunctive bridge rules exhibit interesting semantic properties, and they allow for increased propagation of semantic relations towards complex concepts, if bridge rules are expressed between the concepts from which these are constructed. We will learn in Sect. 4.5 that this indeed secures increased subsumption propagation to the level which we have described as desired.

At this point we need to remark that in our previous report (Homola 2007*b*), where conjunctive bridge rules were first introduced, we have employed a different semantics for them. This original semantics is less complex, it does not require ${G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$ for each n-tuple of conjunctive onto-bridge rules $i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1, \ldots, i : C_n \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_n$, instead it only requires ${G_1}^{\mathcal{I}_j} \cap {G_2}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap {C_2}^{\mathcal{I}_i}\big)$ for each pair of such bridge rules $i : C_1 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_1$ and $i : C_2 \stackrel{\sqsupseteq}{\twoheadrightarrow} j : G_2$ that are present in $\mathfrak{B}$.

This semantics, however, is not correct. Neither Theorem 58 and nor Theorem 60 do hold under this simpler semantics. Consider the following example.

Example 13. *Consider a DDL knowledge base $\mathfrak{T}$ with two local TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and three bridge rules in $\mathfrak{B}$:*

$$ 1 : C \stackrel{\sqsupseteq}{\twoheadrightarrow} 2 : F, \qquad 1 : D \stackrel{\sqsupseteq}{\twoheadrightarrow} 2 : G, \qquad 1 : E \stackrel{\sqsupseteq}{\twoheadrightarrow} 2 : H \ . $$

Consider the distributed interpretation $\mathfrak{I}$, in which $C^{\mathcal{I}_1} = \{x_1, x_2\}$, $D^{\mathcal{I}_1} = \{x_2, x_3\}$, $E^{\mathcal{I}_1} = \{x_3, x_1\}$, $F^{\mathcal{I}_2} = G^{\mathcal{I}_2} = H^{\mathcal{I}_2} = \{y\}$, and $r_{ij}(x_1) = r_{ij}(x_2) = r_{ij}(x_3) = \{y\}$.

In $\mathfrak{I}$, we have $F^{\mathcal{I}_2} \cap G^{\mathcal{I}_2} \subseteq r_{ij}\big(C^{\mathcal{I}_1} \cap D^{\mathcal{I}_1}\big)$ and $G^{\mathcal{I}_2} \cap H^{\mathcal{I}_2} \subseteq r_{ij}\big(D^{\mathcal{I}_1} \cap E^{\mathcal{I}_1}\big)$ and $H^{\mathcal{I}_2} \cap F^{\mathcal{I}_2} \subseteq r_{ij}\big(E^{\mathcal{I}_1} \cap C^{\mathcal{I}_1}\big)$. Hence, $\mathfrak{I}$ is a model of $\mathfrak{T}$ according to the weaker semantics (Homola 2007b). However, in $\mathfrak{I}$, $F^{\mathcal{I}_2} \cap G^{\mathcal{I}_2} \cap H^{\mathcal{I}_2} = \{y\} \nsubseteq \emptyset = r_{ij}\big(C^{\mathcal{I}_1} \cap D^{\mathcal{I}_1} \cap E^{\mathcal{I}_1}\big)$, and hence Theorem 58 is falsified under the weaker semantics.

And so if follows, that the stronger semantics of conjunctive bridge rules, as we have introduced it in this thesis, is required in order to assure the desired level of subsumption propagation, albeit computationally it is likely to be harder to handle.

4.4 Transformational Semantics

It turns out, as we will now show, that conjunctive onto-bridge rules are reducible into normal onto-bridge rules. In light of this result, it will be evident that the expressive power of the framework is not increased by introducing conjunctive bridge rules. Furthermore, if we would forbid normal onto-bridge rules and allow only conjunctive ones, the resulting framework would have reduced expressive power, as the counterintuitive scenarios as those of Examples 11 and 12 would be ruled out as established by Theorem 58.

In a sense, conjunctive bridge rules represent a specific way of modeling with normal bridge rules, and can be seen as a form of macros, added to the framework.

Theorem 61. *Let us assume a DDL knowledge base $\mathfrak{T}$ with an index set I and a set of bridge rules $\mathfrak{B}$, that also contains conjunctive bridge rules. Let us construct another DDL knowledge base $\mathfrak{T}'$, which will use the same index set I and its set of bridge rules is $\mathfrak{B}'$, in three steps:*

1. *set $\mathcal{T}_i' = \mathcal{T}_i$, for each $i \in I$;*
2. *add all normal bridge rules found in $\mathfrak{B}$ into $\mathfrak{B}'$;*
3. *add $i : C_1 \sqcap \cdots \sqcap C_n \xrightarrow{\sqsupseteq} j : G_1 \sqcap \cdots \sqcap G_n$ into $\mathfrak{B}'$, for each n-tuple of distinct bridge rules $i : C_1 \overset{\sqsupseteq}{\rightarrow} j : G_1$, ..., $i : C_n \overset{\sqsupseteq}{\rightarrow} j : G_n$ found in $\mathfrak{B}$, for any $i, j \in I$.*

It follows that $\mathfrak{T}$ and $\mathfrak{T}'$ are equivalent, more precisely, any distributed model of $\mathfrak{T}$ is also a distributed model of $\mathfrak{T}'$ and vice versa.

Proof. Let $\mathfrak{I}$ be a distributed model of $\mathfrak{T}$. We shall prove that $\mathfrak{I}$ is a model of $\mathfrak{T}'$. Due to the construction of $\mathfrak{T}'$ all local TBoxes $\mathcal{T}_i'$ are satisfied by $\mathfrak{I}$, for each $i \in I$, and also all bridge rules that belong to $\mathfrak{B}' \setminus \mathfrak{B}$ are satisfied by $\mathfrak{I}$, because these elements of $\mathfrak{T}'$ are also present in $\mathfrak{T}$ of which $\mathfrak{I}$ is a model. It remains to prove that also all bridge rules added during the third step of the transformation are satisfied by $\mathfrak{I}$. Given any such bridge rule $\phi = i : C_1 \sqcap \cdots \sqcap C_n \xrightarrow{\sqsupseteq} j : G_1 \sqcap \cdots \sqcap G_n$, by the construction, it must be the case that the n bridge rules $i : C_1 \overset{\sqsupseteq}{\rightarrow} j : G_1$, ..., $i : C_n \overset{\sqsupseteq}{\rightarrow} j : G_n$ all belong to $\mathfrak{B}$. Since these bridge rules are satisfied by $\mathfrak{I}$, it follows from Definition 70 that ${G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$ and hence also ϕ is satisfied by $\mathfrak{I}'$.

The other way around, let $\mathfrak{I}'$ be a distributed model of $\mathfrak{T}'$, we will show that it is also a distributed model of $\mathfrak{T}$. Indeed each local TBox $\mathcal{T}_i$, and each normal bridge rule is satisfied by $\mathfrak{I}'$ as these elements are also part of $\mathfrak{T}'$. It remains to show that also each conjunctive bridge rule $\phi = i : C \overset{\sqsupseteq}{\rightarrow} j : G \in \mathfrak{B}$ is satisfied by $\mathfrak{I}'$. We must show that given any other n-tuple of bridge rules $i : D_1 \overset{\sqsupseteq}{\rightarrow} j : H_1$, ..., $i : D_n \overset{\sqsupseteq}{\rightarrow} j : H_n$ that are present in $\mathfrak{B}$, it holds that $G^{\mathcal{I}_j} \cap {H_1}^{\mathcal{I}_j} \cap \cdots \cap {H_n}^{\mathcal{I}_j} \subseteq r_{ij}\big(C^{\mathcal{I}_i} \cap {D_1}^{\mathcal{I}_i} \cap \cdots \cap {D_n}^{\mathcal{I}_i}\big)$. In this case, however, it follows from the construction that there is $\psi \in \mathfrak{B}'$, such that $\psi = i : E_1 \sqcap \cdots \sqcap E_{n+1} \xrightarrow{\sqsupseteq} j : F_1 \sqcap \cdots \sqcap F_{n+1}$, and there is $k \in \{1, \ldots, n+1\}$ with $E_k = C$ and $F_k = D$, and also for each $l \in \{1, \ldots, n\}$ there is $k \in \{1, \ldots, n+1\}$ with $E_k = D_l$ and $F_k = H_l$. Since ψ is satisfied by $\mathfrak{I}'$, if follows that $i : C_1 \sqcap \cdots \sqcap C_n \xrightarrow{\sqsupseteq} j : G_1 \sqcap \cdots \sqcap G_n$, and hence ϕ is satisfied by $\mathfrak{I}$. □

This result theoretically yields a decision algorithm for support for DDL with conjunctive bridge rules. Applying the transformation first, and using the decision algorithm of Serafini & Tamilin (2004), implemented by Serafini et al. (2005) afterwards results into sound and complete decision algorithm. Practical usability of this approach is doubtful, since in the worst case the transformation introduces an exponential blowup in the number of bridge rules. This suggests that further investigation of reasoning in presence of conjunctive bridge rules is needed.

On the other hand, as we shall see in the next section, due to the reduction it follows that the desired properties of DDL, such as directionality and local inconsistency properties are satisfied also by DDL with conjunctive bridge rules. This means that conjunctive bridge rules are very sound from the semantic point of view. This is their advantage compared to DDL with injective domain relations, an alternative to conjunctive bridge rules that we will investigate in Sect. 4.6.

4.5 Evaluation and Comparison

In this section, we evaluate DDL with conjunctive bridge rules with respect to the desiderata that have been postulated for DDL in the literature, and which we have discussed in Sect. 3.2. In light of Theorem 61, it follows rather straightforward that DDL with conjunctive bridge rules do not differ from the original DDL in terms of properties that hold for the original DDL. The main difference is that stronger subsumption propagation patterns occur in situations when conjunctive onto-bridge rules appear on the scene. We provide a characterization of these patterns in the end of this section.

Let us start with the monotonicity property, which states, that all the local entailments which follow from each ontology are not canceled by the mappings. Given the fact that monotonicity holds for classic DDL, it is established for DDL with conjunctive bridge rules as direct consequence of Theorem 61. We also include a direct proof below.

Theorem 62. *The monotonicity property (Property 1) is satisfied by DDL even if conjunctive bridge rules are allowed.*

Proof. We shall prove that the following holds in every distributed TBox $\mathfrak{T}$:

$$\mathcal{T}_i \models C \sqsubseteq D \implies \mathfrak{T} \models_\epsilon i : C \sqsubseteq D \ .$$

Observe, that in any distributed model $\mathfrak{I}$ of $\mathfrak{T}$, $\mathcal{I}_i$ is either a model of $\mathcal{T}_i$ or a hole. Hence, if $\mathcal{T}_i \models C \sqsubseteq D$, it must be that also $\mathcal{I}_i \models_\epsilon i : C \sqsubseteq D$. Therefore also $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$. □

The directionality property is particularly important. It states, that knowledge reuse respects the direction of bridge rules in the knowledge base. In other words, that there is no backflow of information. This property is satisfied in the original semantics. With use of Theorem 61 we easily establish it also in presence of conjunctive bridge rules.

Theorem 63. *The directionality property (Property 2) is satisfied in DDL even if conjunctive bridge rules are allowed.*

Proof. Given a distributed TBox $\mathfrak{T}$ with some index set I such that is also contains conjunctive bridge rules, we shall prove that if there is no directed path between i to j in $\mathrm{G}_{\mathfrak{T}}$, then

$$\mathfrak{T} \models_\epsilon j : C \sqsubseteq D \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D \ .$$

Recall that $\mathfrak{T} \setminus\setminus \{i\}$ is obtained by removing $\mathcal{T}_i$, $\mathfrak{B}_{ik}$ and $\mathfrak{B}_{li}$ from $\mathfrak{T}$ for each $k, l \in I$.

Let us denote by $\mathfrak{T}'$ and $(\mathfrak{T} \setminus\setminus \{i\})'$ the DDL knowledge bases without conjunctive bridge rules obtained from $\mathfrak{T}$ and $\mathfrak{T} \setminus\setminus \{i\}$ (in the respective order) by application of the transformation described in Theorem 61. Observe, that $\mathfrak{T}' \setminus\setminus \{i\} = (\mathfrak{T} \setminus\setminus \{i\})'$. This is due to the fact that for any $i, j \in I$, $\mathfrak{B}'_{ij}$ is computed exclusively based on $\mathfrak{B}_{ij}$ as the input. Finally, for any j-local subsumption formula ϕ we have:

$$\mathfrak{T} \models_\epsilon j : \phi \iff \mathfrak{T}' \models_\epsilon j : \phi \iff \mathfrak{T}' \setminus\setminus \{i\} \models_\epsilon j : \phi \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : \phi \ .$$

The equivalence on the left and as well the one on the right both follow from Theorem 61. The equivalence in the middle is due to the fact that directionality is satisfied by original DDL without conjunctive bridge rules. □

Also the local inconsistency property is very important. This property characterizes the behaviour of the original DDL semantics, for which, we have learned in Sect. 3.2 it is satisfied, in cases when inconsistency accidentally occurs in one of the ontologies between which mapping is expressed in DDL.

Recall, that this characterization makes use of the $\mathfrak{T}(\epsilon_J)$ operator. For classic DDL it is defined as follows. Given two parameters, a distributed TBox $\mathfrak{T}$ over I, and $J \subseteq I$, we have $\mathfrak{T}(\epsilon_J) = \mathfrak{T} \setminus\setminus J \cup \{D \sqsubseteq \bot \mid j : C \xrightarrow{\sqsupseteq} i : D \in \mathfrak{B} \wedge j \in J\}$. However, in order to obtain an analogous characterization we need to take also conjunctive onto-bridge rules into account. Hence, in presence of conjunctive bridge rules, we define $\mathfrak{T}(\epsilon_J) = \mathfrak{T} \setminus\setminus J \cup \{D \sqsubseteq \bot \mid j : C \xrightarrow{\sqsupseteq} i : D \in \mathfrak{B} \wedge j \in J\} \cup \{D \sqsubseteq \bot \mid j : C \xrightarrow{\sqsupseteq}_{\twoheadrightarrow} i : D \in \mathfrak{B} \wedge j \in J\}$.

Theorem 64. *The local inconsistency property (Property 4) is satisfied in DDL even if conjunctive bridge rules are allowed.*

Proof. Given a DDL knowledge base $\mathfrak{T}$ that also possibly contains conjunctive onto-bridge rules, we shall prove that the following holds: $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ if and only if for any $J \subseteq I$, not containing i, it holds that $\mathfrak{T}(\epsilon_J) \models_\epsilon i : C \sqsubseteq D$.

Let us again denote by $(\cdot)'$ the transformation described in Theorem 61. As we know, it takes a DDL knowledge base on the input and transforms it into one without conjunctive bridge rules.

We will now show, that even if they do differ, $(\mathfrak{T}(\epsilon_J))'$ and $\mathfrak{T}'(\epsilon_J)$ are in fact equivalent, given any $\mathfrak{T}$ and any subset J of its index set. From the construction we have $(\mathfrak{T}(\epsilon_J))' \subseteq \mathfrak{T}'(\epsilon_J)$. Hence, it suffices to show that each formula $i : \phi \in \mathfrak{T}'(\epsilon_J) \setminus (\mathfrak{T}(\epsilon_J))'$ is satisfied in each model of $(\mathfrak{T}(\epsilon_J))'$ (this is due to monotonicity of this logical framework). But, from the construction, any such formula $i : \phi$ is of the form $i : \phi = i : D_1 \sqcap \cdots \sqcap D_n \sqsubseteq \bot$ such that $i \notin J$, $j \in J$ and there are bridge rules $j : C_k \xrightarrow{\sqsupseteq}_{\twoheadrightarrow} j : D_k \in \mathfrak{B}$, for $1 \leq k \leq n$. The existence of these bridge rules implies that $i : \phi_k = D_k \sqsubseteq \bot$ belongs to $(\mathfrak{T}(\epsilon_J))'$, for $1 \leq k \leq n$, and thus $i : \phi$ is entailed by each model of $(\mathfrak{T}(\epsilon_J))'$.

Now we will establish the result:

1. $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ if and only if $\mathfrak{T}' \models_\epsilon i : C \sqsubseteq D$ (due to Theorem 61);

2. the latter is if and only if $\mathfrak{T}'(\epsilon_J) \models_\epsilon i : C \sqsubseteq D$ for any $J \subseteq I$, $i \notin J$ (due to the fact that the property holds in classic DDL);

3. this is if and only if $(\mathfrak{T}(\epsilon_J))' \models_\epsilon i : C \sqsubseteq D$ for any $J \subseteq I$, $i \notin J$ (this we have just proven);

4. and this is if and only if $\mathfrak{T}(\epsilon_J) \models_\epsilon i : C \sqsubseteq D$ for any $J \subseteq I$, $i \notin J$ (due to Theorem 61).

□

And so we see, that even in presence of conjunctive bridge rules, the desired properties of DDL remain unaltered, that is, monotonicity, directionality and local inconsistency are all satisfied.

Finally, our attention is shifted towards subsumption propagation. As we have learned in Sect. 4.3, the semantics of conjunctive onto-bridge rules enables propagation of semantic relations, implied by bridge rules, on more complex concepts constructed from concepts which are directly connected by these bridge rules (Theorem 60). Similar property holds for normal into-bridge rules (Theorem 56). By combining these two results, we are able to derive various subsumption propagation patterns, similar to the simple subsumption propagation property (Property 5) and the general subsumption propagation property (Property 6), that are already known from the literature (Borgida & Serafini 2003, Serafini & Tamilin 2004, Serafini et al. 2005). The most general pattern that we are able to establish is expressed by the following theorem.

Theorem 65. *Assume a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$. Let for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1$, …, $i : C_n \xrightarrow{\sqsupseteq} j : G_n$ be all part of $\mathfrak{B}$. Let for some $m > 0$ the bridge rules $i : D_1 \xrightarrow{\sqsubseteq} j : H_1$, …, $i : D_n \xrightarrow{\sqsubseteq} j : H_n$ be all part of $\mathfrak{B}$. Let there be four complex concepts, i-concepts E, E', and j-concepts F, F', constructed as follows:*

1. *E is constructed from the n concepts $C_1, \ldots, C_n$ using any combination of concept intersection and concept union constructors, and no other constructors;*

2. *E' is constructed from the m concepts $D_1, \ldots, D_m$ using any combination of concept intersection and concept union constructors, and no other constructors;*

3. *F is obtained from E by replacing each occurrence of C_i by G_i, for each i, such that $1 \leq i \leq n$;*

4. *F' is obtained from E' by replacing each occurrence of D_i by H_i, for each i, such that $1 \leq i \leq m$.*

Then the following always holds:

$$\mathfrak{T} \models_\epsilon i : E \sqsubseteq E' \implies \mathfrak{T} \models_\epsilon i : F \sqsubseteq F' \ .$$

Proof. Given the assumptions of the theorem, and given any distributed model $\mathfrak{I}$ of $\mathfrak{T}$, since $\mathfrak{T} \models_\epsilon i : E \sqsubseteq E'$, we have $E^{\mathcal{I}_i} \subseteq E'^{\mathcal{I}_i}$. As $r_{ij}(\cdot)$ is a mapping, it holds that $r_{ij}\left(E^{\mathcal{I}_i}\right) \subseteq r_{ij}\left(E'^{\mathcal{I}_i}\right)$. From Theorem 60, we get that $F^{\mathcal{I}_j} \subseteq r_{ij}\left(E^{\mathcal{I}_i}\right) \subseteq r_{ij}\left(E'^{\mathcal{I}_i}\right)$. From Theorem 56, we get that $F^{\mathcal{I}_j} \subseteq r_{ij}\left(E^{\mathcal{I}_i}\right) \subseteq r_{ij}\left(E'^{\mathcal{I}_i}\right) \subseteq F'^{\mathcal{I}_j}$, which we wanted to prove. □

If conjunctive bridge rules are not allowed, that is, if we fall back to original DDL framework, the most general subsumption pattern that we derive is the following one.

Theorem 66. *Assume a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$. Let for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsupseteq} j : G_n$ be all part of $\mathfrak{B}$. Let for some $m > 0$ the bridge rules $i : D_1 \xrightarrow{\sqsubseteq} j : H_1$, ..., $i : D_n \xrightarrow{\sqsubseteq} j : H_n$ be all part of $\mathfrak{B}$. Let there be four complex concepts, i-concepts E, E', and j-concepts F, F', constructed as follows:*

1. *$E = C_1 \sqcup \cdots \sqcup C_n$;*
2. *E' is constructed from the m concepts $D_1, \ldots, D_m$ using any combination of concept intersection and concept union constructors, and no other constructors;*
3. *$F = G_1 \sqcup \cdots \sqcup C_n$;*
4. *F' is obtained from E' by replacing each occurrence of D_i by H_i, for each i, such that $1 \leq i \leq m$.*

Then the following always holds:

$$\mathfrak{T} \models_\epsilon i : E \sqsubseteq E' \implies \mathfrak{T} \models_\epsilon i : F \sqsubseteq F' \ .$$

Proof. The proof is analogous to the proof of Theorem 65, but instead of Theorem 60 it uses Theorem 54. □

These propagation patterns basically correspond to our previous findings regarding the propagation of semantic relations between concepts implied by bridge rules on more complex concepts. More specifically, we see, that for the original DDL, concept union and concept intersection constructors are "supported" – in this sense – by into-bridge rules; however only concept union is supported by onto-bridge rules. This problem is eliminated by use of conjunctive onto-bridge rules, which support both concept union and concept intersection.

The subsumption propagation properties, as established in the literature (Borgida & Serafini 2003, Serafini & Tamilin 2004, Serafini et al. 2005), are now simple consequences of these more general subsumption propagation patterns. Moreover, as a very trivial consequence of Theorem 57, they hold also when conjunctive onto-bridge rules are involved. Just for sake of completeness we formulate them as corollaries.

Corollary 67. *The simple subsumption propagation property (Property 5) is satisfied in DDL even if conjunctive bridge rules are allowed. That is, for each*

distributed TBox $\mathfrak{T}$ with an index set I the following holds: if $i : C \overset{\exists}{\rightsquigarrow} j : G \in \mathfrak{B}$ and $i : D \xrightarrow{\sqsubseteq} j : H \in \mathfrak{B}$ then

$$\mathfrak{T} \models_\epsilon i : C \sqsubseteq D \implies \mathfrak{T} \models_\epsilon j : G \sqsubseteq H \ .$$

Corollary 68. *The general subsumption propagation property (Property 6) is satisfied in DDL even if conjunctive bridge rules are allowed. That is, for each distributed TBox $\mathfrak{T}$ with an index set I, the following holds: if $i : C \overset{\exists}{\rightsquigarrow} j : G \in \mathfrak{B}$ and $i : D_k \xrightarrow{\sqsubseteq} j : H_k \in \mathfrak{B}$, for $1 \leq k \leq n$ then*

$$\mathfrak{T} \models_\epsilon i : C \sqsubseteq \bigsqcup_{k=1}^{n} D_k \implies \mathfrak{T} \models_\epsilon j : G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \ .$$

And so, we conclude our investigation of DDL enriched with conjunctive bridge rules and their semantic properties. We have learned, that interaction of bridge rules with the concept intersection constructor is now increased, yielding very general subsumption propagation patterns. Moreover, semantically this enrichment of the framework is reasonable in that sense that it does not break any of the desired properties.

4.6 DDL with Injective Domain Relations

So far in this chapter, we have addressed the problem of unintuitive handling of the concept intersection constructor by introducing a new type of bridge rules, with specifically tailored semantics. This semantics does generate additional semantic constraints when needed, that is, when a possibly problematic combination of bridge rules is detected. Although the semantic properties of this approach are encouraging, we have seen, that the number of additionally generated conditions is possibly very large.

In this section, we propose a different approach to this problem. We will take a more global perspective – instead of generating a large number of local constraints, we will place some global restrictions on the admissible domain relation that is part of each distributed interpretation. It is remarkable, that this idea will be also applied later on in this work, in Chap. 5, in order to improve the extent of subsumption propagation between remote ontologies within a DDL knowledge base. We conjecture, that such a global approach may possibly result into a semantics with better computational properties.

As much as the previous one, also this approach is rooted in Example 11. Recall that, in this example we deal with two concepts Bird_1 and $\mathsf{NonFlying}_1$, which are from the same local ontology $\mathcal{T}_1$, and are disjoint. These two concepts are then both mapped to a 2-concept $\mathsf{Penguin}_2$ by onto-bridge rules. We have learned from Example 11, that all these axioms together are not enough to imply that $\mathsf{Penguin}_2$ is unsatisfiable. This is possible because the classic semantics of DDL (Borgida & Serafini 2003, Serafini & Tamilin 2004) does not impose any restrictions on the domain relation. And so, in some distributed model, the intersection of the two r-images $r_{12}\left(\mathsf{Bird}_1^{\mathcal{I}_1}\right)$ and $r_{12}\left(\mathsf{NonFlying}_1^{\mathcal{I}_1}\right)$ is possibly nonempty. We have seen a depiction of one such a model in Fig. 4.1.

It immediately follows, that if only domain relations that are injective were allowed, the problem would not occur. DDL with injective domain relations

have been already discussed by Serafini et al. (2005), however, instead of a global restriction, a more fine grained local application of injectivity tied to particular selected concepts has been suggested. This is because, the idea of injective domain relation is conflicting with some of the basic intuitions behind DDL (see the discussion below). In this work we take a different perspective, and we will investigate the derivate of DDL semantics that imposes injectivity globally. We will be interested in the semantic properties of this semantics, since we hope that it may be computationally more feasible as local restrictions such as for instance those imposed by conjunctive bridge rules.

We have commonly treated the relation r_{ij} as a mapping $r_{ij} : \Delta^{\mathcal{I}_i} \rightarrow 2^{\Delta^{\mathcal{I}_j}}$. That is, its values are sets, subsets of $\Delta^{\mathcal{I}_j}$. We define injectivity for this types of mappings as follows.

Definition 71 (Injective mapping)**.** *Given two sets S and T, let $m : S \mapsto 2^T$ be a mapping. The mapping m is said to be injective,*[2] *if there is no element $t \in T$ such that there are two elements $s_1, s_2 \in S$ with $s_1 \neq s_2$, $t \in m(s_1)$ and $t \in m(s_2)$.*

The new semantics is then obtained simply by requiring injectivity of the domain relation in any given model.

Definition 72 (Distributed model under injectivity)**.** *Given a DDL knowledge base $\mathfrak{T}$ over an index set I. A distributed interpretation $\mathfrak{I}$ is a model of $\mathfrak{T}$ under the DDL semantics with injectivity, if it is a distributed model of $\mathfrak{T}$ in the sense of Definition 59 and if its domain relation is injective.*

In the rest of this section we always assume distributed models under the DDL semantics of injectivity. As usual, we will also call them ϵ-models. If an ϵ-model is globally consistent it is a d-model.

We will now notice, that this novel semantics is strictly stronger that the original one, that is, if a formula is entailed by the original semantics, it is also entailed by the one with injectivity.

Theorem 69. *Given a distributed TBox $\mathfrak{T}$ and a subsumption formula $i : \phi$, if $\mathfrak{T} \models_\epsilon i : \phi$ according to the original semantics of DDL, then $\mathfrak{T} \models_\epsilon i : \phi$ also holds in DDL with injective domain relations.*

Proof. This follows from the fact that each distributed model with injective domain relation is also a model with respect to the original DDL semantics. If a formula $i : \phi$ is satisfied in each model from the set of models of $\mathfrak{T}$ according to the original semantics, it is also satisfied by each model from its subset – the set of models of $\mathfrak{T}$ with injective domain relations. □

We now shift our attention towards the semantic properties of this novel semantics. Our main interest is in the interaction of onto-bridge rules and the concept intersection constructor. As it turns out, we are to able prove a variant of Theorem 58, only in this case we use normal and not conjunctive onto-bridge rules.

Theorem 70. *Given a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$, if for some $n > 0$ the bridge*

[2] In previous work we have referred to this notion also as strict injectivity (Homola 2007*a*).

rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1, \ldots, i : C_n \xrightarrow{\sqsupseteq} j : G_n$ *are all part of* $\mathfrak{B}$*, then in every distributed model* $\mathfrak{I}$ *such that its domain relation* r *is injective, we have*

$$r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big) \supseteq (G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \quad .$$

Proof. We shall prove $(G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \subseteq r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big)$ which by definition is equivalent to ${G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$.

In the knowledge base, we have n bridge rules, each of the form. $i : C_k \xrightarrow{\sqsupseteq} j : G_k$, for $1 \leq k \leq n$. This implies ${G_k}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_k}^{\mathcal{I}_i}\big)$, for $1 \leq k \leq n$. Hence ${G_1}^{\mathcal{I}_j} \cap \cdots \cap {G_n}^{\mathcal{I}_j} \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_n}^{\mathcal{I}_i}\big)$, and it now remains to prove $r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_n}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$.

By mathematical induction on n. In the base case $n = 1$, and $r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i}\big)$ holds trivially.

In the induction step we have $n = m > 1$, and we ought to prove $r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_m}^{\mathcal{I}_i}\big)$. From the induction hypothesis we obtain:

$$r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}\big) \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big) \quad ,$$

and hence it remains to prove:

$$r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}\big) \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_m}^{\mathcal{I}_i}\big) \quad .$$

This proposition is proven by contradiction. Assume the contrary, that is, $r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}\big) \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big) \not\subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_m}^{\mathcal{I}_i}\big)$. This implies that there is $x \in r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}\big) \cap r_{ij}\big({C_m}^{\mathcal{I}_i}\big)$ such that $x \notin r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_m}^{\mathcal{I}_i}\big)$. This implies that x has a preimage in both ${C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}$ and ${C_m}^{\mathcal{I}_i}$, but not in their intersection. But then these must be two distinct preimages $y_1 \in {C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_{m-1}}^{\mathcal{I}_i}$ and $y_2 \in {C_m}^{\mathcal{I}_i}$, $y_1 \neq y_2$. This observation violates injectivity, however, which in turn closes the inductive step.

Hence $r_{ij}\big({C_1}^{\mathcal{I}_i}\big) \cap \cdots \cap r_{ij}\big({C_n}^{\mathcal{I}_i}\big) \subseteq r_{ij}\big({C_1}^{\mathcal{I}_i} \cap \cdots \cap {C_n}^{\mathcal{I}_i}\big)$ and therefore we have $(G_1 \sqcap \cdots \sqcap G_n)^{\mathcal{I}_j} \subseteq r_{ij}\Big((C_1 \sqcap \cdots \sqcap C_n)^{\mathcal{I}_i}\Big)$.

□

Indeed, if we reinspect Examples 11 and 12, we see that the problems described therein do not occur any more if only distributed interpretations with injective domain relations are allowed. The newly derived semantics is stronger in a sense: some concepts may become unsatisfiable, as $\mathsf{Penguin}_2$ in Example 11 and some additional subsumption formulae are entailed as the one discussed in Example 12. In fact, it follows from the theorem which we have just proven, that the semantics with injectivity is strictly stronger than the one with conjunctive bridge rules.

Corollary 71. *Given a distributed TBox* $\mathfrak{T}$ *with a set of bridge rules* $\mathfrak{B}$ *and an index set* I*, that only uses conjunctive onto-bridge rules, let* $\mathfrak{T}'$ *be obtained from* $\mathfrak{T}$ *by replacing each conjunctive onto-bridge rule* $i : C \xrightarrow{\sqsupseteq}_{\twoheadrightarrow} j : G$ *by a normal onto-bridge rule* $i : C \xrightarrow{\sqsupseteq} j : G$*. Let* $i : \phi$ *be a subsumption formula. If* $\mathfrak{T} \models_\epsilon i : \phi$ *according to the semantics with conjunctive bridge rules, then* $\mathfrak{T}' \models_\epsilon i : \phi$ *also holds according to the semantics with injective domain relations.*

Proof. We will show that each distributed model of $\mathfrak{T}'$ with respect to the semantics with injective domain relations is also a distributed model of $\mathfrak{T}$ with respect to the semantics with conjunctive bridge rules. Given a distributed model $\mathfrak{I}'$ of $\mathfrak{T}'$ with respect to semantics with injectivity, we must show, that each conjunctive bridge rule of $\mathfrak{T}$ is satisfied by $\mathfrak{I}'$. This is only if for each n-tuple of other conjunctive onto-bridge rules from $\mathfrak{T}$, the clause number 5 of Definition 70 is satisfied. However, this is an immediate consequence of the construction of $\mathfrak{T}'$ and of Theorem 70 □

In a sense, conjunctive bridge rules assert some kind of local injectivity of the domain relation only for those elements of local domains for which it is necessarily required.

As a consequence, it follows that the DDL semantics with injectivity propagates at least as much semantic relations on complex concepts constructed by combination of the concept union and intersection constructor, as the one with conjunctive bridge rules, and the most general subsumption propagation pattern applies.

Corollary 72. *Assume a distributed TBox $\mathfrak{T}$ with a set of bridge rules $\mathfrak{B}$ and some local TBoxes $\mathcal{T}_i$ and $\mathcal{T}_j$ such that $i \neq j$. Let for some $n > 0$ the bridge rules $i : C_1 \xrightarrow{\sqsupseteq} j : G_1$, ..., $i : C_n \xrightarrow{\sqsupseteq} j : G_n$ be all part of $\mathfrak{B}$. Let for some $m > 0$ the bridge rules $i : D_1 \xrightarrow{\sqsubseteq} j : H_1$, ..., $i : D_n \xrightarrow{\sqsubseteq} j : H_n$ be all part of $\mathfrak{B}$. Let there be four complex concepts, i-concepts E, E', and j-concepts F, F', constructed as follows:*

1. *E is constructed from the n concepts $C_1, \ldots, C_n$ using any combination of concept intersection and concept union constructors, and no other constructors;*

2. *E' is constructed from the m concepts $D_1, \ldots, D_m$ using any combination of concept intersection and concept union constructors, and no other constructors;*

3. *F is obtained from E by replacing each occurrence of C_i by G_i, for each i, such that $1 \leq i \leq n$;*

4. *F' is obtained from E' by replacing each occurrence of D_i by H_i, for each i, such that $1 \leq i \leq m$.*

Under the DDL semantics with injective domain relations, the following always holds:

$$\mathfrak{T} \models_\epsilon i : E \sqsubseteq E' \implies \mathfrak{T} \models_\epsilon i : F \sqsubseteq F' \ .$$

Once again, as this subsumption propagation pattern is very general, both subsumption propagation properties trivially follow.

Corollary 73. *The simple subsumption propagation property (Property 5) is satisfied in DDL under the semantics with injective domain relations.*

Corollary 74. *The general subsumption propagation property (Property 6) is satisfied in DDL under the semantics with injective domain relations.*

And so we have just learned that the newly introduced DDL semantics with injective domain relations is a viable alternative of the DDL semantics with conjunctive bridge rules. At least, with regards to subsumption propagation, it indeed is. However, as we know, there are other important desiderata for DDL. Let us now have a look at these desiderata and evaluate the newly introduced semantics with respect to them.

Theorem 75. *The monotonicity property (Property 1) is satisfied by DDL under the semantics with injective domain relations.*

Proof. We shall prove that the following holds in every distributed TBox $\mathfrak{T}$:

$$\mathcal{T}_i \models C \sqsubseteq D \implies \mathfrak{T} \models_\epsilon i : C \sqsubseteq D \ .$$

Observe, that in any distributed model $\mathfrak{I}$ of $\mathfrak{T}$, $\mathcal{I}_i$ is either a model of $\mathcal{T}_i$ or a hole. Hence, if $\mathcal{T}_i \models C \sqsubseteq D$, it must be that also $\mathcal{I}_i \models_\epsilon i : C \sqsubseteq D$. Therefore also $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$. □

Theorem 76. *The directionality property (Property 2) is satisfied in DDL under the semantics with injective domain relations.*

Proof. Given a distributed TBox $\mathfrak{T}$ with some index set I, we shall prove that if there is no directed path from i to j in $\mathrm{G}_\mathfrak{T}$, then

$$\mathfrak{T} \models_\epsilon j : C \sqsubseteq D \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D \ .$$

Recall that $\mathfrak{T} \setminus\setminus \{i\}$ is obtained by removing $\mathcal{T}_i$, $\mathfrak{B}_{ik}$ and $\mathfrak{B}_{li}$ from $\mathfrak{T}$ for each $k, l \in I$. We will denote $\mathfrak{T} \setminus\setminus \{i\}$ by $\mathfrak{T}'$.

The if part. Given any distributed model $\mathfrak{I}$ of $\mathfrak{T}$, by pruning it off $\mathcal{I}_i$, and r_{ik} and r_{li}, for any $k, l \in I$, we obtain a distributed model $\mathfrak{I}'$ of $\mathfrak{T}'$, whose domain relation is injective. We have assumed that $\mathfrak{T}' \models_\epsilon j : C \sqsubseteq D$, hence $\mathfrak{I}' \models_\epsilon j : C \sqsubseteq D$. However, $\mathcal{I}_j = \mathcal{I}'_j$ and therefore $\mathfrak{I} \models_\epsilon j : C \sqsubseteq D$. Since this holds for any model $\mathfrak{I}$, then also $\mathfrak{T} \models_\epsilon j : C \sqsubseteq D$.

The only-if part. Let $\mathfrak{I}'$ be any distributed model of $\mathfrak{T}'$. Let $J \subseteq I$ be a set that includes i and also any $k \in I$ that is reachable from i by a directed path in $\mathrm{G}_\mathfrak{T}$. Note that $j \notin J$.

We construct a distributed interpretation $\mathfrak{I}$ with the index set I, starting from $\mathfrak{I}'$, as follows: set $\mathcal{I}_k = \mathcal{I}^\epsilon$, for all $k \in J$; set $r_{kl} = \emptyset$ and set $r_{li} = \emptyset$, if $k \in J$ or $l \in J$. We will now prove that $\mathfrak{I}$ is a model of $\mathfrak{T}$:

- $\mathfrak{I} \models_\epsilon \mathcal{T}_k$, for each $k \notin J$, since $\mathcal{I}_k = \mathcal{I}'_k \models \mathcal{T}_k$;
- $\mathfrak{I} \models_\epsilon \mathcal{T}_k$, for each $k \in J$, because $\mathcal{I}_k = \mathcal{I}^\epsilon$ always satisfies any $\mathcal{T}_k$ disregarding its content;
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k, l \in I$, $k \notin J$, $l \notin J$, since $\mathcal{I}_k = \mathcal{I}'_k$, $\mathcal{I}_l = \mathcal{I}'_l$, and $r_{kl} = r'_{kl}$;
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k, l \in I$, $k \in J$, $l \notin J$, since then by construction $\mathfrak{B}_{kl} = \emptyset$, as otherwise $l \in J$.
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k \in I$, $l \in J$, since: for any into-bridge rule $k : G \xrightarrow{\sqsubseteq} l : H \in \mathfrak{B}_{kl}$, we have $r_{kl}\big(G^{\mathcal{I}_k}\big) = H^{\mathcal{I}_l} = \emptyset$ and hence $r_{kl}\big(G^{\mathcal{I}_k}\big) \subseteq H^{\mathcal{I}_l}$; analogously, for any onto-bridge rule $k : G \xrightarrow{\sqsupseteq} l : H \in \mathfrak{B}_{kl}$, we have $r_{kl}\big(G^{\mathcal{I}_k}\big) = H^{\mathcal{I}_l} = \emptyset$ and hence $H^{\mathcal{I}_l} \subseteq r_{kl}\big(G^{\mathcal{I}_k}\big)$;

- finally, r is injective, since $r \subseteq r'$, and r' is injective.

We have assumed that $\mathfrak{T} \models_\epsilon j : C \sqsubseteq D$, hence $\mathfrak{I} \models_\epsilon j : C \sqsubseteq D$. However, $\mathcal{I}'_j = \mathcal{I}_j$ and therefore $\mathfrak{I}' \models_\epsilon j : C \sqsubseteq D$. Since this holds for any model $\mathfrak{I}$, then also $\mathfrak{T}' \models_\epsilon j : C \sqsubseteq D$. □

Theorem 77. *The local inconsistency property (Property 4) is satisfied in DDL under the semantics with injective domain relations.*

Proof. Given a DDL knowledge base $\mathfrak{T}$, we shall prove that $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ if and only if for any $J \subseteq I$, not containing i, it holds that $\mathfrak{T}(\epsilon_J) \models_\mathrm{d} i : C \sqsubseteq D$. Recall that $\mathfrak{T}(\epsilon_J) = \mathfrak{T} \setminus\setminus J \cup \{D \sqsubseteq \bot \mid j : C \xrightarrow{\sqsupseteq} i : D \in \mathfrak{B} \wedge j \in J\}$.

The only-if part. Assuming that $\mathfrak{T}(\epsilon_J) \models_\mathrm{d} i : C \sqsubseteq D$, for any $J \subseteq I$, $i \notin J$, we ought to prove that $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$. Let $\mathfrak{I}$ be arbitrary model of $\mathfrak{T}$. Let $J = \{i \in I \mid \mathcal{I}_i = \mathcal{I}^\epsilon\}$. If $i \in J$, then trivially $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$. Let us assume $i \notin J$. We construct a distributed interpretation $\mathfrak{I}'$ by removing from $\mathfrak{I}$ all $\mathcal{I}_k$, for each $k \in J$, and also all $\mathfrak{B}_{kl}$, for any $k, l \in I$ such that $k \in J$ or $l \in J$. Now, $\mathfrak{I}'$ is a distributed interpretation over the index set $I \setminus J$, and it is straightforward to see that it is a d-model of $\mathfrak{T}(\epsilon_J)$:

- $\mathfrak{I}'$ is a d-interpretation, since it does not contain holes;
- $\mathfrak{I}' \models_\mathrm{d} \mathcal{T}_k$, for each $k \in I \setminus J$, because $\mathcal{I}'_k = \mathcal{I}_k \models \mathcal{T}_k$;
- $\mathfrak{I}' \models_\mathrm{d} \mathfrak{B}_{kl}$, for each $k, l \in I \setminus J$, as $\mathcal{I}'_k = \mathcal{I}_k$, $\mathcal{I}'_l = \mathcal{I}_l$ and $r'_{kl} = r_{kl}$;
- finally, r' is injective, since $r' \subseteq r$, and r is injective.

We have assumed that $\mathfrak{T}(\epsilon_J) \models_\mathrm{d} i : C \sqsubseteq D$, which means that $\mathfrak{I}' \models_\mathrm{d} i : C \sqsubseteq D$. This in turn implies $\mathfrak{I} \models_\mathrm{d} i : C \sqsubseteq D$, because $\mathcal{I}_i = \mathcal{I}'_i$. Since this holds in any distributed model of $\mathfrak{T}$, we have that $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$, which concludes the only-if part.

The if part. Assuming $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ we ought to prove, that for every $J \subseteq I$, such that $i \notin J$, $\mathfrak{T}(\epsilon_J) \models_\mathrm{d} i : C \sqsubseteq D$. Let J be any subset of I with $i \notin J$. Let $\mathfrak{I}'$ be any d-model of $\mathfrak{T}(\epsilon_J)$. Let us build a new distributed interpretation $\mathfrak{I}$. $\mathfrak{I}$ will use I as its index set. For $k, l \in I \setminus J$, set $\mathcal{I}_k := \mathcal{I}'_k$ and $r_{kl} = r'_{kl}$. For $k \in J$, set $\mathcal{I}_k = \mathcal{I}^\epsilon$. For $k, l \in I$, such that $k \in J$ or $l \in J$, set $r_{kl} = \emptyset$. We will now show that $\mathfrak{I} \models_\epsilon \mathfrak{T}$:

- $\mathfrak{I} \models_\epsilon \mathcal{T}_k$, for each $k \in I \setminus J$, since $\mathcal{I}_k = \mathcal{I}'_k \models \mathcal{T}_k$, as we have assumed;
- $\mathfrak{I} \models_\epsilon \mathcal{T}_k$, for each $k \in J$, since $\mathcal{I}_k = \mathcal{I}^\epsilon \models \mathcal{T}_k$ holds for any $\mathcal{T}_k$;
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k, l \in I \setminus J$, since $\mathfrak{I}' \models_\mathrm{d} \mathfrak{B}_{kl}$ and we have $\mathcal{I}_k = \mathcal{I}'_k$, $\mathcal{I}_l = \mathcal{I}'_l$ and $r_{kl} = r'_{kl}$;
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k \in I$, $l \in J$, since in this case $\Delta^{\mathcal{I}_l} = r_{kl} = \emptyset$ and therefore $\mathfrak{I} \models_\epsilon \phi$ for any bridge rule $\phi \in \mathfrak{B}_{kl}$;
- $\mathfrak{I} \models_\epsilon \mathfrak{B}_{kl}$, for any $k \in J$, $l \in I \setminus J$, because: for any into-bridge rule $k : G \xrightarrow{\sqsubseteq} l : H \in \mathfrak{B}_{kl}$ we have $r_{kl}\big(G^{\mathcal{I}_k}\big) = \emptyset \subseteq H^{\mathcal{I}_l}$; and for each onto-bridge rule $k : G \xrightarrow{\sqsupseteq} l : H \in \mathfrak{B}_{kl}$, from the construction of $\mathfrak{T}(\epsilon_J)$ we have $H \sqsubseteq \bot \in \mathcal{T}_l$, and hence $H^{\mathcal{I}_l} = \emptyset \subseteq r_{kl}\big(G^{\mathcal{I}_k}\big) = \emptyset$;
- finally, r is injective, since in fact $r = r'$, and r' is injective.

As we have assumed, $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$. Then also $\mathfrak{I} \models_\epsilon i : C \sqsubseteq D$. Since $\mathcal{I}'_i = \mathcal{I}_i$, then also $\mathfrak{I}' \models_\mathrm{d} i : C \sqsubseteq D$ (recall that $\mathfrak{I}'$ is a d-model of $\mathfrak{T}(\epsilon_J)$). Since this holds in any d-model of $\mathfrak{T}(\epsilon_J)$ for any $J \subseteq I$, $i \notin J$, the if part is now closed. □

Thus, it turns out, that the newly introduced semantics of DDL in which only injective domain relations are allowed, exhibits interesting semantic behaviour. Not only it satisfies all the desired properties as postulated by Borgida & Serafini (2003) and Serafini & Tamilin (2004), but in addition it provides a sufficient amount of subsumption propagation on complex concepts.

It has to be noted, however, that restricting the domain relation to injective only, is not particularly well aligned with the original intuitions that are behind DDL. DDL have been introduced also with the motivation of overcoming possible modeling heterogeneity between ontologies that are to be integrated. In such cases, it is natural to associate one individual with multiple distinct individuals by the mapping. For further discussion, see the work by Serafini et al. (2005). Hence, while we have shown that in our basic DDL language with bridge rules only between concepts, the injectivity restriction does not cause any harm, in more general DDL languages, especially with heterogeneous bridge rules (Ghidini & Serafini 2006*b*), this approach may no longer be applicable.

4.7 Summary

In this section we have studied the impact of bridge rules on complex concepts that are not directly interconnected by bridge rules but instead they are constructed from more simple concepts that take part in bridge rules directly. As we deal with the basic DDL in which only bridge rules between concepts are allowed, the two concept constructors involved are the concept intersection constructor and the concept union constructor.

We have shown, that if the mapping is expressed with into-bridge rules, DDL allow for sufficient subsumption propagation on complex concepts which are derived by means of concept intersection, or by means of concept union, and even in cases when a combination of these two constructors is used. This does not equally hold for onto-bridge rules however, where for complex concepts constructed by concept intersection, subsumption propagation need not necessarily occur.

In order to cope with this issue, we have proposed two adjustments to the DDL framework. In our first proposal, we have introduced a new type of onto-bridge rules with a modified semantics. These new onto-bridge rules are called conjunctive. There is no need for conjunctive into-bridge rules, as into-bridge rules are not affected by the problem. This new kind of bridge rules turns out to be effective in eliminating the problem that we have previously identified. If conjunctive onto-bridge rules are used instead of normal ones, the amount of subsumption propagation on complex concepts is sufficient. In addition, also other desired semantic properties, as postulated by Borgida & Serafini (2003) and Serafini & Tamilin (2004) are satisfied, and hence at least semantically the approach of conjunctive bridge rules is viable.

Further we have shown, that conjunctive onto-bridge rules are possibly reduced into normal onto-bridge rules, although, in the worst case, an exponential number of additional axioms is generated by the reduction. This means that

by introducing conjunctive bridge rules, we have not extended the expressive power of the framework. Instead, it is possible to understand conjunctive bridge rules as a form of macros: by using them we assert a specific discipline on the modeling with normal onto-bridge rules.

While theoretically, the reduction provides us with a decision procedure for DDL with conjunctive bridge rules, since one is known for the original case (Serafini & Tamilin 2005), practically the worst-case exponential blowup is a serious obstacle. This fact leads us to the second approach which we have investigated in this chapter. We have noticed that the problem is underpinned by the fact, that the domain relation in DDL is not required to be injective. With conjunctive onto-bridge rules, we somehow assert injectivity locally, only for the part of the domain for which it is necessary. It has to be noted, that restricting the domain relation by injectivity has been previously suggested in the literature as a possible solution to this problem, see for instance the discussion by Serafini et al. (2005), but to our best knowledge it has not been studied in depth. We have fostered this approach in the second part of this chapter, where we have investigated the semantic properties of DDL in which injectivity of the domain relations is required universally and globally.

We have shown that, as one might guess, this semantics is stronger than the original DDL semantics, and it is even stronger than the semantics of conjunctive onto-bridge rules. DDL under this new semantics perform equally well with subsumption propagation on complex concepts as in the case when modeling with conjunctive bridge rules. Also, all the desired properties of Borgida & Serafini (2003) and Serafini & Tamilin (2004) are satisfied. We conjecture, that the semantics with injectivity is more suitable for developing a well optimized decision procedure. Later in this work, we will see a tableaux reasoning algorithm, that warrants domain relations to be transitive. We believe, that this algorithm may possibly be adopted also for the case of injectivity.

Chapter 5

Subsumption Propagation and Remote Ontologies in DDL

Subsumption propagation in DDL stands in the center of the attention of this thesis. In Chap. 3, we have learned about the simple subsumption propagation and the generalized subsumption propagation properties (Properties 5 and 6) which have been investigated in the literature (Borgida & Serafini 2003, Serafini & Tamilin 2004, Serafini et al. 2005). Later on, in Chap. 4, we have concentrated on the problem of subsumption propagation on complex concepts in cases when mapping is expressed between more elementary concepts from which these complex concepts are constructed. In all these cases, the scenario involves only two local ontology modules, and it is investigated whether subsumption propagation is caused as a consequence of some combination of onto- and into-bridge rules that map between these local ontologies.

Intended application scenarios of DDL go far beyond such trivial cases. DDL are intended to represent complex distributed knowledge bases of multiple ontologies, arbitrarily interconnected with semantic mapping. Naturally, we are curious about how does subsumption propagate in such complex systems. Notably, whether consequences of local subsumption hierarchy in some local ontology, say $\mathcal{T}_i$, are carried over to local TBoxes that are not connected directly, but are two and more bridge rules away from $\mathcal{T}_i$. In such cases we talk about subsumption propagation between remote ontologies. We investigate subsumption propagation in such complex cases in this chapter.

To illustrate the issue, let us start from a simple subsumption propagation case, as illustrated in Fig. 5.1. Here we deal with two ontologies. On the left is an excerpt from an ontology which contains a general classification of animal species, and on the right is a small ontology in which we intend to model types of things that we keep in our backyard and the relations between them. As we have learned, due to the local subsumption between Felis and Felidae, and the two respective bridge rules, we derive that MyCat is subsumed by DangerousAnimal in the backyard ontology. This matches our intention that we have with this modeling (i.e., our cat is possibly dangerous to our hamster).

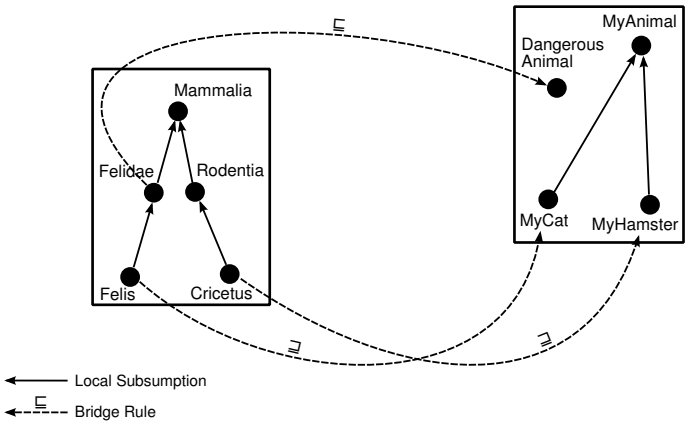

Figure 5.1: Simple case of subsumption propagation in DDL. In this knowledge base it is entailed, that MyCat is a subconcept of DangerousAnimal

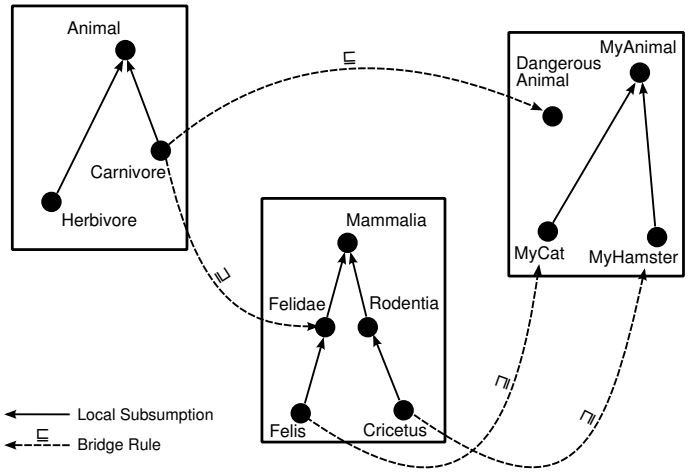

Figure 5.2: Example of complex concept mapping between three ontologies. Even if intuitively one may expect, that subsumption is entailed between MyCat and DangerousAnimal, this is not true under the original DDL semantics

In Fig. 5.2, the situation is similar, yet more complex. In addition to the classification ontology, we now make use also of a third ontology that deals with animal behaviour (on the left). The concept Carnivore is mapped by an into-bridge rule to the concept DangerousAnimal in the backyard ontology. We also see from the figure, that Carnivore is also connected to the concept MyCat, but this connection is indirect. It spans through multiple onto-bridge rules and local subsumptions. Such indirect connections will be called chains of bridge rules. As all the concepts on the way, starting from Carnivore, are always asserted to be more specific by the axioms that form the chain, we would expect that in the end the subsumption between MyCat and DangerousAnimal is entailed, but this is not the case under the original DDL semantics.

The example depicted in Fig. 5.2 contains a chain of onto-bridge rules. Analogously chains of into-bridge rules may also appear in DDL. As we will learn, the original DDL semantics has problems with both kinds of such chains. In this chapter, we take steps forward in order to deliver an adjusted semantics for DDL that would exhibit subsumption propagation along chains of bridge rules. We propose and investigate three possible extensions of the semantics, each derived by some restriction placed on the domain relation. We start by exploiting the compositional consistency requirement, that is known from P-DL (Bao et al. 2006*b*, 2009). We will see that this semantics provides a satisfactory level of subsumption propagation between remote ontologies, however, it fails to satisfy some elementary properties, that are considered important for DDL, namely directionality (Property 2) and local inconsistency (Property 4). Hence this semantics is perhaps too strong. We proceed by gradually weakening the semantics, first by applying the restriction more cautiously and then by requiring only transitivity, which is a weaker condition compared to compositional consistency. The resulting DDL semantics with transitivity satisfies both directionality and local inconsistency properties, however it only handles onto-bridge rules properly, failing to propagate subsumption along chains of into-bridge rules.

In the final part of this chapter, we provide a tableaux reasoning algorithm for DDL with restricted domain relations. As out of the three restricted semantics, that we investigate in this chapter, only the semantics with transitivity is reasonably sound, the reasoning algorithm is tailored specifically for this semantics.

5.1 Analysis

This chapter is concerned with remote ontologies. The basic intuition behind this notion is of two local ontologies within a distributed system, that are not connected directly by a single bridge rule. As we are interested in subsumption propagation, we further refine this notion considering the following two aspects. First, we are only interested in cases when these ontologies are in fact connected by some directed path of bridge rules; this is due to the directionality desideratum which allows no knowledge reuse if this is not the case. Second, even if two ontologies are connected directly, we must still consider also the remote connections between them, that is, paths of bridge rules spanning across multiple ontologies.

The characteristic pattern of interest, when dealing with remote ontologies is captured by the notion of a *chain of bridge rules*. Chain of bridge rules is a directed path of bridge rules which combine together with respect to subconcept-superconcept relations.

Definition 73. *Let $\mathfrak{T}$ be a distributed TBox with an index set I. A chain of bridge rules is either a directed path of into-bridge rules*

$$\langle 1 : C_1 \xrightarrow{\sqsubseteq} 2 : D_2,\ 2 : C_2 \xrightarrow{\sqsubseteq} 3 : D_3,\ \ldots,\ n-1 : C_{n-1} \xrightarrow{\sqsubseteq} n : D_n \rangle$$

of length $n > 0$ such that for each $1 < i < n$ we have $\mathfrak{T} \models_\epsilon i : D_i \sqsubseteq C_i$ (into-chain); or a directed path of onto-bridge rules

$$\langle 1 : C_1 \xrightarrow{\sqsupseteq} 2 : D_2,\ 2 : C_2 \xrightarrow{\sqsupseteq} 3 : D_3,\ \ldots,\ n-1 : C_{n-1} \xrightarrow{\sqsupseteq} n : D_n \rangle$$

of length $n > 0$ such that for each $1 < i < n$ we have $\mathfrak{T} \models_\epsilon i : C_i \sqsubseteq D_i$ (onto-chain).

Let us first have a look, how the original DDL semantics is able to tackle chains of bridge rules. It turns out, that in certain cases subsumption indeed propagates to remote ontologies under this semantics. Given two local ontologies $\mathcal{T}_1$ and $\mathcal{T}_n$ which are not connected directly, but only by two chains of bridge rules, one into-chain and one onto-chain, subsumption possibly propagates in a limited scenario, as characterized by Theorem 78, when these chains consecutively traverse the local ontologies $\mathcal{T}_2, \ldots, \mathcal{T}_n$, one by one, both chains "together".

Theorem 78. *In every distributed TBox $\mathfrak{T}$, such as depicted in Fig. 5.3, with set of bridge rules $\mathfrak{B}$ that features n local TBoxes $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and concepts $C_i, D_i \in \mathcal{T}_i$, for $1 \leq i \leq n$, such that*

1. $C_1 \sqsubseteq D_1$,
2. $i : C_i \xrightarrow{\sqsupseteq} i+1 : C_{i+1} \in \mathfrak{B}$, *for* $1 \leq i < n$,
3. $i : D_i \xrightarrow{\sqsubseteq} i+1 : D_{i+1} \in \mathfrak{B}$, *for* $1 \leq i < n$,

then the following holds: $\mathfrak{T} \models_\epsilon n : C_n \sqsubseteq D_n$.

Proof. By mathematical induction on n. In the base case, for $n = 1$, we get exactly the simple subsumption propagation property, which holds for DDL, as we already know from Chapter 3.

Induction step. Let $n > 1$. We get $\mathfrak{T} \models_\epsilon n-1 : C_{n-1} \sqsubseteq D_{n-1}$ from induction hypothesis. Since we have two bridge rules $n-1 : C_{n-1} \xrightarrow{\sqsupseteq} n : C_n \in \mathfrak{B}$ and $n-1 : D_{n-1} \xrightarrow{\sqsubseteq} n : D_n \in \mathfrak{B}$, we apply simple subsumption propagation once again and thus we derive $\mathfrak{T} \models_\epsilon n : C_n \sqsubseteq D_n$. □

Indeed, it is easy to observe, that Theorem 78 amounts to a straightforward generalization of the simple subsumption propagation property (Property 5). Subsumption propagation occurs between every two consecutive local ontologies on the way between $\mathcal{T}_1$ and $\mathcal{T}_n$ and involves two concepts C_k and D_k inside of each $\mathcal{T}_k$, $1 < k < n$.

It turns out, that the assumption that for each k the concepts C_k and D_k both belong to the very same local ontology is strict. If this requirement is

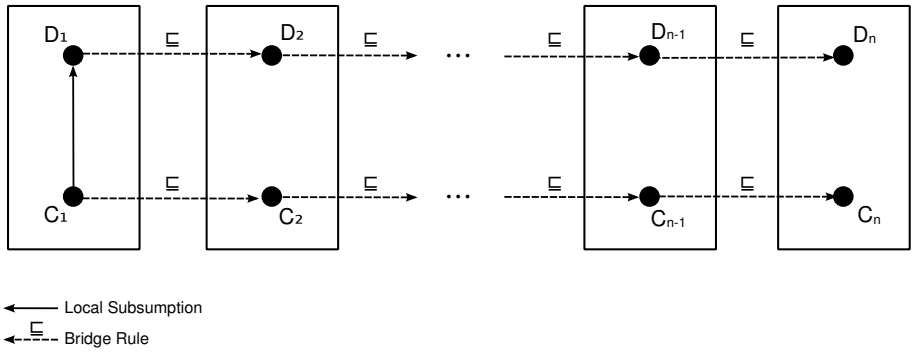

Figure 5.3: Distributed TBox from Theorem 78, representing the subsumption propagation pattern between remote ontologies characteristic for the original DDL semantics

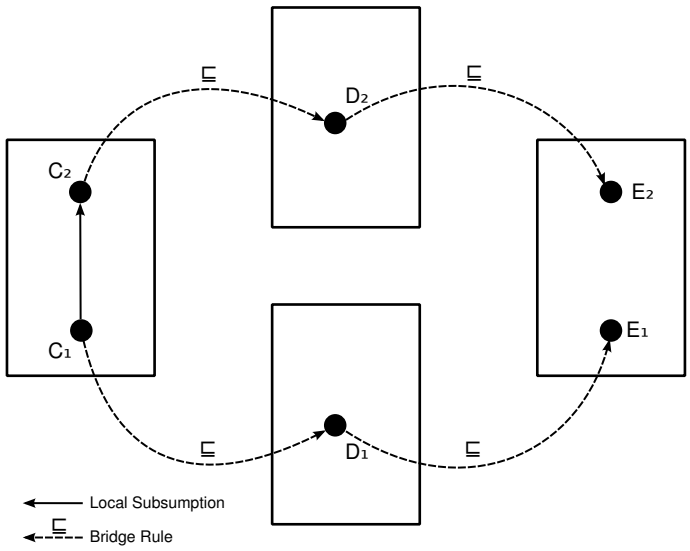

Figure 5.4: Distributed ontology from Example 14. A simple example of problematic chains of bridge rules, along which subsumption does not propagate under the original semantics

violated, subsumption propagation no longer occurs. This fact is demonstrated by the following example, where we encounter a chain of two onto-bridge rules along with a chain of two into-bridge rules. The two concepts in the middle C_2 and D_2 belong each to a separate local ontology $\mathcal{T}_2$ and $\mathcal{T}_{2'}$, respectively. For this reason subsumption propagation does not appear in the example.

Example 14. *Consider a distributed TBox $\mathfrak{T}$, as depicted in Fig. 5.4, featuring local TBoxes $\mathcal{T}_i$, $\mathcal{T}_j$, $\mathcal{T}_{j'}$, and $\mathcal{T}_k$, with concepts $C_1, C_2 \in \mathcal{T}_i$, $D_1 \in \mathcal{T}_j$, $D_2 \in \mathcal{T}_{j'}$, $E_1, E_2 \in \mathcal{T}_k$, such that:*

1. $\mathfrak{T} \models_\epsilon 1 : C_1 \sqsubseteq C_2$,
2. $i : C_2 \xrightarrow{\sqsubseteq} j' : D_2 \in \mathfrak{B}$, $j' : D_2 \xrightarrow{\sqsubseteq} k : E_2 \in \mathfrak{B}$,
3. $i : C_1 \xrightarrow{\sqsupseteq} j : D_1 \in \mathfrak{B}$, $j : D_1 \xrightarrow{\sqsupseteq} k : E_1 \in \mathfrak{B}$.

The subsumption relation between C_1 and C_2 that holds in $\mathcal{T}_i$ does not propagate to $\mathcal{T}_k$, since in each distributed model $\mathfrak{I}$ of $\mathfrak{T}$ the interpretations of the two concepts "on the way" – ${D_1}^{\mathcal{I}_j}$ and ${D_2}^{\mathcal{I}_{j'}}$ – are totally unrelated.

By composition of the bridge rules that are available here we derive the inclusions: ${E_1}^{\mathcal{I}_k} \subseteq r_{jk}\big({D_1}^{\mathcal{I}_j}\big) \subseteq r_{jk}\big(r_{ij}\big({C_1}^{\mathcal{I}_i}\big)\big)$ and $r_{j'k}\big(r_{ij'}\big({C_2}^{\mathcal{I}_i}\big)\big) \subseteq r_{j'k}\big({D_2}^{\mathcal{I}_{j'}}\big) \subseteq {E_2}^{\mathcal{I}_k}$. However $r_{jk}\big(r_{ij}\big({C_1}^{\mathcal{I}_i}\big)\big)$ and $r_{j'k}\big(r_{ij'}\big({C_2}^{\mathcal{I}_i}\big)\big)$ are not related in $\Delta^{\mathcal{I}_k}$.

Observe, that ${C_1}^{\mathcal{I}_i} \subseteq {C_2}^{\mathcal{I}_i}$ in $\Delta^{\mathcal{I}_i}$, and hence also in $\Delta^{\mathcal{I}_k}$ we have $r_{ik}\big({C_1}^{\mathcal{I}_i}\big) \subseteq r_{ik}\big({C_2}^{\mathcal{I}_i}\big)$, however, $r_{jk}\big(r_{ij}\big({C_1}^{\mathcal{I}_i}\big)\big)$ is not related anyhow to $r_{ik}\big({C_1}^{\mathcal{I}_i}\big)$ and neither is $r_{j'k}\big(r_{ij'}\big({C_2}^{\mathcal{I}_i}\big)\big)$ related to $r_{ik}\big({C_2}^{\mathcal{I}_i}\big)$. This observation marks a need for increased compositionality of domain relations within a distributed model, if subsumption propagation over chains of bridge rules is to be achieved.

Another possible view, besides for composition of domain relations, is related to the idea of composition of bridge rules. While there is no notion of bridge rules composition or bridge rules inference in the literature, and also in this work we do not intend to introduce this kind of reasoning with bridge rules formally, we will use the notion of bridge rules composition as a metaphor in what follows. The point is, that the desired extent of subsumption propagation according to our intuitions is characterized by a hypothetical semantics in which each chain of bridge rules between C of $\mathcal{T}_i$ and E of $\mathcal{T}_k$ has the power equal to a single bridge rule between these two concepts. In the basic case this propagation pattern is characterized by the following composition of bridge rules:

$$i : C \xrightarrow{\sqsubseteq} j : D \text{ and } j : D \xrightarrow{\sqsubseteq} k : E \qquad \text{causes} \qquad i : C \xrightarrow{\sqsubseteq} k : E \ ,$$
$$i : C \xrightarrow{\sqsupseteq} j : D \text{ and } j : D \xrightarrow{\sqsupseteq} k : E \qquad \text{causes} \qquad i : C \xrightarrow{\sqsupseteq} k : E \ .$$

Semantically these two composition rules correspond to the following two conditions (in the respective order):

$$r_{ij}(C^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_j} \wedge r_{jk}(D^{\mathcal{I}_j}) \subseteq E^{\mathcal{I}_k} \implies r_{ik}(C^{\mathcal{I}_i}) \subseteq E^{\mathcal{I}_k} \ ,$$
$$r_{ij}(C^{\mathcal{I}_i}) \supseteq D^{\mathcal{I}_j} \wedge r_{jk}(D^{\mathcal{I}_j}) \supseteq E^{\mathcal{I}_k} \implies r_{ik}(C^{\mathcal{I}_i}) \supseteq E^{\mathcal{I}_k} \ .$$

Since the above two conditions must hold for any interpretation of C, D and E, then they correspond to the following inclusions between domain relations:

$$r_{ij} \circ r_{jk} \supseteq r_{ik} \ , \tag{5.1}$$
$$r_{ij} \circ r_{jk} \subseteq r_{ik} \ . \tag{5.2}$$

Notice, that when any two of i, j and k are equal, the above properties include r_{ii} for some $i \in I$, which is not defined in a DDL. Hence, the above properties are only applicable in case when i, j and k are three mutually distinct indices.

And so, we again conclude, that certain amount of compositionality should be perhaps exhibited by the domain relations within a distributed model. Our investigation proceeds with studying the properties of an amended semantics of DDL which obeys restrictions (5.1) and (5.2).

5.2 Coping with Chains of Bridge Rules

In the previous section, we have learned about remote ontologies and chains of bridge rules, and we have discussed situations in which subsumption propagation occurs between remote ontologies under the original DDL semantics. We have learned that the amount of subsumption propagation along chains of bridge rules of length two and higher is rather low. Subsumption does not propagate in scenarios such as the one of Example 14, which is depicted in Fig. 5.4, nor in the scenario depicted in Fig. 5.2. In both these cases we have argued, that such an outcome is unintuitive, and that subsumption propagation would be intuitively expected.

Our analysis of this problem shows, that unrestricted domain relations, which are employed by DDL are not satisfactory if propagation of subsumption between remote ontologies is to be achieved, because there is no compositionality between these relations. By this "lack of compositionality" we understand the fact, which we have observed, that the two domain relations r_{ij} and r_{jk}, between some local ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$, and $\mathcal{T}_j$ and $\mathcal{T}_k$ respectively, have no influence on the domain relation r_{ik} between $\mathcal{T}_i$ and $\mathcal{T}_k$. And also vice versa, r_{ik} poses no restrictions on r_{ij} and r_{jk}.

In this section, we investigate three variants of the DDL semantics, in which the domain relations are restricted, and compositionality between them is required. We will see, however, that the solution to the problem is not straightforward, and there is a certain amount of trade-off. If the restrictions placed on the domain relations are too strong, certain desirable properties of the semantics may be lost. On the other hand, if the restrictions are too weak, other desired properties of the semantics are maintained, but subsumption propagation is not increased to the expected level.

5.2.1 DDL with Compositionally Consistent Domain Relations

Our analysis above suggests, that the two conditions (5.1) and (5.2) could possibly increase the amount of subsumption propagation along chains of onto- and into-bridge rules, if imposed on the domain relations in distributed models. Combining both of them, we will require that the composition of any two adjoint domain relations r_{ij} and r_{jk} must be exactly equal to r_{ik}. This requirement is not new in the literature dealing with distributed and modular ontologies. It is called compositional consistency, and it has been previously applied in Package-based Description Logics (Bao et al. 2006*b*, 2009). The semantics of P-DL, also called semantics of importing, makes use of domain relations too, however these are rather strictly constrained. We will borrow the compositional consistency

restriction, and investigate its application in the setting of DDL. See Chap. 6 for more information on P-DL and on the relation of P-DL and DDL. In the setting of DDL, compositional consistency is formalized as follows.

Definition 74 (Compositional consistency)**.** *Given a distributed interpretation $\mathfrak{I}$ with domain relation r, we say that r (and also $\mathfrak{I}$) satisfies the compositional consistency requirement, if for any three mutually distinct indices $i, j, k \in I$ we have $r_{ij} \circ r_{jk} = r_{ik}$.*

Now, the adjusted semantics is simply obtained from the original DDL semantics by allowing only distributed interpretations that satisfy compositional consistency.

Definition 75 (Distributed model under compositional consistency)**.** *Given a DDL knowledge base $\mathfrak{T}$ over an index set I. A distributed interpretation $\mathfrak{I}$ is a model of $\mathfrak{T}$ under the DDL semantics with compositional consistency, if it is a distributed model of $\mathfrak{T}$ in the sense of Definition 59 and if its domain relation satisfies the compositional consistency requirement.*

In accordance, we often use the wording "DDL under compositional consistency" when referring to this semantics. Also, in short we sometimes say "compositionality" instead of "compositional consistency".

The adjusted semantics is actually a proper strengthening of the original one. If any subsumption formula ϕ is entailed by a distributed TBox $\mathfrak{T}$ in the original semantics, then it is also entailed by $\mathfrak{T}$ under compositional consistency. The only difference is that in the adjusted semantics possibly some more subsumption formulae are entailed in addition. This is formally stated in the following theorem.

Theorem 79. *Given a distributed TBox $\mathfrak{T}$ and a subsumption formula ϕ, if $\mathfrak{T} \models_\epsilon \phi$ according to the original semantics, then $\mathfrak{T} \models_\epsilon \phi$ also holds in DDL under compositional consistency.*

Proof. This follows from the fact that each model that satisfies compositional consistency is also a model in the original DDL semantics. If a formula ϕ is satisfied in each model from the set of models of $\mathfrak{T}$ according to the original semantics, it is also satisfied by each model from its subset – the set of models of $\mathfrak{T}$ that we obtain under compositional consistency. □

In order to demonstrate the effect of the compositional consistency requirement we revisit the example from Fig. 5.2 more formally. We will see that domain relations, which are now constrained under the new semantics, assure far more subsumption propagation than we have previously observed.

Example 15. *Recall the example depicted in Fig. 5.2. Let us simplify the notation by renaming the concepts as follows: $C :=$ MyCat, $D :=$ DangerousAnimal, $E :=$ Felis, $F :=$ Felidae, $G :=$ Carnivore. Remaining concepts are of no interest (see Fig. 5.5). Assume the index set $I = \{b, c, y\}$, where b represents the behaviour ontology (on the left), c represents the classification ontology (in the middle) and y represents the backyard ontology (on the right). There are various axioms in $\mathfrak{T} = \langle\{\mathcal{T}_b, \mathcal{T}_c, \mathcal{T}_y, \}, \mathfrak{B}\rangle$, but of the GCI only $c : E \sqsubseteq F$, actually*

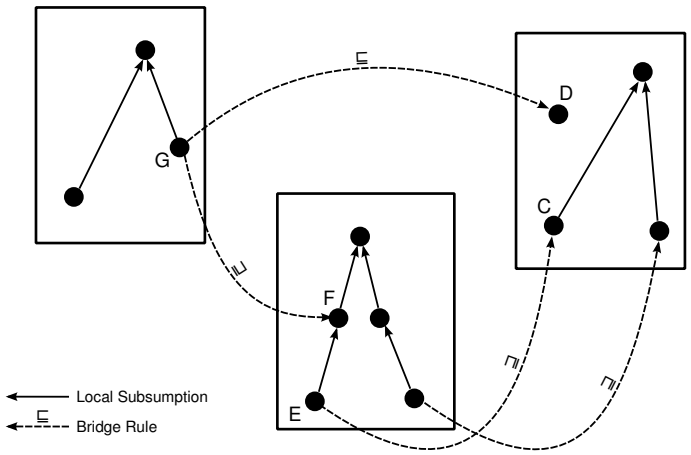

Figure 5.5: Renaming of concepts from Fig. 5.2 employed in Example 15. The three local ontologies are referred to as $\mathcal{T}_b$ – behaviour ontology (left), $\mathcal{T}_c$ – classification ontology (bottom), and $\mathcal{T}_y$ – backyard ontology (right)

matters to us, and there are three bridge rules in $\mathfrak{B}$*:*

$$b : G \xrightarrow{\sqsupseteq} c : F \ , \qquad c : E \xrightarrow{\sqsupseteq} y : C \ ,$$
$$b : G \xrightarrow{\sqsubseteq} y : D \ .$$

We query whether it holds that $\mathfrak{T} \models_\epsilon y : C \sqsubseteq D$*. Assume a distributed interpretation* $\mathfrak{I}$*. Let us first assume that* $\mathfrak{I}$ *contains no hole. From Definition 59 we get* $r_{cy}\big(E^{\mathcal{I}_c}\big) \supseteq C^{\mathcal{I}_y}$*. We also have* $r_{bc}\big(G^{\mathcal{I}_b}\big) \supseteq F^{\mathcal{I}_c}$*, but since* $r_{cy}(\cdot)$ *is a mapping we get that* $r_{cy}\big(r_{bc}\big(G^{\mathcal{I}_b}\big)\big) \supseteq r_{cy}\big(F^{\mathcal{I}_c}\big)$*. And from compositional consistency we get that* $r_{cy}\big(r_{bc}\big(G^{\mathcal{I}_b}\big)\big) = r_{by}\big(G^{\mathcal{I}_b}\big)$ *implies* $r_{by}\big(G^{\mathcal{I}_b}\big) \supseteq r_{cy}\big(F^{\mathcal{I}_c}\big)$*. From the GCI* $c : E \sqsubseteq F$ *we get* $F^{\mathcal{I}_c} \supseteq E^{\mathcal{I}_c}$ *and from properties of mapping we again get* $r_{cy}\big(F^{\mathcal{I}_c}\big) \supseteq r_{cy}\big(E^{\mathcal{I}_c}\big)$*. Putting this all together we derive:*

$$r_{by}\big(G^{\mathcal{I}_b}\big) \supseteq r_{cy}\big(F^{\mathcal{I}_c}\big) \supseteq r_{cy}\big(E^{\mathcal{I}_c}\big) \supseteq C^{\mathcal{I}_y} \ ,$$

and that amounts to $r_{by}\big(G^{\mathcal{I}_b}\big) \supseteq C^{\mathcal{I}_y}$*. On the other hand, from the into-bridge rule between* $\mathcal{T}_b$ *and* $\mathcal{T}_y$ *we derive* $r_{by}\big(G^{\mathcal{I}_b}\big) \subseteq D^{\mathcal{I}_y}$*. And so we finally get* $C^{\mathcal{I}_y} \subseteq D^{\mathcal{I}_y}$*.*

For the case with holes assume for instance that $\mathcal{I}_b = \mathcal{I}^\epsilon$*. In that case* $G^{\mathcal{I}_b} = \emptyset$*. Thanks to the constraint generated by bridge rules and the GCI we easily derive that also* $F^{\mathcal{I}_c} = \emptyset$*,* $E^{\mathcal{I}_c} = \emptyset$*, and also* $C^{\mathcal{I}_y} = \emptyset$*. In such a case, however,* $C^{\mathcal{I}_y} \subseteq D^{\mathcal{I}_y}$ *holds trivially. If we substitute other local interpretations for holes, we get the very same result analogously.*

Summing up, in every model of $\mathfrak{T}$ *we have* $C^{\mathcal{I}_y} \subseteq D^{\mathcal{I}_y}$ *and hence* $\mathfrak{T} \models_\epsilon y : C \sqsubseteq D$*. Recall that we have actually used the compositional consistency requirement in our argumentation. Without it, we would not be able to establish the result: we would not be able to prove that* $r_{by}\big(G^{\mathcal{I}_b}\big) \supseteq r_{cy}\big(F^{\mathcal{I}_c}\big)$*. In fact, the original DDL semantics allows models that violate this inclusion and hence* $\mathfrak{T} \models_\epsilon y : C \sqsubseteq D$ *does not hold under the original semantics.*

A more general characterization of the cases when subsumption propagates to remote ontologies follows below in Theorem 80. This characterization generalizes the settings from Examples 14–15. As the theorem shows, under the adjusted semantics subsumption does in fact propagate in each of the cases when we have previously observed lack of subsumption propagation. In a nutshell, Theorem 80 basically says that the effect of bridge rules is now transitive, and hence subsumption propagates even between remote ontologies within a DDL knowledge base. More specifically, if there are two chains of bridge rules between two local ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$, one into-chain between the concepts C and G, and one onto-chain between the concepts D and H, then the semantics allows for as much subsumption propagation as if the concepts C and G were connected by a single into-bridge rule and D and H by a single onto-bridge rule. Unlike in Theorem 78, the concepts "on the way" in the into-chain are independent from the concepts that take part in the onto-chain; each of the chains is free to traverse through a completely different set of local ontologies.

Theorem 80. *Given a distributed TBox, as illustrated in Fig. 5.6, with an index set I and a set of bridge rules $\mathfrak{B}$, that features $n+1$ local TBoxes $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n$ with concepts $E, F \in \mathcal{T}_0$, and $C_i, D_i \in \mathcal{T}_i$, for $1 \leq i \leq n$, and k with $1 \leq k \leq n$ such that:*

1. $\mathfrak{T} \models_\epsilon i : C_i \sqsubseteq D_i$, *for* $1 \leq i \leq n$,
2. $i+1 : C_{i+1} \xrightarrow{\sqsupseteq} i : D_i \in \mathfrak{B}$, *for* $1 \leq i < k$,
3. $i : D_i \xrightarrow{\sqsubseteq} i+1 : C_{i+1} \in \mathfrak{B}$, *for* $k \leq i < n$,
4. $1 : C_1 \xrightarrow{\sqsupseteq} 0 : E \in \mathfrak{B}$ *and* $n : D_n \xrightarrow{\sqsubseteq} 0 : F \in \mathfrak{B}$.

In DDL under compositional consistency it follows that $\mathfrak{T} \models_\epsilon 0 : E \sqsubseteq F$.

Proof. Let $\mathfrak{T}$ be a distributed TBox with index set I, such that the assumptions of the theorem are satisfied. Let there be some distributed interpretation $\mathfrak{I}$ of $\mathfrak{T}$ such that $\mathfrak{I} \models_\epsilon \mathfrak{T}$. We prove the theorem by proving the following chain of inclusions:

$$E^{\mathcal{I}_0} \overset{1}{\subseteq} r_{k0}\big({C_k}^{\mathcal{I}_k}\big) \overset{2}{\subseteq} r_{k0}\big({D_k}^{\mathcal{I}_k}\big) \overset{3}{\subseteq} F^{\mathcal{I}_0} \quad ,$$

which implies that $\mathfrak{T} \models_\epsilon 0 : E \sqsubseteq F$.

We start by Inclusion 2. Consider the assumption $\mathfrak{T} \models_\epsilon k : C_k \sqsubseteq D_k$. This implies ${C_k}^{\mathcal{I}_k} \subseteq {D_k}^{\mathcal{I}_k}$ and since $r_{k0}(\cdot)$ is a mapping, we also have $r_{k0}\big({C_k}^{\mathcal{I}_k}\big) \subseteq r_{k0}\big({D_k}^{\mathcal{I}_k}\big)$. Inclusion 3 divides into:

$$r_{k0}\big({D_k}^{\mathcal{I}_k}\big) \overset{4}{=} r_{n0}\big(r_{n-1n}\big(r_{n-2n-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big({D_k}^{\mathcal{I}_k}\big)\big)\cdots\big)\big)\big) \overset{5}{\subseteq} F^{\mathcal{I}_0} \quad .$$

We start with Equation 4 on the left hand side. We first prove, by mathematical induction on n, an auxiliary equation:

$$r_{kn}\big({D_k}^{\mathcal{I}_k}\big) = r_{n-1n}\big(r_{n-2n-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big({D_k}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \quad .$$

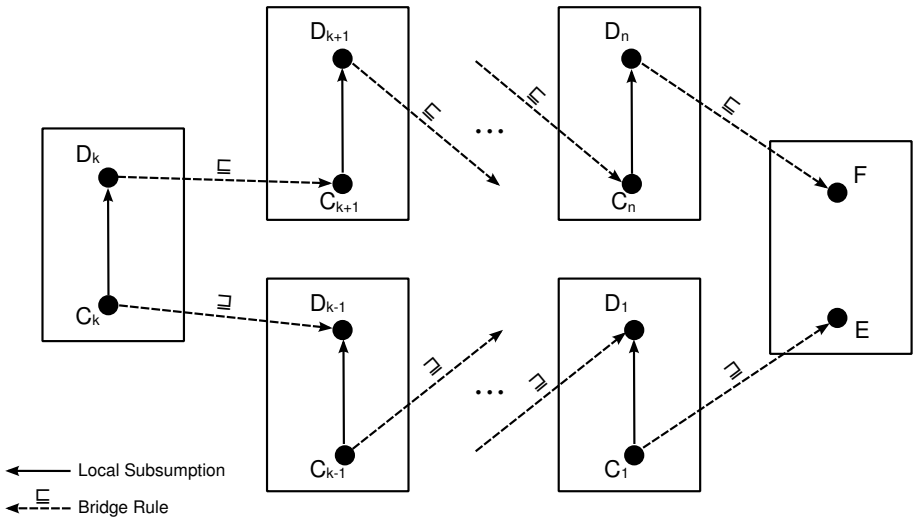

Figure 5.6: Distributed TBox from Theorem 80, representing the general subsumption propagation patterns that occur in the DDL semantics with compositional consistency

For the base case consider $n = k + 2$. We ought to prove that $r_{kk+2}\big(D_k{}^{\mathcal{I}_k}\big) = r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)$, but this follows directly from compositional consistency. For the induction step consider $n = m$ such that $k+2 < m$. We ought to prove:

$$r_{km}\big(D_k{}^{\mathcal{I}_k}\big) = r_{m-1m}\big(r_{m-2m-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \quad .$$

From induction hypothesis we get:

$$r_{km-1}\big(D_k{}^{\mathcal{I}_k}\big) = r_{m-2m-1}\big(r_{m-3m-2}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \quad .$$

Thanks to the latter equation we are able to make a replacement in the former one and we get:

$$r_{km}\big(D_k{}^{\mathcal{I}_k}\big) = r_{m-1m}\big(r_{km-1}\big(D_k{}^{\mathcal{I}_k}\big)\big) \quad ,$$

which remains to be proven. But this equation once again follows directly from the fact that the domain relation r satisfies the compositional consistency requirement. Now, thanks to our auxiliary equation we make a replacement in Equation 4 and it remains to prove that:

$$r_{k0}\big(D_k{}^{\mathcal{I}_k}\big) = r_{n0}\big(r_{kn}\big(D_k{}^{\mathcal{I}_k}\big)\big) \quad ,$$

which we again get directly from compositional consistency. And so we have proven Equation 4.

We now continue with Inclusion 5. We first prove the auxiliary equation:

$$r_{n-1n}\big(r_{n-2n-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq D_n{}^{\mathcal{I}_n} \quad .$$

The proof is again by mathematical induction on n. For the base case $n = k + 1$ and we shall prove that $r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big) \subseteq D_{k+1}{}^{\mathcal{I}_n}$. We get this from the

assumptions of the theorem. We have assumed the bridge rule $k : D_k \xrightarrow{\sqsubseteq} k+1 : C_{k+1} \in \mathfrak{B}$ which implies $r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big) \subseteq C_{k+1}{}^{\mathcal{I}_{k+1}}$. We have also assumed that $\mathfrak{T} \models_\epsilon k+1 : C_{k+1} \sqsubseteq D_{k+1}$, and so $C_{k+1}{}^{\mathcal{I}_{k+1}} \subseteq D_{k+1}{}^{\mathcal{I}_{k+1}}$. By combining these two inclusions we get the one we needed to prove. For the induction step consider $n = m$ such that $m > k+1$. We shall prove:

$$r_{m-1m}\big(r_{m-2m-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq D_m{}^{\mathcal{I}_m} \quad .$$

By induction hypothesis we are free to assume:

$$r_{m-2m-1}\big(r_{m-3m-2}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq D_{m-1}{}^{\mathcal{I}_{m-1}} \quad .$$

As $r_{m-1m}(\cdot)$ is a mapping, it follows that if we apply it on both sides of the inclusion above, the inclusion still holds, that is:

$$r_{m-1m}\big(r_{m-2m-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq r_{m-1m}\big(D_{m-1}{}^{\mathcal{I}_{m-1}}\big) \quad .$$

Now, we have assumed the bridge rule $m-1 : D_{m-1} \xrightarrow{\sqsubseteq} m : C_m \in \mathfrak{B}$ which implies $r_{m-1m}\big(D_{m-1}{}^{\mathcal{I}_{m-1}}\big) \subseteq C_m{}^{\mathcal{I}_m}$. Also we have assumed that $\mathfrak{T} \models_\epsilon m : C_m \sqsubseteq D_m$ which amounts to $C_m{}^{\mathcal{I}_m} \subseteq D_m{}^{\mathcal{I}_m}$. The last three inclusions that we have shown give us the auxiliary inclusion above that we wanted to prove. Now, in order to get Inclusion 5, we apply the mapping $r_{n0}(\cdot)$ on both sides of this auxiliary inclusion and we get:

$$r_{n0}\big(r_{n-1n}\big(r_{n-2n-1}\big(\cdots r_{k+1k+2}\big(r_{kk+1}\big(D_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big)\big) \subseteq r_{n0}\big(D_n{}^{\mathcal{I}_n}\big) \quad .$$

Finally, looking once again in between the assumptions of the theorem we find there the bridge rule $n : D_n \xrightarrow{\sqsubseteq} 0 : F \in \mathfrak{B}$ which implies $r_{n0}\big(D_n{}^{\mathcal{I}_n}\big) \subseteq F^{\mathcal{I}_0}$. The last two inclusions combined together result exactly into Inclusion 5.

It remains to prove Inclusion 1. This inclusion divides into:

$$E^{\mathcal{I}_0} \subseteq r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) = r_{k0}\big(C_k{}^{\mathcal{I}_k}\big) \quad .$$

Both these inclusions ought to be proven by mathematical induction. The first inclusion is proven by induction on k, base case is $E^{\mathcal{I}_0} \subseteq r_{10}\big(C_1{}^{\mathcal{I}_1}\big)$. This is a direct consequence of the onto-bridge rule $1 : C_1 \xrightarrow{\sqsupseteq} 0 : E \in \mathfrak{B}$. In the induction step, by the induction hypothesis we have:

$$E^{\mathcal{I}_0} \subseteq r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big(r_{k-1k-2}\big(C_{k-1}{}^{\mathcal{I}_{k-1}}\big)\big)\cdots\big)\big) \quad .$$

From the assumptions of the theorem we derive:

$$C_{k-1}{}^{\mathcal{I}_{k-1}} \subseteq D_{k-1}{}^{\mathcal{I}_{k-1}} \subseteq r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big) \quad .$$

The composition of mappings $r_{10}(r_{21}(\cdots r_{k-1k-2}(\cdot)\cdots))$ is still a mapping and so by applying it on the above inclusion we get:

$$r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big(C_{k-1}{}^{\mathcal{I}_{k-1}}\big)\cdots\big)\big) \subseteq r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \quad .$$

Together with the induction hypothesis this amounts to the inclusion we wanted to prove.

As for the second inclusion, we shall prove a slightly more general proposition:

$$r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq r_{kn}\big(C_k{}^{\mathcal{I}_k}\big) \quad ,$$

where $0 \leq n \leq k-2$. The inclusion we ought to prove is then derived by setting $n = 0$. The proposition is proven by mathematical induction on $k - n$. As for the base case consider $k - n = 2$, and so $n = k - 2$. That means we have to show $r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big) \subseteq r_{kk-2}\big(C_k{}^{\mathcal{I}_k}\big)$, which indeed holds because $\mathfrak{I}$ satisfies the transitivity condition. The induction step is for $k - n = j > 2$, and so $n = k - j$. From the induction hypothesis we get:

$$r_{n+2n+1}\big(r_{n+3n+2}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big) \quad .$$

And since $r_{n+1n}(\cdot)$ is a mapping, when we apply it to the above inclusion we get:

$$r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq r_{n+1n}\big(r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big)\big) \quad .$$

Since $\mathfrak{I}$ satisfies the transitivity condition, then the mappings must satisfy $r_{n+1n}\big(r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big)\big) \subseteq r_{kn}\big(C_k{}^{\mathcal{I}_k}\big)$. By combining the last two inclusions we get exactly:

$$r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big)\cdots\big)\big) \subseteq r_{kn}\big(C_k{}^{\mathcal{I}_k}\big) \quad .$$

By setting $n = 0$ we derive the inclusion that we wanted to prove, and hence the proof is concluded. □

And so, we have learned that subsumption propagation is much improved under the new DDL semantics with compositionally consistent domain relations. In the light of our analysis from Sect. 5.1 we conclude, that this level of subsumption propagation between remote ontologies is satisfactory accurate. Also the monotonicity property is satisfied by this semantics.

Theorem 81. *The monotonicity property is satisfied in DDL under the semantics with compositional consistency.*

Proof. Given $\mathcal{T}_i \in \mathfrak{T}$ and $C, D \in \mathcal{T}_i$ such that $\mathcal{T}_i \models_\epsilon C \sqsubseteq D$, only models of $\mathcal{T}_i$ and $\mathcal{I}^\epsilon$ are allowed in the $\mathcal{I}_i$ slot of any distributed model $\mathfrak{I}$ of $\mathfrak{T}$. Since in each of these $C \sqsubseteq D$ holds, we get $\mathfrak{T} \models_\epsilon i : C \sqsubseteq D$ directly from the definition. □

Unfortunately, the newly introduced semantics also exhibits some undesirable side effects. In particular, the introduction of compositionality to this extent invalidates the directionality and the local inconsistency properties.[1] Let us take a closer look at this problem. The following example presents a scenario in which directionality (Property 2) is violated.

Example 16. *Consider the following distributed TBox $\mathfrak{T}$ with the index set $I = \{1, 2, 3\}$, such that $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \emptyset$, $\mathcal{T}_3 = \emptyset$ and two bridge rules $\mathfrak{B} = \{2 : C \xrightarrow{\sqsubseteq} 1 : \bot, 2 : C \xrightarrow{\sqsupseteq} 3 : D\}$.*

Under the semantics with compositional consistency, it follows that $\mathfrak{T} \models_\epsilon 3 : D \sqsubseteq \bot$. There is no directed path from $\mathcal{T}_1$ to $\mathcal{T}_3$. If $\mathfrak{T}$ satisfies the directionality property, it should be possible to remove $\mathcal{T}_1$ from $\mathfrak{T}$ without losing any of the entailed subsumptions. But this is not the case since $\mathfrak{T}^{-1} \not\models_\epsilon 3 : D \sqsubseteq \bot$. The proof that $\mathfrak{T} \models_\epsilon 3 : D \sqsubseteq \bot$ is as follows:

[1] Please note, that in one of our previous reports (Homola 2008) we have erroneously claimed the contrary (i.e., that both directionality and local inconsistency hold under the DDL semantics with compositional consistency). This is not true, as shown by Examples 16 and 17.

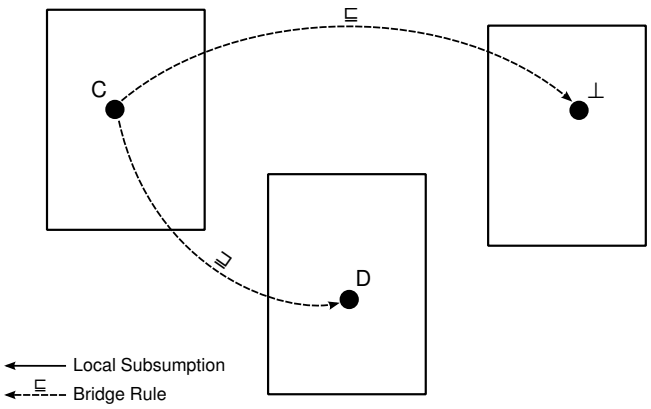

Figure 5.7: Distributed TBox from Example 16, demonstrating an apparent violation of directionality

1. $\mathfrak{T} \models_\epsilon 3 : D \sqsubseteq \bot$ *because by the second bridge rule,* $r_{23}\big(C^{\mathcal{I}_2}\big) \supseteq D^{\mathcal{I}_3}$ *and* $C^{\mathcal{I}_2} = \emptyset$*, as we will prove;*
2. *assume* $r_{23}\big(C^{\mathcal{I}_2}\big) \neq \emptyset$*;*
3. *hence there are* $x \in C^{\mathcal{I}_2}$ *and* $y \in \Delta^{\mathcal{I}_3}$ *such that* $y \in r_{23}(x)$*;*
4. *from compositional consistency there must be* $z \in \Delta^{\mathcal{I}_1}$ *such that* $z \in r_{21}(x)$ *and* $y \in r_{13}(z)$*;*
5. *but from the first bridge rule we now get* $z \in \bot^{\mathcal{I}_1}$ *which is a contradiction, as* $x \in C^{\mathcal{I}_2}$*,* $z \in r_{21}(x)$ *and* $r_{21}\big(C^{\mathcal{I}_2}\big) \subseteq \bot^{\mathcal{I}_1}$*.*

Also the local inconsistency property (Property 4) is violated under the newly introduced semantics, as shown by the next example.

Example 17. *Consider a distributed TBox* $\mathfrak{T}$ *with index set* $I = \{1, 2, 3\}$ *and a set of bridge rules* $\mathfrak{B}$*, such that* $\mathcal{T}_1 = \{\top \sqsubseteq \bot\}$*,* $\mathcal{T}_2 = \emptyset$*,* $\mathcal{T}_3 = \emptyset$ *which only contains one bridge rule*

$$2 : C \xrightarrow{\sqsupseteq} 3 : D \ .$$

Since $\mathcal{T}_1$ *is inconsistent, the only admissible local interpretation is a hole. Similarly as in the example above,* $r_{23} = \emptyset$*, because otherwise* $\Delta^{\mathcal{I}_1}$ *would be non-empty. Hence* $\mathfrak{T} \models_\epsilon 3 : D \sqsubseteq \bot$*. If local inconsistency would hold, then for each* $J \subseteq I$*,* $3 \notin J$ *we must have* $\mathfrak{T}(\epsilon_J) \models_{\mathrm{d}} 3 : D \sqsubseteq \bot$*. For* $J = \{1\}$ *this does not hold, however, as we will show. In this case* $\mathfrak{T}(\epsilon_J) = \langle\{\mathcal{T}_2, \mathcal{T}_3\}, \mathfrak{B}\rangle$*. Consider the d-interpretation* $\mathfrak{I}$ *with* $\Delta^{\mathcal{I}_2} = C^{\mathcal{I}_2} = \{x\}$*,* $\Delta^{\mathcal{I}_3} = D^{\mathcal{I}_3} = \{y\}$*,* $r_{23} = \{\langle x, y\rangle\}$ *and* $r_{32} = \emptyset$*. Clearly,* $\mathfrak{I}$ *is a d-model of* $\mathfrak{T}(\epsilon_J)$*, computational consistency is satisfied by* r*, and indeed* $\mathfrak{I} \not\models_{\mathrm{d}} 3 : D \sqsubseteq \bot$ *since* $D^{\mathcal{I}_3} \neq \emptyset$*.*

In this section we have adjusted the semantics of DDL in order to deal with modeling scenarios in which subsumption propagation does not occur to the desired extent, as we have explained and motivated. The adjustment places further restriction, so called compositional consistency, that we borrow from P-DL

(Bao et al. 2006*b*), on the domain relation in each admissible distributed model. We have formally proven that under such semantics subsumption propagation is increased, especially between remote ontologies within a DDL knowledge base, that are only connected indirectly by bridge rules.

As shown by Examples 16 and 17, important properties of the original DDL semantics are violated under the compositional consistency requirement. Willing to preserve these properties, most notably directionality, we see two possible directions: the first option is to apply the condition (5.1) more cautiously, for instance, only in cases when $\langle i, j\rangle$ and $\langle j, k\rangle$ are connected in the bridge graph; the other option is to drop condition (5.1) completely, that is, to fall back from compositional consistency to transitivity. In the following two sections we investigate both these directions.

5.2.2 DDL with Restricted Compositionality

We will now follow the first of the two directions, that we have outlined above. In this approach we will apply both conditions (5.1) and (5.2), but we will try to apply them more cautiously. This idea is motivated by the following example.

Example 18. *Consider a distributed TBox $\mathfrak{T}$ which contains local ontologies about cats $\mathcal{T}_c$, music $\mathcal{T}_m$, botany $\mathcal{T}_b$ and another one about fruit $\mathcal{T}_f$. Suppose that there is some knowledge encoded within each of them and they are all consistent. There are also some bridge rules between $\mathcal{T}_b$ and $\mathcal{T}_f$ for instance:*

$$b : \mathsf{Fruit} \xrightarrow{\sqsubseteq} f : \mathsf{Fruit} \ , \qquad b : \mathsf{Berry} \xrightarrow{\sqsupseteq} f : \mathsf{Berry} \ .$$

In contrast the two local ontologies $\mathcal{T}_c$ and $\mathcal{T}_t$ are completely isolated. Let $\mathfrak{I}$ be a model of $\mathfrak{T}$ in which $\mathsf{Berry}^{\mathcal{I}_f}$ is nonempty and hence there are some $x \in \mathsf{Berry}^{\mathcal{I}_b}$, $y \in \mathsf{Berry}^{\mathcal{I}_f}$ such that $\langle x, y\rangle \in r_{bf}$. So far this situation is perfectly natural. However, compositional consistency requires existence of $z_1 \in \Delta^{\mathcal{I}_c}$ and $z_2 \in \Delta^{\mathcal{I}_m}$ such that $\langle x, z_1\rangle \in r_{bc}$, $\langle z_1, y\rangle \in r_{cf}$, $\langle x, z_2\rangle \in r_{bm}$ and $\langle z_2, y\rangle \in r_{mf}$. In turn, $\langle x, z_1\rangle \in r_{bc}$ implies existence of $\langle x, w_1\rangle \in r_{bm}$, $\langle w_1, z_1\rangle \in r_{mc}$, $\langle x, w_2\rangle \in r_{bf}$ and $\langle w_2, z_1\rangle \in r_{fc}$, etc. As we can see, even if local TBoxes $\mathcal{T}_c$ and $\mathcal{T}_m$ are totally unrelated, compositional consistency implies existence of elements in $\mathcal{I}_c$ and $\mathcal{I}_m$ and generates many r-connections between these elements and elements of the other local TBoxes. Such behaviour is indeed rather strange and as we have seen it even causes trouble in the sense that some important DDL properties are violated.

From the example above we conjecture that compositional consistency is perhaps a stronger condition than we really need in order to support subsumption propagation. It does the job, as established by Theorem 80, but in addition it has undesirable side effects. In the previous section, we have applied compositional consistency universally. Now we take a look at a different approach, in which this condition is applied more carefully. Given a distributed TBox and a distributed interpretation over some index set I, we do not require compositional consistency to hold for every $i, j, k \in I$, but only when local TBoxes $\mathcal{T}_i$, $\mathcal{T}_j$ and $\mathcal{T}_k$ are actually connected with bridge rules. The relaxed version of the restriction is as follows.

Definition 76 (Restricted compositionality)**.** *Given a distributed TBox $\mathfrak{T}$ with an index set I and bridge graph $\mathrm{G}_{\mathfrak{T}}$, let $\mathfrak{I}$ be a distributed interpretation of $\mathfrak{T}$ with*

domain relation r, we say that r (and also $\mathfrak{I}$) satisfies restricted compositional consistency (also restricted compositionality), if for any three mutually distinct indices $i, j, k \in I$ such that there is a directed path from i to j and another one from j to k in $\mathrm{G}_{\mathfrak{T}}$ we have $r_{ij} \circ r_{jk} = r_{ik}$.

The respective semantics of DDL is derived by allowing distributed models that satisfy restricted compositionality. In accordance, we refer to this semantics as DDL under restricted compositionality.

Definition 77 (Distributed model under restricted compositionality)**.** *Given a DDL knowledge base $\mathfrak{T}$ over an index set I. A distributed interpretation $\mathfrak{I}$ is a model of $\mathfrak{T}$ under the DDL semantics with restricted compositionality, if it is a distributed model of $\mathfrak{T}$ in the sense of Definition 59 and if its domain relation satisfies the restricted compositionality requirement.*

As one might easily guess, this restricted semantics is also a strengthening of the original DDL semantics. Furthermore, if we are to order the three semantics according to their strength, the semantics with restricted compositionality is to be placed between the two which we got to know before.

Theorem 82. *Given a distributed TBox $\mathfrak{T}$ and a subsumption formula ϕ, if $\mathfrak{T} \models_\epsilon \phi$ according to the original semantics, then $\mathfrak{T} \models_\epsilon \phi$ also holds in DDL under restricted compositionality.*

In addition, given any subsumption formula ψ, if $\mathfrak{T} \models_\epsilon \psi$ under the semantics with restricted compositionality, then $\mathfrak{T} \models_\epsilon \psi$ also holds in DDL under compositional consistency.

Proof. The theorem follows from the fact, that given any DDL knowledge base $\mathfrak{T}$, the set of all models of $\mathfrak{T}$ under compositional consistency is a subset of its set of models under restricted compositionality, which in turn is a subset of its set of models under the original semantics. □

As we will see, even if the semantics is relaxed when compared to full compositional consistency, the amount of subsumption propagation is the same. That is, subsumption propagates along both: chains of into- and chains of onto-bridge rules, as we formally state in the theorem.

Theorem 83. *Given a distributed TBox $\mathfrak{T}$, as illustrated in Fig. 5.6, with an index set I and a set of bridge rules $\mathfrak{B}$, that features $n+1$ local TBoxes $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n$ with concepts $E, F \in \mathcal{T}_0$, and $C_i, D_i \in \mathcal{T}_i$, for $1 \leq i \leq n$, and k with $1 \leq k \leq n$ such that:*

1. $\mathfrak{T} \models_\epsilon i : C_i \sqsubseteq D_i$, *for* $1 \leq i \leq n$,
2. $i+1 : C_{i+1} \xrightarrow{\sqsupseteq} i : D_i \in \mathfrak{B}$, *for* $1 \leq i < k$,
3. $i : D_i \xrightarrow{\sqsubseteq} i+1 : C_{i+1} \in \mathfrak{B}$, *for* $k \leq i < n$,
4. $1 : C_1 \xrightarrow{\sqsupseteq} 0 : E \in \mathfrak{B}$ *and* $n : D_n \xrightarrow{\sqsubseteq} 0 : F \in \mathfrak{B}$.

In DDL with restricted compositionality it follows that $\mathfrak{T} \models_\epsilon 0 : E \sqsubseteq F$.

Proof. In fact, we have already proven the theorem when we proved Theorem 80. This is easily verified by inspection of the proof of Theorem 80. Compositional consistency is always applied in this proof in such cases, in which it is justified also by the weaker semantics with restricted compositionality. Since the propagation pattern is the same in both cases, the proof of Theorem 80, without any change effectively proves also the current theorem. □

Of course, we are interested in the desired properties, that have been postulated for DDL and proven to hold for the original semantics (Borgida & Serafini 2003, Serafini et al. 2005). It is quite trivial to prove monotonicity. The proof is exactly the same as for Theorem 81 and we will not repeat it again.

Theorem 84. *The monotonicity property is satisfied in DDL under the semantics with restricted compositional consistency.*

The greatest advantage of the semantics with restricted compositionality is the fact that it satisfies the directionality property, which is not satisfied by the stronger semantics, as we have learned before.

Theorem 85. *The directionality property is satisfied in DDL under restricted compositionality.*

Proof. Given a distributed TBox $\mathfrak{T}$ we shall prove that if there is no directed path from i to j in $\mathrm{G}_{\mathfrak{T}}$, then

$$\mathfrak{T} \models_\epsilon j : C \sqsubseteq D \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D \ .$$

Recall, that $\mathfrak{T} \setminus\setminus \{i\}$ is obtained from $\mathfrak{T}$ by removing $\mathcal{T}_i$, $\mathfrak{B}_{ik}$ and $\mathfrak{B}_{li}$, for each $k, l \in I$, $k, l \neq i$.

If part. By contradiction. Assume that $\mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D$ and $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$. If $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$, then there is a distributed model $\mathfrak{I}$ of $\mathfrak{T}$ in which $C^{\mathcal{I}_j} \not\subseteq D^{\mathcal{I}_j}$. Let $\mathfrak{I}^{-i}$ be obtained from $\mathfrak{I}$ by removing $\mathcal{I}_i$ and r_{ik} and r_{ki} for all $k \neq i$. Since $\mathfrak{I}$ is a model of $\mathfrak{T}$, $\mathfrak{I}^{-i}$ is a model of $\mathfrak{T} \setminus\setminus \{i\}$ (this follows directly from the construction). However $C^{\mathcal{I}_j^{-i}} = C^{\mathcal{I}_j}$ and $D^{\mathcal{I}_j^{-i}} = D^{\mathcal{I}_j}$ and hence $C^{\mathcal{I}_j^{-i}} \not\subseteq D^{\mathcal{I}_j^{-i}}$. This implies that $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$, but we have assumed the contrary.

Only-if part. By contradiction. Assume that $\mathfrak{T} \models_\epsilon j : C \sqsubseteq D$ and $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$. If $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$, then there is a distributed model $\mathfrak{I}^{-i}$ of $\mathfrak{T} \setminus\setminus \{i\}$ in which $C^{\mathcal{I}_j^{-i}} \not\subseteq D^{\mathcal{I}_j^{-i}}$. Let $J \subset I$ be the set of nodes reachable from i by some directed path in $\mathrm{G}_{\mathfrak{T}}$. Let $\mathfrak{I}$ be the following distributed interpretation of $\mathfrak{T}$:

1. $\mathcal{I}_m = \mathcal{I}_m^{-i}$ for each $m \in I \setminus J$;
2. $\mathcal{I}_m = \mathcal{I}^\epsilon$ for each $m \in J$;
3. $r_{mn} = r_{mn}^{-i}$ if $m, n \in I \setminus J$;
4. $r_{mn} = \emptyset$ if either $m \in J$ or $n \in J$.

We now show that $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$. In order to demonstrate this, we need to show that local axioms and bridge rules of $\mathfrak{T}$ are satisfied by $\mathfrak{I}$, and that the domain relation of $\mathfrak{I}$ satisfies the restricted compositionality condition. One by one, this is shown as follows:

1. local axioms: for any $m \in I$, each local GCI axiom $\phi = m : G \sqsubseteq H$ must be satisfied by $\mathcal{I}_m$. If $m \in J$ then $\mathcal{I}_m$ is a hole, which satisfies any local GCI axiom, and hence it also satisfies ϕ. If $m \in I \setminus J$ then $\mathcal{I}_m = \mathcal{I}_m^{-i}$, which satisfies any $\phi \in \mathcal{T}_m$ by the construction;

2. bridge rules: given $m, n \in I$ and a bridge rule b that maps from $m : G$ to $n : H$, we must show that b is satisfied by $\mathfrak{I}$. We distinguish between several cases. First, if $m, n \notin J$, then b is satisfied by $\mathfrak{I}$ because $r_{mn}(G^{\mathcal{I}_m}) = r_{mn}^{-i}(G^{\mathcal{I}_m^{-i}})$, $H^{\mathcal{I}_n} = H^{\mathcal{I}_n^{-i}}$, and b is satisfied by $\mathfrak{I}^{-i}$. The case when $m \in J$ and $n \in I \setminus J$ is impossible, because there is a bridge rule from m to n and hence if m belongs to J, n must also belong to J. In all remaining cases $n \in J$. In this case it follows that $r_{mn}(G^{\mathcal{I}_m}) = \emptyset$ and $H^{\mathcal{I}_n} = \emptyset$, because $r_{mn} = \emptyset$ and $\mathcal{I}_n$ is a hole. In such a case however $\mathfrak{I}$ satisfies ϕ;

3. restricted compositionality: Let $k, m, n \in I$ be three mutually distinct indices such that there exists a directed path from k to m in $\mathrm{G}_{\mathfrak{T}}$ and another one from m and n. We must now show that $r_{km} \circ r_{mn} = r_{kn}$. If none of k, m, n belongs to J, then the fact that $r_{km} = r_{km}^{-i}$, $r_{mn} = r_{mn}^{-i}$, and $r_{kn} = r_{kn}^{-i}$ implies that $r_{km} \circ r_{mn} = r_{kn}$. If $k \in J$, then the fact that $r_{km} = \emptyset$ and $r_{kn} = \emptyset$ implies that $r_{km} \circ r_{mn} = r_{kn}$. If $m \in J$ then necessarily also $n \in J$ since there is a directed path between these two nodes in $\mathrm{G}_{\mathfrak{T}}$. But $n \in J$ implies that $r_{mn} = r_{kn} = \emptyset$, thus implying $r_{km} \circ r_{mn} = r_{kn}$. Finally, if $n \in J$ then again $r_{mn} = r_{kn} = \emptyset$, and so $r_{km} \circ r_{mn} = r_{kn}$.

And so we have shown that $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$. However, since there is no directed path from i to j in $\mathrm{G}_{\mathfrak{T}}$, it follows that $j \notin J$ and hence $\mathcal{I}_j = \mathcal{I}_j^{-i}$ and hence $C^{\mathcal{I}_j} \not\subseteq D^{\mathcal{I}_j}$. This in turn implies that $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$, but we have assumed the contrary. □

We have just shown that this restricted form of compositional consistency does not break directionality. Unfortunately, we did not manage to eliminate all of the undesirable side effects, since, as the example bellow shows, the local inconsistency property is not satisfied.

Example 19. *Consider a distributed TBox $\mathfrak{T}$ with index set $I = \{1, 2, 3\}$ and a set of bridge rules $\mathfrak{B}$, such that $\mathcal{T}_1 = \{\top \sqsubseteq \bot\}$, $\mathcal{T}_2 = \emptyset$, $\mathcal{T}_3 = \emptyset$ which contains bridge rules:*

$$2 : E \xrightarrow{\sqsupseteq} 1 : F \ , \qquad 1 : G \xrightarrow{\sqsupseteq} 3 : H \ ,$$
$$2 : C \xrightarrow{\sqsupseteq} 3 : D \ .$$

Since $\mathcal{T}_1$ is inconsistent, the only admissible interpretation is a hole, and so due to compositional consistency, even in the restricted case, $r_{23} = \emptyset$. Hence $\mathfrak{T} \models_\epsilon 3 : D \sqsubseteq \bot$. If local inconsistency would hold, then for each $J \subseteq I$, $3 \notin J$ we must have $\mathfrak{T}(\epsilon_J) \models_{\mathrm{d}} 3 : D \sqsubseteq \bot$. Similarly as in Example 17, for $J = \{1\}$ this does not hold, as $\mathfrak{T}(\epsilon_J)$ admits a d-model $\mathfrak{I}$ with $\Delta^{\mathcal{I}_2} = C^{\mathcal{I}_2} = \{x\}$, $\Delta^{\mathcal{I}_3} = D^{\mathcal{I}_3} = \{y\}$, $r_{23} = \{\langle x, y\rangle\}$ and $r_{32} = \emptyset$, for which $\mathfrak{I} \not\models_{\mathrm{d}} 3 : D \sqsubseteq \bot$ since $D^{\mathcal{I}_3} \neq \emptyset$.

And so, we have learned, that the more cautious approach, in which we apply compositional consistency only if two local ontologies are actually connected by

bridge rules, provides us with the same amount of subsumption propagation as before, when we have applied compositional consistency universally. We have learned, that also directionality is satisfied, however, as we have just seen, the local inconsistency property is violated.

This leaves us with a semantics, which is better than the previous one, however, we are not fully satisfied, since the local inconsistency property is also a very important desideratum for DDL. We cannot just forget about this condition, because in the heterogeneous and dynamic application scenarios that we have envisaged for DDL, such as for instance its application on the Semantic Web, occasional local inconsistency is inevitable. A viable representation formalism ought to provide reasonable behaviour also in such cases. Hence, we investigate also the other approach, in which the restrictions placed on the domain relation are further weakened.

5.2.3 DDL with Transitive Domain Relations

The other approach that we have outlined above suggests falling back from the compositional consistency restriction, and to solely require transitivity of the domain relation in distributed models. That is, we will drop condition (5.1) completely but we will still require condition (5.2). We will see, that subsumption propagation is more limited under this weaker condition, but on the other hand this semantics satisfies all of the desiderata previously postulated for DDL including the directionality and the local inconsistency properties. In addition, we will be able to introduce a distributed tableaux reasoning algorithm that decides satisfiability for this semantics later in Sect. 5.3. Let us start by introducing transitive domain relations formally.

Definition 78 (Transitivity)**.** *Given a distributed interpretation $\mathfrak{I}$ with domain relation r, we say that r (and also $\mathfrak{I}$) satisfies the transitivity requirement, if for any three mutually distinct indices $i, j, k \in I$ we have $r_{ij} \circ r_{jk} \subseteq r_{ik}$.*

Apparently, a distributed interpretation satisfies the transitivity condition if and only if its domain relation is transitive. The semantics obtained by allowing only distributed interpretations that satisfy the transitivity condition is called DDL under transitivity or DDL with transitive domain relations, and it is formally established as follows.

Definition 79 (Distributed model under transitivity)**.** *Given a DDL knowledge base $\mathfrak{T}$ over an index set I. A distributed interpretation $\mathfrak{I}$ is a model of $\mathfrak{T}$ under the DDL semantics with transitivity, if it is a distributed model of $\mathfrak{T}$ in the sense of Definition 59 and if its domain relation satisfies the transitivity requirement.*

Again, it is easy to guess, that this new semantics is also a strengthening of the original DDL semantics. Since this semantics constitutes a further relaxation from the semantics with restricted compositionality, on the scale of strength it is placed between the original semantics of DDL, which is properly weaker, and the semantics with restricted compositionality, which is properly stronger. The semantics with full compositional consistency is further properly stronger than these three, as we have already learned.

Theorem 86. *Given a distributed TBox $\mathfrak{T}$ and a subsumption formula ϕ, if $\mathfrak{T} \models_\epsilon \phi$ according to the original semantics, then $\mathfrak{T} \models_\epsilon \phi$ also holds in DDL under transitivity.*

In addition, given any subsumption formula ψ, if $\mathfrak{T} \models_\epsilon \psi$ under the semantics with transitivity, then $\mathfrak{T} \models_\epsilon \psi$ also holds in DDL under restricted compositionality.

Proof. Again, it is easy to observe, that given any DDL knowledge base $\mathfrak{T}$, the set of all models of $\mathfrak{T}$ under restricted compositionality is a subset of its set of models under transitivity, which in turn is a subset of the set of models of $\mathfrak{T}$ under the original semantics. This implies the proposition of the theorem. □

Unfortunately, under the semantics with transitivity, which, as we have just learned, is the weakest of the three strengthenings investigated in this chapter, the amount of subsumption propagation is not at the same level as under the two semantics that we have studied above. This is apparent from the following example.

Example 20. *Consider a distributed TBox $\mathfrak{T}$ with three local TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$, with local concepts C_1, D_1 in $\mathcal{T}_1$, E_2 in $\mathcal{T}_2$ and F_3 in $\mathcal{T}_3$, and with bridge rules:*

$$2 : E_2 \xrightarrow{\sqsubseteq} 3 : F_3 \ , \qquad 3 : F_3 \xrightarrow{\sqsubseteq} 1 : D_1 \ ,$$
$$2 : E_2 \xrightarrow{\sqsupseteq} 1 : C_1 \ .$$

This distributed TBox is clearly a very simple instance of the setting assumed by Theorem 80. According to this theorem, we should be able to derive $\mathfrak{T} \models_{\mathrm{d}} 1 : C_1 \sqsubseteq D_1$. This is not true, however, as a distributed model of $\mathfrak{T}$ exists in which ${C_1}^{\mathcal{I}_1} \not\subseteq {D_1}^{\mathcal{I}_1}$. This distributed model $\mathfrak{I}$ has three local domains $\Delta^{\mathcal{I}_1} = \{c_1, d_1\}$, $\Delta^{\mathcal{I}_2} = \{e_2\}$, and $\Delta^{\mathcal{I}_3} = \{f_3\}$. Local concepts are interpreted as follows: ${C_1}^{\mathcal{I}_1} = \{c_1, d_1\}$, ${D_1}^{\mathcal{I}_1} = \{d_1\}$, ${E_2}^{\mathcal{I}_2} = \{e_2\}$, and ${F_3}^{\mathcal{I}_3} = \{f_3\}$. Finally, the domain relation is the following: $r_{21} = \{\langle e_2, c_1\rangle, \langle e_2, d_1\rangle\}$, $r_{23} = \{\langle e_2, f_3\rangle\}$, $r_{31} = \{\langle f_3, d_1\rangle\}$, $r_{12} = r_{32} = r_{13} = \emptyset$. It is easily verified that r is transitive, hence the transitivity condition is satisfied, and hence Theorem 80 does not stand if compositional consistency is relaxed to transitivity. At the same time, this example does not invalidate Theorem 80 under compositional consistency, as the stronger condition is not satisfied by $\mathfrak{I}$ because $r_{21}(e_2) = \{c_1, d_1\} \not\subseteq \{d_1\} = r_{31}(r_{23}(e_2))$.

From the example we see, that DDL with transitive domain relation have problems with chains of into-bridge rules. Luckily enough the problem does not affect chains of onto-bridge rules – if we analyse Example 15 we will see that transitivity is enough to guarantee the two chaining onto-bridge rules in this example to propagate the subsumption relationship as expected. This observation holds in general and is formally established by Theorem 87 below. This theorem is a weaker version of Theorem 80 and it provides a characterization of the semantics of DDL under the transitivity condition.

Theorem 87. *Given a distributed TBox, as illustrated in Fig. 5.8, with an index set I and a set of bridge rules $\mathfrak{B}$, that features $k+1$ local TBoxes $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_k$ with concepts $E, F \in \mathcal{T}_0$, and $C_i, D_i \in \mathcal{T}_i$, for $1 \leq i \leq k$, such that:*

1. *$\mathfrak{T} \models_\epsilon i : C_i \sqsubseteq D_i$, for $1 \leq i \leq k$,*
2. *$i+1 : C_{i+1} \xrightarrow{\sqsupseteq} i : D_i \in \mathfrak{B}$, for $1 \leq i < k$,*

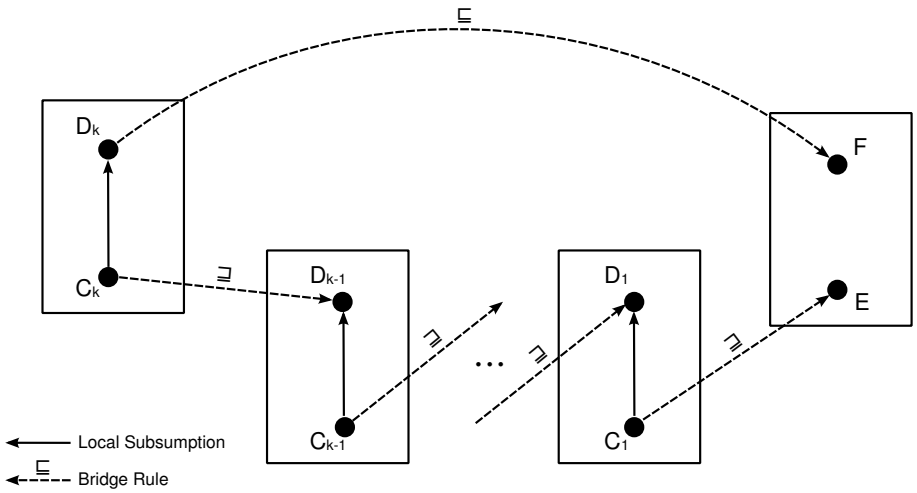

Figure 5.8: Distributed TBox from Theorem 87, representing the subsumption propagation pattern occurring under the DDL semantics with transitivity

3. $1 : C_1 \xrightarrow{\sqsupseteq} 0 : E \in \mathfrak{B}$ *and* $k : D_k \xrightarrow{\sqsubseteq} 0 : F \in \mathfrak{B}$.

In DDL under the transitivity condition it follows that $\mathfrak{T} \models_\epsilon 0 : E \sqsubseteq F$.

Proof. The theorem is proven by the following chain of inclusions:

$$E^{\mathcal{I}_0} \overset{1}{\subseteq} r_{k0}\big({C_k}^{\mathcal{I}_k}\big) \overset{2}{\subseteq} r_{k0}\big({D_k}^{\mathcal{I}_k}\big) \overset{3}{\subseteq} F^{\mathcal{I}_0} \quad .$$

Inclusion 3 is a direct consequence of the into-bridge rule $k : D_k \xrightarrow{\sqsubseteq} 0 : F \in \mathfrak{B}$. As for Inclusion 2, $\mathfrak{T} \models_{\mathrm{d}} k : C_k \sqsubseteq D_k$ implies ${C_k}^{\mathcal{I}_k} \subseteq {D_k}^{\mathcal{I}_k}$ and since $r_{k0}(\cdot)$ we also have $r_{k0}\big({C_k}^{\mathcal{I}_k}\big) \subseteq r_{k0}\big({D_k}^{\mathcal{I}_k}\big)$. Inclusion 1 divides into:

$$E^{\mathcal{I}_0} \subseteq r_{10}\big(r_{21}\big(\cdots r_{kk-1}\big({C_k}^{\mathcal{I}_k}\big)\cdots\big)\big) \subseteq r_{k0}\big({C_k}^{\mathcal{I}_k}\big) \quad .$$

Both these inclusions ought to be proven by mathematical induction. The first inclusion is proven by induction on k. In the base case, we ought to prove $E^{\mathcal{I}_0} \subseteq r_{10}\big({C_1}^{\mathcal{I}_1}\big)$. This is a direct consequence of the onto-bridge rule $1 : C_1 \xrightarrow{\sqsupseteq} 0 : E \in \mathfrak{B}$.

Now we move on to the induction step. From the induction hypothesis we have $E^{\mathcal{I}_0} \subseteq r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big({C_{k-1}}^{\mathcal{I}_{k-1}}\big)\cdots\big)\big)$. From the assumptions of the theorem we derive:

$${C_{k-1}}^{\mathcal{I}_{k-1}} \subseteq {D_{k-1}}^{\mathcal{I}_{k-1}} \subseteq r_{kk-1}\big({C_k}^{\mathcal{I}_k}\big) \quad .$$

The composition of mappings $r_{10}(r_{21}(\cdots r_{k-1k-2}(\cdot)\cdots))$ is a mapping, and so $r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big({C_{k-1}}^{\mathcal{I}_{k-1}}\big)\cdots\big)\big) \subseteq r_{10}\big(r_{21}\big(\cdots r_{k-1k-2}\big(r_{kk-1}\big({C_k}^{\mathcal{I}_k}\big)\big)\cdots\big)\big)$. Together with the induction hypothesis this amounts to the inclusion we wanted to prove.

As for the second inclusion, we shall prove a slightly more general proposition:

$$r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{kk-1}\big({C_k}^{\mathcal{I}_k}\big)\cdots\big)\big) \subseteq r_{kn}\big({C_k}^{\mathcal{I}_k}\big) \quad ,$$

where $0 \leq n \leq k-2$. The inclusion we ought to prove is then derived by setting $n = 0$. The proposition is proven by mathematical induction on $k - n$. As for the base case consider $k-n = 2$, and so $n = k-2$. That means we have to show $r_{k-1k-2}\big(r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\big) \subseteq r_{kk-2}\big(C_k{}^{\mathcal{I}_k}\big)$, which holds because $\mathfrak{I}$ satisfies transitivity. The induction step is for $k-n = j > 2$, and so $n = k-j$. From the induction hypothesis we get $r_{n+2n+1}\big(r_{n+3n+2}\big(\cdots r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\cdots\big)\big) \subseteq r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big)$. And since $r_{n+1n}(\cdot)$ is a mapping, we also get $r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\cdots\big)\big) \subseteq r_{n+1n}\big(r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big)\big)$. Once again, since $\mathfrak{I}$ satisfies the transitivity condition we have $r_{n+1n}\big(r_{kn+1}\big(C_k{}^{\mathcal{I}_k}\big)\big) \subseteq r_{kn}\big(C_k{}^{\mathcal{I}_k}\big)$. By combining the last two inclusions we directly get:

$$r_{n+1n}\big(r_{n+2n+1}\big(\cdots r_{kk-1}\big(C_k{}^{\mathcal{I}_k}\big)\cdots\big)\big) \subseteq r_{kn}\big(C_k{}^{\mathcal{I}_k}\big) \quad .$$

By setting $n = 0$ we derive the inclusion we wanted to prove, and hence the theorem. □

And so we see, that this weaker semantics only partially solves the problem of lacking subsumption propagation between remote ontologies, as we have identified it at the beginning of Sect. 5.1. Subsumption propagates along chains of onto-bridge rules, but it fails to propagate along chains of into-bridge rules. On the other hand, compared to the stronger semantics, it satisfies the requirements that have been placed on DDL (Borgida & Serafini 2003, Serafini et al. 2005), as we formally demonstrate below.

Theorem 88. *The monotonicity property is satisfied in DDL under transitivity.*

While monotonicity is a rather trivial property that expectedly holds also for this semantics, recall that other important properties, such as directionality and local inconsistency, were problematic for the stronger semantics that we have investigated. Fortunately, the semantics of DDL under transitivity behaves more reasonably. Let us first show that directionality is satisfied.

Theorem 89. *The directionality property is satisfied in DDL under transitivity.*

Proof. Given a distributed TBox $\mathfrak{T}$, we shall prove that if there is no directed path from i to j in $\mathrm{G}_{\mathfrak{T}}$, then

$$\mathfrak{T} \models_\epsilon j : C \sqsubseteq D \iff \mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D \quad .$$

Recall, that $\mathfrak{T} \setminus\setminus \{i\}$ is obtained from $\mathfrak{T}$ by removing $\mathcal{T}_i$, $\mathfrak{B}_{ik}$ and $\mathfrak{B}_{li}$, for each $k, l \in I$, $k, l \neq i$.

If part. By contradiction. Assume that $\mathfrak{T} \setminus\setminus \{i\} \models_\epsilon j : C \sqsubseteq D$ and $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$. If $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$ then there is a distributed model $\mathfrak{I}$ of $\mathfrak{T}$ in which $C^{\mathcal{I}_j} \not\subseteq D^{\mathcal{I}_j}$. Let $\mathfrak{I}^{-i}$ be obtained from $\mathfrak{I}$ by removing $\mathcal{I}_i$ and r_{ik} and r_{ki} for all $k \neq i$. Since $\mathfrak{I}$ is a model of $\mathfrak{T}$, $\mathfrak{I}^{-i}$ is a model of $\mathfrak{T} \setminus\setminus \{i\}$ (this follows directly from the construction). However, $C^{\mathcal{I}_j^{-i}} = C^{\mathcal{I}_j}$ and $D^{\mathcal{I}_j^{-i}} = D^{\mathcal{I}_j}$ and hence $C^{\mathcal{I}_j^{-i}} \not\subseteq D^{\mathcal{I}_j^{-i}}$. This implies that $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$, but we have assumed the contrary.

Only-if part. By contradiction. Assume that $\mathfrak{T} \models_\epsilon j : C \sqsubseteq D$ and $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$. If $\mathfrak{T} \setminus\setminus \{i\} \not\models_\epsilon j : C \sqsubseteq D$, then there is a distributed model $\mathfrak{I}^{-i}$ of $\mathfrak{T} \setminus\setminus \{i\}$ in which $C^{\mathcal{I}_j^{-i}} \not\subseteq D^{\mathcal{I}_j^{-i}}$. Let $J \subset I$ be the set of nodes reachable from i by some directed path in $\mathrm{G}_{\mathfrak{T}}$. Let $\mathfrak{I}$ be the following distributed interpretation of $\mathfrak{T}$:

1. $\mathcal{I}_m = \mathcal{I}_m^{-i}$ for each $m \in I \setminus J$;
2. $\mathcal{I}_m = \mathcal{I}^\epsilon$ for each $m \in J$;
3. $r_{mn} = r_{mn}^{-i}$ if $m, n \in I \setminus J$;
4. $r_{mn} = \emptyset$ if either $m \in J$ or $n \in J$.

We now show that $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$. In order to demonstrate this, we need to show that local axioms and bridge rules of $\mathfrak{T}$ are satisfied by $\mathfrak{I}$, and that the domain relation of $\mathfrak{I}$ is transitive. One by one, this is shown as follows:

1. local axioms: for any $m \in I$, each local GCI axiom $\phi = m : G \sqsubseteq H$ must be satisfied by $\mathcal{I}_m$. If $m \in J$ then $\mathcal{I}_m$ is a hole, which satisfies any local GCI axiom, and hence it also satisfies ϕ. If $m \in I \setminus J$ then $\mathcal{I}_m = \mathcal{I}_m^{-i}$, which satisfies any ϕ of $\mathcal{T}_m$, because of the construction;

2. bridge rules: given $m, n \in I$ and a bridge rule b that maps from $m : G$ to $n : H$, we must show that b is satisfied by $\mathfrak{I}$. We distinguish between several cases. First, if $m, n \notin J$, then b is satisfied by $\mathfrak{I}$ because $r_{mn}\big(G^{\mathcal{I}_m}\big) = r_{mn}^{-i}(G^{\mathcal{I}_m^{-i}})$, $H^{\mathcal{I}_n} = H^{\mathcal{I}_n^{-i}}$, and b is satisfied by $\mathfrak{I}^{-i}$. The case when $m \in J$ and $n \in I \setminus J$ is impossible, because there is a bridge rule from m to n and hence if m belongs to J, n must also belong to J. In all remaining cases $n \in J$. In this case it follows that $r_{mn}\big(G^{\mathcal{I}_m}\big) = \emptyset$ and $H^{\mathcal{I}_n} = \emptyset$, because $r_{mn} = \emptyset$ and $\mathcal{I}_n$ is a hole. In such a case however $\mathfrak{I}$ satisfies ϕ.

3. transitivity: We have to show for any three mutually distinct indices $k, m, n \in I$ that $r_{km} \circ r_{mn} \subseteq r_{kn}$. If none of k, m, n belongs to J, then the fact that $r_{km} = r_{km}^{-i}$, $r_{mn} = r_{mn}^{-i}$, and $r_{kn} = r_{kn}^{-i}$ implies $r_{km} \circ r_{mn} \subseteq r_{kn}$. If at least one of k, m and n belongs to J, then either r_{km} or r_{mn} is equal to $\emptyset$, which implies that $r_{km} \circ r_{mn} = \emptyset \subseteq r_{kn}$.

And so we have shown that $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$. However, since there is no directed path from i to j in $\mathrm{G}_{\mathfrak{T}}$, it follows that $j \notin J$ and hence $\mathcal{I}_j = \mathcal{I}_j^{-i}$ and hence $C^{\mathcal{I}_j} \not\subseteq D^{\mathcal{I}_j}$. This in turn implies that $\mathfrak{T} \not\models_\epsilon j : C \sqsubseteq D$, but we have assumed the contrary. □

Also local inconsistency property is satisfied by the semantics with transitivity. It turns out, this property is violated, whenever we have tried to apply the condition (5.1), even in the case when we have applied it in a cautious manner. When this condition is dropped, the problem no longer occurs. As we have learned above, there is some trade-off, however, as the amount of subsumption propagation decreases. The question, whether the condition (5.1) can be applied in such a way, that none of the desiderata is violated, is left open for future research.

Theorem 90. *The local inconsistency property is satisfied in DDL under transitivity.*

Proof. Given a distributed TBox $\mathfrak{T}$ over an index set I, and given some $i \in I$, and a subsumption formula $\phi = i : C \sqsubseteq D$, we ought to prove the equivalence:

$$\mathfrak{T} \models_\epsilon \phi \iff ((\forall J)\ J \subseteq I \wedge i \notin J \implies \mathfrak{T}(\epsilon_J) \models_{\mathrm{d}} \phi) \quad .$$

If part. Suppose by contradiction that for all subsets J of I such that $i \notin J$, $\mathfrak{T}(\epsilon_J) \models_{\mathrm{d}} \phi$ and $\mathfrak{T} \not\models_\epsilon \phi$. In that case there is an ϵ-model $\mathfrak{I}^1$ of $\mathfrak{T}$ in which $C^{\mathcal{I}_i^1} \not\subseteq D^{\mathcal{I}_i^1}$. Since this is an ϵ-model, some of its local interpretations are possibly holes. Let J be the set of all indices in I that hold a hole as local interpretation, i.e., $J = \{j \in I \mid \mathcal{I}_j^1 = \mathcal{I}^\epsilon\}$. As $\mathfrak{I}^1 \not\models_\epsilon \phi$ we have $i \notin J$, because otherwise $\mathcal{I}_i = \mathcal{I}^\epsilon$ and a hole entails any formula including ϕ. Let $\mathfrak{I}^2$ be the distributed interpretation obtained by cutting the holes off from $\mathfrak{I}^1$, i.e., $\mathfrak{I}^2$ has $I \setminus J$ as its index set, $\mathcal{I}_j^2 = \mathcal{I}_j^1$ for each $j \in I \setminus J$ and $r_{jk}^2 = r_{jk}^1$ for any $j, k \in I \setminus J$. We will now show that $\mathfrak{I}^2$ is a d-model of $\mathfrak{T}(\epsilon_J)$:

1. $\mathfrak{I}^2$ is a d-interpretation, because it does not contain holes any more;

2. given any $j \in I \setminus J$ and any j-local GCI ψ, either ψ also belongs to $\mathcal{T}_j$ of $\mathfrak{T}$ and hence it is satisfied by $\mathcal{I}_j^2 = \mathcal{I}_j^1$ since $\mathfrak{I}^1$ satisfies $\mathcal{T}_j$, or by construction of $\mathfrak{T}(\epsilon_J)$ $\psi = j : D \sqsubseteq \bot$ and there is some bridge rule $b = k : C \xrightarrow{\exists} j : D$ in $\mathfrak{T}$ with $k \in J$. Since b is satisfied by $\mathfrak{I}^1$ and $\mathcal{I}_k^1 = \mathcal{I}^\epsilon$ we have $D^{\mathcal{I}_j^2} = D^{\mathcal{I}_j^1} = \emptyset$ and hence ψ is satisfied;

3. each bridge rule of $\mathfrak{T}(\epsilon_J)$, say b between $\mathcal{T}_j$ and $\mathcal{T}_k$, $j, k \in I \setminus J$, is satisfied by $\mathfrak{I}^2$ since b also appears in $\mathfrak{T}$ and $\mathcal{I}_j^2 = \mathcal{I}_j^1$, $\mathcal{I}_k^2 = \mathcal{I}_k^1$, $r_{jk}^2 = r_{jk}^1$ and $\mathfrak{I}^1$ satisfies b;

4. finally, the interpretation $\mathfrak{I}^2$ satisfies transitivity because $\mathfrak{I}^1$ does and $r_{jk}^2 = r_{jk}^1$ for any $j, k \in I \setminus J$.

Hence we have found a contradiction because $\mathfrak{T}(\epsilon_J)$ ought to entail ϕ but it does not, since $\mathfrak{I}^2$ is a model of $\mathfrak{T}(\epsilon_J)$ in which $C^{\mathcal{I}_i^2} \not\subseteq D^{\mathcal{I}_i^2}$ (because $C^{\mathcal{I}_i^1} \not\subseteq D^{\mathcal{I}_i^1}$, $C^{\mathcal{I}_i^2} = C^{\mathcal{I}_i^1}$ and $D^{\mathcal{I}_i^2} = D^{\mathcal{I}_i^1}$).

Only-if part. Suppose by contradiction that $\mathfrak{T} \models_\epsilon \phi$ and there is $J \subseteq I$, not containing i, such that $\mathfrak{T}(\epsilon_J) \not\models_{\mathrm{d}} \phi$. In this case there must be a d-model of $\mathfrak{T}(\epsilon_J)$, say $\mathfrak{I}^1$ such that $C^{\mathcal{I}_i^1} \not\subseteq D^{\mathcal{I}_i^1}$. Let us construct a distributed interpretation $\mathfrak{I}^2$, starting from $\mathfrak{I}^1$ and adding a hole into each slot $j \in J$, i.e., $\mathcal{I}_j^2 = \mathcal{I}_j^1$ for each $j \in I \setminus J$ and $\mathcal{I}_j^2 = \mathcal{I}^\epsilon$ for each $j \in J$. Moreover, for any pair of indices $j, k \in I$ we set $r_{jk}^2 = r_{jk}^1$ if both $i, j \in I \setminus J$ and $r_{jk}^2 = \emptyset$ otherwise. We now prove that $\mathfrak{I}^2$ is an ϵ-model of $\mathfrak{T}$.

1. obviously $\mathfrak{I}^2$ is an ϵ-interpretation of $\mathfrak{T}$;

2. each local GCI ψ in $\mathfrak{T}$ is satisfied, as if $\psi \in \mathcal{T}_j$, $j \in I \setminus J$ then ψ also appears in $\mathfrak{T}(\epsilon_J)$ and since $\mathcal{I}_j^2 = \mathcal{I}_j^1$ we have that $\mathcal{I}_j^2$ entails ψ. If $\psi \in \mathcal{T}_k$, $k \in J$ then ψ is satisfied since $\mathcal{I}_k^2$ is a hole;

3. let b be a bridge rule directed from $\mathcal{T}_j$ and $\mathcal{T}_k$, $j, k \in I$ that maps between $j : C$ and $k : D$. If both $j, k \in I \setminus J$ then b is satisfied because it is satisfied by $\mathfrak{I}^1$ and we have $\mathcal{I}_j^2 = \mathcal{I}_j^1$, $\mathcal{I}_k^2 = \mathcal{I}_k^1$, $r_{jk}^2 = r_{jk}^1$. If both $j, k \in J$ then b is satisfied because $\mathcal{I}_j^2 = \mathcal{I}^\epsilon$, $\mathcal{I}_k^2 = \mathcal{I}^\epsilon$ and $r_{jk}^2 = \emptyset$. If $j \in I \setminus J$ and $k \in J$ then b is satisfied because $\mathcal{I}_k^2 = \mathcal{I}^\epsilon$ and $r_{jk}^2 = \emptyset$. If $j \in J$ and $k \in I \setminus J$ and $b = j : C \xrightarrow{\sqsubseteq} k : D$ then b is satisfied because $r_{jk}^2(C^{\mathcal{I}_j^2}) = \emptyset$ which surely is a subset of $D^{\mathcal{I}_k^2}$. The last case occurs when $j \in J$, $k \in I \setminus J$ and $b = j : C \xrightarrow{\exists} k : D$. In this case b is also satisfied because by construction

the local GCI $k : D \sqsubseteq \bot$ belongs to $\mathfrak{T}(\epsilon_J)$ and hence $D^{\mathcal{I}_k^2} = D^{\mathcal{I}_k^1} = \emptyset$ which surely is a subset of $r_{jk}^2(C^{\mathcal{I}_j^2})$;

4. finally, we show that the domain relation r^2 is transitive, that is, for any three mutually distinct $k, m, n \in I$ it holds that $r_{km}^2 \circ r_{mn}^2 \subseteq r_{kn}^2$. If none of k, m, n belongs to J then this holds because $r_{km}^2 = r_{km}^1$, $r_{mn}^2 = r_{mn}^1$, $r_{kn}^2 = r_{kn}^1$, and r^1 is transitive. Thanks to the construction of $\mathfrak{I}^2$ however, in any other case either $r_{km}^2 = \emptyset$ (if $k \in J$ or $m \in J$) or $r_{mn}^2 = \emptyset$ (if $m \in J$ or $n \in J$). Hence in all these cases transitivity trivially holds, since $r_{km}^2 \circ r_{mn}^2 = \emptyset \subseteq r_{kn}^2$.

And so we have just shown that $\mathfrak{I}^2$ is an ϵ-model of $\mathfrak{T}$. Now we again reach a contradiction, as we have assumed that $\mathfrak{T}$ entails ϕ, but $C^{\mathcal{I}_i^2} \not\subseteq D^{\mathcal{I}_i^2}$ (because $C^{\mathcal{I}_i^1} \not\subseteq D^{\mathcal{I}_i^1}$, $C^{\mathcal{I}_i^2} = C^{\mathcal{I}_i^1}$ and $D^{\mathcal{I}_i^2} = D^{\mathcal{I}_i^1}$). □

And so we have shown, that in the semantics of DDL under transitivity the level of subsumption propagation between remote ontologies is limited to propagation along chaining onto-bridge rules in scenarios that instantiate Theorem 87. Propagation of subsumption along chaining into-bridge rules is not granted, but all desiderata postulated for DDL including directionality and local inconsistency are satisfied.

Summing up, we have investigated and evaluated three possible alternative semantics for DDL, in order to deal with the problem of insufficient subsumption propagation between remote ontologies.

We have learned, that under the DDL semantics with full compositional consistency, subsumption propagation between remote ontologies is improved, but there are also undesirable consequences of such an approach: the directionality and the local inconsistency properties are violated. In order to retain the desired propertied, we have investigated two other approaches.

We have first tried to apply compositional consistency more cautiously, only if the local ontologies that are involved are connected by some directed path of bridge rules. This approach provides just the same level of subsumption propagation as the semantics under unrestricted compositionality and it also retains directionality, but it still violates the local inconsistency property.

The other approach amounts to falling back from compositional consistency to transitivity. We have learned, that under DDL with transitivity subsumption fails to propagate along chains of into-bridge rules, it does propagate to the satisfactory extent along chains of onto-bridge rules. So, the amount of subsumption propagation is less than in the previous two approaches, it is still greater compared to the original DDL semantics. On the other hand, all desired properties of DDL, including the directionality and local inconsistency properties are retained.

Therefore, we conclude that with respect to the current state of the art, the semantics of DDL with transitive domain relation provides sort of a compromise between subsumption propagation and other desired properties of DDL. In the next section we take a look at reasoning with this semantics.

5.3 Distributed Tableaux Algorithm for Transitive DDL

In this section, we introduce a distributed tableaux algorithm for deciding satisfiability of concepts with respect to an acyclic distributed TBox for DDL over $\mathcal{ALC}$ under the transitivity requirement. As much as in the original algorithm (Serafini et al. 2005), also in our approach, local reasoners are run independently. We keep precise track of the domain relation r however, and the communication between local reasoners is divided into multiple queries. Hence, while the original algorithm can be seen as working with autonomous local tableaux by passing queries, our algorithm can be seen as working with a truly distributed tableau.

Definition 80 (Distributed completion tree)**.** *Assume a distributed TBox* $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i\in I}, \mathfrak{B}\rangle$ *over* $\mathcal{ALC}$ *with index set* I*, concept names* $N_{\mathrm{C}} = \bigcup_{i\in I} N_{\mathrm{C}i}$ *and role names* $N_{\mathrm{R}} = \bigcup_{i\in I} N_{\mathrm{R}i}$*. Let* CC_i *be the set of all (atomic and complex) concepts over* $N_{\mathrm{C}i}$ *and* $N_{\mathrm{R}i}$ *in negation normal form. A* distributed completion tree[2] $T = \{T_i\}_{i\in I}$ *is a set of labeled trees* $T_i = \langle V_i, E_i, \mathcal{L}_i, r_i\rangle$*, such that for each* $i \in I$*:*

1. *the trees* $\{\langle V_i, E_i\rangle\}_{i\in I}$ *are mutually disjoint;*
2. *the labeling function* $\mathcal{L}_i$ *labels each node* $x \in V_i$ *with* $\mathcal{L}_i(x) \subseteq 2^{\mathrm{CC}_i}$ *and each edge* $\langle x, y\rangle \in E_i$ *with* $\mathcal{L}_i(\langle x, y\rangle) \in N_{\mathrm{R}i}$*;*
3. *the labeling function* r_i *labels each node* $x \in V_i$ *with a set of references to its r-images* $r_i(x) \subseteq \{j : y \mid j \in I \wedge y \in V_j\}$*.*

During the run of the tableaux algorithm, tableaux expansion rules are applied on the completion tree and the tree is expanded by each rule application. If no more rules are applicable any more, we say that the completion tree is *complete*. Besides for the completion tree to be complete, we need to detect whether the completion tree is sound. Recall that the notion of clash is used for this, the completion tree is sound if it is clash-free.

Definition 81 (Clash)**.** *There is a* clash *in the completion tree* T *if for some* $x \in V_i$ *and for some* $C \in N_{\mathrm{C}i}$ *we have* $\{C, \neg C\} \subseteq \mathcal{L}_i(x)$*. If there is no clash in* T *then we say that* T *is* clash-free*.*

In order to assure termination we use the standard subset blocking technique that is common for $\mathcal{ALC}$.

Definition 82 (Subset blocking)**.** *Given a distributed completion tree* $T = \{T_i\}_{i\in I}$*, a node* $x \in V_i$ *is blocked, if it has an ancestor* $y \in V_i$ *such that* $\mathcal{L}_i(x) \subseteq \mathcal{L}_i(y)$*. In such a case we also say that* x *is blocked by* y*.*

Similarly to the completion trees of classic DL, a node $y \in V_i$ is said to be an *R-successor* of $x \in V_i$, if $\langle x, y\rangle \in E_i$ and $\mathcal{L}_i(\langle x, y\rangle) = R$. We are now ready to present the tableaux algorithm for transitive DDL.

[2] We prefer the usual naming, and we use the name distributed completion *tree* even if technically it is no longer a tree but rather a forest, that is, a graph composed of several isolated independently rooted trees.

$\sqcap$-rule:
If $C_1 \sqcap C_2 \in \mathcal{L}_i(x)$ for some $x \in V_i$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}_i(x)$,
and x is not blocked,
then set $\mathcal{L}_i(x) = \mathcal{L}_i(x) \cup \{C_1, C_2\}$.

$\sqcup$-rule:
If $C_1 \sqcup C_2 \in \mathcal{L}_i(x)$ for some $x \in V_i$ and $\{C_1, C_2\} \cap \mathcal{L}_i(X) = \emptyset$,
and x is not blocked,
then either set $\mathcal{L}_i(x) = \mathcal{L}_i(x) \cup \{C_1\}$ or set $\mathcal{L}_i(x) = \mathcal{L}_i(x) \cup \{C_2\}$.

$\forall$-rule:
If $\forall R.C \in \mathcal{L}_i(x)$ for some $x \in V_i$, and there is R-successor y of x s.t. $C \notin \mathcal{L}_i(y)$,
and x is not blocked,
then set $\mathcal{L}_i(y) = \mathcal{L}_i(y) \cup \{C\}$.

$\exists$-rule:
If $\exists R.C \in \mathcal{L}_i(x)$, for some $x \in V_i$ with no R-successor y s.t. $C \in \mathcal{L}_i(y)$,
and x is not blocked,
then add new node z to V_i, add the edge $\langle x, z\rangle$ to E_i, and set $\mathcal{L}_i(z) = \{C\}$
and $\mathcal{L}_i(\langle x, z\rangle) = \{R\}$.

$\mathcal{T}$-rule:
If $C \sqsubseteq D \in \mathcal{T}$ and for some $x \in V_i$ $\mathrm{nnf}(\neg C \sqcup D) \notin \mathcal{L}_i(x)$, and x is not blocked,
then set $\mathcal{L}_i(x) = \mathcal{L}_i(x) \cup \{\mathrm{nnf}(\neg C \sqcup D)\}$.

$\xrightarrow{\sqsupseteq}$-rule:
If $G \in \mathcal{L}_j(x)$ for some $x \in V_j$, $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$,
and there is no $y \in V_i$ s.t. $C \in \mathcal{L}_i(y)$ and $j : x \in r_i(y)$,
and x is not blocked,
then add new node y to V_i and set $\mathcal{L}_i(y) = \{C\}$, and set $r_i(y) = \{j : x\} \cup r_j(x)$.

$\xrightarrow{\sqsubseteq}$-rule:
If $D \in \mathcal{L}_i(x)$ for some $x \in V_i$, $i : D \xrightarrow{\sqsubseteq} j : H \in \mathfrak{B}$
and there is $y \in V_j$ s.t. $j : y \in r_i(x)$ and $H \notin \mathcal{L}_j(y)$
then set $\mathcal{L}_j(y) = \mathcal{L}_j(y) \cup \{H\}$.

Figure 5.9: Tableaux expansion rules for DDL over $\mathcal{ALC}$ with transitivity

Algorithm 8 (Satisfiability in DDL with transitivity)**.** *Given a distributed TBox $\mathfrak{T}$, a concept C in NNF and $i \in I$ as inputs, the algorithm executes three steps:*

1. initialization: *create a new completion tree $T = \{T_j\}_{j \in I}$ such that $T_j = \langle \{s_0\}, \emptyset, \{s_0 \mapsto \{C\}\}, \emptyset\rangle$ for $j = i$ and $T_j = \langle \emptyset, \emptyset, \emptyset, \emptyset\rangle$ for $j \neq i$;*
2. tableau expansion: *apply the tableaux expansion rules of Fig. 5.9 exhaustively. This step is over when none of the rules is applicable;*
3. answer: *if the completion tree is clash-free, answer "C is satisfiable". Otherwise, answer "C is unsatisfiable".*

The first five of the tableaux rules, as seen in Fig. 5.9, are basically the $\mathcal{ALC}$ tableaux rules adjusted for distributed tableaux. In addition, the algorithm

uses two new rules. The $\xrightarrow{\sqsupseteq}$-rule is fired by onto-bridge rules. It is triggered in some node x of T_j and it generates a new node y in T_i, that is, in the part of the completion tree corresponding to the local model of $\mathcal{T}_i$. This rule is also responsible for domain relation tracking, and it also initializes the label $r_i(y)$. Note that the rule assures transitivity of the domain relation by adding also all the elements of $r_j(x)$ into $r_i(y)$.

The $\xrightarrow{\sqsubseteq}$ is fired by into-bridge rules, and it is responsible for propagating the subsumptions that have been computed in the remote part of the completion tree, say from the subtree T_i, back to the part from which the $\xrightarrow{\sqsupseteq}$-rule previously was fired, say T_j. Note that the domain relation tracking is important here, in order to determine precisely, into which node of T_j the concept, which is implied by the into-bridge rules which fired the $\xrightarrow{\sqsubseteq}$-rule, is to be added.

Below we present a formal correctness proof for the newly introduced algorithm. The proof is based on the classic proof for $\mathcal{ALC}$. The new thing to prove here is that the algorithm uses the $\xrightarrow{\sqsupseteq}$-rule and the $\xrightarrow{\sqsubseteq}$-rule to combine several autonomous local $\mathcal{ALC}$ reasoners correctly. The proof is done in three parts. We first prove termination: on every input the algorithm always terminates and never ends up in an infinite loop; then soundness: if the algorithm answers that C is satisfiable with respect to $\mathfrak{T}$ for some i-local concept C and a distributed TBox $\mathfrak{T}$, then there actually exists some model of $\mathfrak{T}$ that supports this; and finally we prove completeness: for every concept C that is satisfiable with respect to $\mathfrak{T}$, the algorithm indeed gives a correct answer.

Theorem 91. *Given a distributed TBox $\mathfrak{T}$ over $\mathcal{ALC}$ with acyclic bridge graph $\mathrm{G}_{\mathfrak{T}}$ and an i-local concept C on the input, the distributed tableaux algorithm for deciding satisfiability of concepts with respect to a distributed TBox over $\mathcal{ALC}$ for DDL with transitive domain relation always terminates and it is sound and complete.*

Proof. Termination. Given a distributed TBox $\mathfrak{T}$ with local TBox $\mathcal{T}_i$ and an i-local concept C, we ought to prove that the algorithm, once started with $\mathfrak{T}$, C and i on input, eventually terminates. The algorithm initializes the completion tree to $T = \{T_j\}_{j \in I}$ such that $T_j = \langle \{s_0\}, \emptyset, \{s_0 \mapsto \{C\}\}, \emptyset \rangle$ for $j = i$ and $T_j = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$ for $j \neq i$. The computation then continues by expanding T_i. If there are no onto-bridge rules ingoing into $\mathcal{T}_i$ the algorithm eventually terminates thanks to subset blocking, this result is known for $\mathcal{ALC}$. If there are some ingoing onto-bridge rules, then possibly, in some $x \in V_i$, such that $C \in \mathcal{L}_i(x)$ and C appears on the right hand side of an onto-bridge rule, say $k : D \xrightarrow{\sqsupseteq} i : C$, the $\xrightarrow{\sqsupseteq}$-rule is applied and computation is triggered in T_k. By structural induction we assume that this computation eventually terminates (the length of the longest incoming path of onto-bridge rules decreased for $\mathcal{T}_k$ compared to $\mathcal{T}_i$, and all such paths are finite because of acyclicity). During this process we possibly get some new concepts in $\mathcal{L}_i(x)$ that are introduced thanks to incoming into-bridge rules that trigger the $\xrightarrow{\sqsubseteq}$-rule – but only finitely many. The computation now continues in x and its descendants and possibly the $\xrightarrow{\sqsupseteq}$-rule is triggered again in some $y \in V_i$, a descendant of x. But thanks to subset blocking, this happens only finitely many times. Hence the algorithm eventually terminates.

Soundness. Now that we know that the algorithm terminates, we shall prove that if it answers "C is satisfiable" on input $\mathfrak{T}$, C and i, then it also holds that C is satisfiable in $\mathcal{T}_i$ with respect to $\mathfrak{T}$. That is, we must show that in such a case there exists a distributed model $\mathfrak{I}$ of $\mathfrak{T}$ such that $C^{\mathcal{I}_i} \neq \emptyset$.

Given $\mathfrak{T}$, C and i, let T be the complete and clash-free completion tree that the algorithm has constructed to support the decision. Let us construct the distributed interpretation $\mathfrak{I}$ as follows:

1. $\Delta^{\mathcal{I}_i} = V_i$, for each $i \in I$;
2. $x \in A^{\mathcal{I}_i}$, for each atomic concept $A \in \mathcal{L}_i(x)$, for each $x \in V_i$ and each $i \in I$;
3. $\langle x, y\rangle \in R^{\mathcal{I}_i}$, for each $\langle x, y\rangle \in E_i$ such that $\mathcal{L}_i(\langle x, y\rangle) = R$, for each $i \in I$, if y is not blocked;
4. $\langle x, z\rangle \in R^{\mathcal{I}_i}$ for each $\langle x, y\rangle \in E_i$ such that $\mathcal{L}_i(\langle x, y\rangle) = R$, for each $i \in I$, in case that y is blocked by z;
5. $r_{ij}(x) = y$ for each $x \in V_i$ and for each $j : y \in r_i(x)$.

Please observe that if computation was never triggered within some T_j of T during the run of the algorithm, then $\mathcal{I}_j = \mathcal{I}^\epsilon$. It remains to show, that $\mathfrak{I}$ is in fact a model of $\mathfrak{T}$ and $C^{\mathcal{I}_i} \neq \emptyset$. We will first prove the following proposition:

> given any $i \in I$, for each $E \in \mathcal{L}_i(x)$ (i.e., also for complex concepts) we have $x \in E^{\mathcal{I}_i}$.

This is proven by induction on the structure of E. We need to consider the following cases:

1. E is atomic: we know that $x \in E^{\mathcal{I}_i}$ from the construction;
2. $E = \neg E_1$, E_1 atomic: since T is clash-free, $E_1 \notin \mathcal{L}_i(x)$ and by construction $x \notin {E_1}^{\mathcal{I}_i}$. In that case however $x \in E^{\mathcal{I}_i} = \Delta^{\mathcal{I}_i} \setminus {E_1}^{\mathcal{I}_i}$;
3. $E = E_1 \sqcap E_2$: since T is complete, the $\sqcap$-rule is not applicable and hence also $E_1 \in \mathcal{L}_i(x)$ and $E_2 \in \mathcal{L}_i(x)$. By induction we now have that $x \in {E_1}^{\mathcal{I}_i}$ and $x \in {E_2}^{\mathcal{I}_i}$ and hence also $x \in E^{\mathcal{I}_i}$;
4. $E = E_1 \sqcup E_2$: since T is complete, the $\sqcup$-rule is not applicable and hence either $E_1 \in \mathcal{L}_i(x)$ or $E_2 \in \mathcal{L}_i(x)$. By induction we now either have $x \in {E_1}^{\mathcal{I}_i}$ or we have $x \in {E_2}^{\mathcal{I}_i}$. In either case however also $x \in E^{\mathcal{I}_i}$;
5. $E = \exists R.E_1$: since T is complete, the $\exists$-rule is not applicable and hence there must be $y \in V_i$, an R-successor of x such that $E_1 \in \mathcal{L}_i(y)$. By induction $y \in {E_1}^{\mathcal{I}_i}$ and by construction of $\mathfrak{I}$ $\langle x, y\rangle \in R^{\mathcal{I}_i}$. But that means that $x \in E^{\mathcal{I}_i}$;
6. $E = \forall R.E_1$: since T is complete, the $\forall$-rule is not applicable and hence for every $y \in V_i$, an R-successor of x, we have $E_1 \in \mathcal{L}_i(y)$. By induction $y \in {E_1}^{\mathcal{I}_i}$ and by construction of $\mathfrak{I}$ $\langle x, y\rangle \in R^{\mathcal{I}_i}$. But that means that $x \in E^{\mathcal{I}_i}$.

Thus, we have verified that the local interpretation $\mathcal{I}_i$ is indeed an $\mathcal{ALC}$ interpretation and that C has an instance in this interpretation (since $C \in \mathcal{L}_i(s_0)$). In addition we have in fact also proven that each i-local GCI axiom $E_1 \sqsubseteq E_2$ is satisfied by $\mathcal{I}_i$ – from the completeness of T, $\mathrm{nnf}(\neg E_1 \sqcup E_2) \in x$,

for each $x \in V_i$, and hence $x \in (\mathrm{nnf}(\neg E_1))^{\mathcal{I}_i} \cup {E_2}^{\mathcal{I}_i}$. It follows that $x \in {E_1}^{\mathcal{I}_i}$ implies $x \in {E_2}^{\mathcal{I}_i}$.

It remains to show, that $\mathfrak{I}$ is indeed a distributed model of $\mathfrak{T}$ in the adjusted semantics. First, all the bridge rules must be satisfied. Given an onto-bridge rule $k : E_1 \xrightarrow{\sqsupseteq} l : E_2 \in \mathfrak{B}$, we ought to show that $r_{kl}\big({E_1}^{\mathcal{I}_k}\big) \supseteq {E_2}^{\mathcal{I}_l}$. So let $x \in {E_2}^{\mathcal{I}_l}$, that is $x \in V_l$ and $E_2 \in \mathcal{L}_l(x)$. But T is complete, $\xrightarrow{\sqsupseteq}$-rule is not applicable, and hence there must be $y \in V_k$ with $E_1 \in \mathcal{L}_k(y)$ and $l : x \in r_k(y)$. That means however that $y \in {E_1}^{\mathcal{I}_k}$ and $\langle x, y\rangle \in r_{kl}$, and so $x \in r_{kl}\big({E_1}^{\mathcal{I}_k}\big)$. Therefore the bridge rule is satisfied by $\mathfrak{I}$.

Given an into-bridge rule $k : E_1 \xrightarrow{\sqsubseteq} l : E_2 \in \mathfrak{B}$, we ought to show that $r_{kl}\big({E_1}^{\mathcal{I}_k}\big) \subseteq {E_2}^{\mathcal{I}_l}$. Let $x \in r_{kl}\big({E_1}^{\mathcal{I}_k}\big)$. Then there is $y \in V_k$ such that $l : x \in r_l(y)$. But since T is complete, it must be the case that $E_2 \in \mathcal{L}_l(x)$ and hence $x \in {E_2}^{\mathcal{I}_l}$. Therefore the bridge rule is satisfied by $\mathfrak{I}$.

The last thing in order to verify that $\mathfrak{I}$ is indeed a model of $\mathfrak{T}$, is to show that $\mathfrak{I}$ satisfies the transitivity requirement. We ought to show that for each $k, l, m \in I$, if $y \in r_{kl}(x)$ and $z \in r_{lm}(y)$ then also $z \in r_{km}(x)$. Given the two assumptions we know by construction that $l : y \in r_k(x)$ and $m : z \in r_l(y)$. That means that $x \in V_k$ was initialized after several "chained" applications of the $\xrightarrow{\sqsupseteq}$-rule starting from $y \in V_l$ and y in turn after several "chained" $\xrightarrow{\sqsupseteq}$-rule applications starting from $z \in V_m$. In that case however $r_l(y) \subseteq r_k(x)$. Hence $m : z \in r_k(x)$ and so $z \in r_{km}(x)$.

And thus $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$ that satisfies the transitivity condition and ${C}^{\mathcal{I}_i} \neq \emptyset$ – in other words, C is satisfiable in $\mathcal{T}_i$ with respect to $\mathfrak{T}$.

Completeness. Given $\mathfrak{T}$ with index set I, a concept C and $i \in I$ such that C is satisfiable in $\mathcal{T}_i$ with respect to $\mathfrak{T}$, we shall prove that the algorithm answers "C is satisfiable", if run on input $\mathfrak{T}$, C and i. Let $\mathfrak{I}$ be a distributed model of $\mathfrak{T}$ with ${C}^{\mathcal{I}_i} \neq \emptyset$, we know that one must exist. We will simulate the run of the algorithm. During the initialization T_i is set to $\langle \{s_0\}, \emptyset, \{s_0 \mapsto \{C\}\}, \emptyset\rangle$ and all the other T_j, $i \neq j$, are set to $\langle \emptyset, \emptyset, \emptyset, \emptyset\rangle$. There is no clash in T. We will show by induction (on the number of tableau expansion steps) that after each tableau expansion step the completion tree T is expanded in such a way that no clash is introduced. In order to demonstrate this, we inductively construct an auxiliary mapping $\pi : \bigcup_{i \in I} V_i \to \bigcup_{i \in I} \Delta^{\mathcal{I}_i}$ to track the relation between $\mathfrak{I}$ and T. This mapping will keep the property $(*)$:

> for each node $x \in V_i$, for each $i \in I$: if $C \in \mathcal{L}_i(x)$ then $\pi(x) \in {C}^{\mathcal{I}_i}$.

Let x be an arbitrary member of ${C}^{\mathcal{I}_i}$, place $\pi(s_0) = x$. So, if a tableaux expansion rule is triggered in some $x \in V_i$ of the clash-free completion tree T (from induction hypothesis), we consider several cases, based on the kind of completion rule that was triggered:

1. $\sqcap$-rule is triggered in x because $E_1 \sqcap E_2 \in \mathcal{L}_i(x)$. Then by induction hypothesis $\pi(x) \in E_1 \sqcap {E_2}^{\mathcal{I}_i}$. In that case also $\pi(x) \in {E_1}^{\mathcal{I}_i}$ and $\pi(x) \in {E_2}^{\mathcal{I}_i}$, and hence the property $(*)$ is maintained after the algorithm adds E_1 and E_2 to $\mathcal{L}_i(x)$;

2. $\sqcup$-rule is triggered in x because $E_1 \sqcup E_2 \in \mathcal{L}_i(x)$. Then by induction hypothesis $\pi(x) \in E_1 \sqcup {E_2}^{\mathcal{I}_i}$. In that case however either $\pi(x) \in {E_1}^{\mathcal{I}_i}$ or $\pi(x) \in {E_2}^{\mathcal{I}_i}$. Without loss of generality, let it be the case that $\pi(x) \in {E_1}^{\mathcal{I}_i}$. Again, without loss of generality, we assume that the algorithm adds E_1

to $\mathcal{L}_i(x)$, as a non-deterministic decision takes place. Hence the property $(*)$ is maintained after this step;

3. $\exists$-rule is triggered in x because it has an R-successor y, and $\exists R.E_1 \in \mathcal{L}_i(x)$. Then $\pi(x) \in \exists R.{E_1}^{\mathcal{I}_i}$ and so there is some $y' \in \Delta^{\mathcal{I}_i}$ such that $y' \in {E_1}^{\mathcal{I}_i}$ and $\langle \pi(x), y' \rangle \in R^{\mathcal{I}_i}$. When the algorithm generates a new node y in this step we set $\pi(y) = y'$ and the property $(*)$ is maintained;

4. $\forall$-rule is triggered in x because it has an R-successor y, and $\forall R.E_1 \in \mathcal{L}_i(x)$. As a consequence E_1 is added to $\mathcal{L}_i(y)$. We know that y was created by an application of the $\exists$-rule and hence $\pi(y)$ is an R-successor of x in $\mathcal{I}_i$. But $\mathcal{I}_i$ is a model of $\mathcal{T}_i$ and hence $\pi(y) \in y^{\mathcal{I}_i}$ since from induction hypothesis we know that $\pi(x) \in \forall R.{E_1}^{\mathcal{I}_i}$. Hence $(*)$ is maintained after this step;

5. $\mathcal{T}$-rule is triggered in x, with the result of adding $\mathrm{nnf}(\neg E_1 \sqcup E_2)$ into $\mathcal{L}_i(x)$. Then $(*)$ is maintained as $\mathcal{I}_i$ is a model of $\mathcal{T}_i$ and so it holds that $\pi(x) \in (\mathrm{nnf}(\neg E_1 \sqcup E_2))^{\mathcal{I}_i}$;

6. $\xrightarrow{\sqsupseteq}$-rule is triggered in x because $E_2 \in \mathcal{L}_i(x)$ and $j : E_1 \xrightarrow{\sqsupseteq} i : E_2 \in \mathfrak{B}$. From induction hypothesis, $\pi(x) \in {E_2}^{\mathcal{I}_i}$, and hence $\pi(x) \in r_{ji}(y')$ for some $y' \in {E_1}^{\mathcal{I}_j}$. Set $\pi(y) = y'$ for the node y that is newly created in V_i as the result of this expansion step. The label $\mathcal{L}_j(y)$ has been set to E_1 but since $y' \in {E_1}^{\mathcal{I}_j}$ then $(*)$ is maintained;

7. $\xrightarrow{\sqsubseteq}$-rule is triggered in x because of $E_1 \in \mathcal{L}_i(x)$, $j : y \in r_i(x)$ and $i : E_1 \xrightarrow{\sqsubseteq} j : E_2 \in \mathfrak{B}$, with the result of adding E_2 into $\mathcal{L}_j(y)$. From the induction hypothesis we have $\pi(x) \in {E_1}^{\mathcal{I}_i}$. Observe that the definition of $\xrightarrow{\sqsupseteq}$-rule and the inductive construction of $\pi(\cdot)$ in previous steps also assure that $j : y \in r_i(x)$ implies $\pi(y) \in r_{ij}(\pi(x))$. This is because initialization of new $y \in V_j$ always follows incoming r-edge and r is transitive because of computational consistency. It follows that $\pi(y) \in {E_2}^{\mathcal{I}_j}$, since $\mathfrak{I}$ is a distributed model of $\mathfrak{T}$ and the bridge rule assures this. Hence $(*)$ is maintained even after this step.

We already know that the algorithm always terminates. Once this happens, it follows that T is now complete, because no rule is applicable, and clash-free, because the property $(*)$ is maintained all the way up to this point. Hence the algorithm answers "C is satisfiable" and hence the theorem. □

The algorithm for the original DDL semantics of (Serafini et al. 2005) is obviously truly distributed in the sense that it supports a scenario in which several autonomous reasoning services run independently, one for each local ontology, and communicate by passing queries. While this is also the case for the newly introduced algorithm, it is not necessarily that obvious. In order to clarify this, we provide a message protocol that handles $\xrightarrow{\sqsupseteq}$-rule and $\xrightarrow{\sqsubseteq}$-rule execution in a truly distributed fashion, and it also collects the information that the completion tree is complete within the reasoner that has initialized the computation.

The algorithm employs three different types of messages $\mathsf{querySat}(x, r, C)$, $\mathsf{answerSat}(x, C, answer)$, and $\mathsf{pushConcept}(x, C)$. When asked for satisfiability of some j-local concept C with respect to $\mathfrak{T}$, it starts by initializing the local completion tree T_j. More local completion trees are initialized during the

Message querySat(x, r, C):
Send: if $G \in \mathcal{L}_j(x)$ for some $x \in V_j$, $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$, and x is not blocked, pass the message querySat$(x, r_j(x), C)$ to T_i.
Receive: upon receipt of a message querySat(x, r, C) from T_i, create new completion tree $\langle\{s_0\}, \emptyset, \{s_0 \mapsto \{C\}\}, \{s_0 \mapsto \{i : x\} \cup r\}\rangle$ within T_j and start the computation. Record the sent message in S_j.

Message answerSat$(x, C, answer)$:
Send: if the local completion tree for concept C within T_j, that has been initialized due the message querySat(x, r, C) previously received from T_i, is now complete, and there are no outgoing messages recorded in S_j, send the message answerSat$(x, C, answer)$ to $\mathcal{T}_i$ with $answer$ set to true if T_j is clash-free and to false otherwise.
Receive: upon receipt of a message answerSat$(x, C, answer)$ from $\mathcal{T}_i$, remove the message querySat(x, r, C) from S_j, that has been previously send to T_i. If $answer$ is false then add $E \sqcap \neg E$ to $\mathcal{L}_j(x)$ for some new concept name E.

Message pushConcept(x, C):
Send: if D has been added to $\mathcal{L}_j(x)$ in the previous step, for some $x \in V_j$, then for each $j : D \xrightarrow{\sqsubseteq} i : H \in \mathfrak{B}$ s.t. $i : y \in r_j(x)$ send the message pushConcept(y, H) to T_i.
Receive: upon receipt of a message pushConcept(x, C) from $\mathcal{T}_i$, add C to $\mathcal{L}_j(x)$.

Figure 5.10: Message protocol of the distributed tableaux algorithm

runtime by passing a querySat(x, r, C) message every time the $\xrightarrow{\sqsupseteq}$-rule is fired. When $\xrightarrow{\sqsubseteq}$-rule is fired, the consequences of the into-bridge rule are propagated by passing a pushConcept(x, C) message. Once a local completion tree T_k is complete this is announced by passing a answerSat$(x, C, answer)$ message to the local reasoner that has triggered the computation in $\mathcal{T}_k$.

More detailed specification of the protocol messages is given in Fig. 5.10. For each kind of message we specify when the message is sent from some local reasoning service T_j and also what happens when the message is received in some local reasoning service T_j (T_j is always the local reasoning service in the figure). An auxiliary data structure S_j is introduced in each T_j in order to track messages that have been sent in order to assure termination.

5.4 Summary

In this chapter, we have studied subsumption propagation between remote ontologies in DDL. These are local ontology units which are not connected directly by bridge rules, but are instead connected only indirectly by so called chains of bridge rules. As there are two types of bridge rules, we distinguish between chains of into-bridge rules and chains of onto-bridge rules.

We have learned, that under the original DDL semantics the effect of subsumption propagation between remote ontologies is rather minute. This has lead us to the study of different strengthenings of the semantics that could possibly exhibit increased level of subsumption propagation between remote ontologies.

Starting from the idea of compositionally consistent domain relation, which has been previously applied in P-DL (Bao et al. 2006*b*, 2009), we have studied three such strengthenings of the DDL semantics.

In our first proposal, we require, that the domain relation in each admissible distributed model must satisfy the compositional consistency requirement. We have shown, that under this semantics, subsumption propagation between remote ontologies increases to the satisfactory level, however some important desiderata for DDL (notably, directionality and local inconsistency) are not satisfied. In our second approach we have shown, that if compositional consistency is applied more cautiously, only in cases when two local ontologies are in fact connected by a path of bridge rules, subsumption propagation is at the same level as in the previous approach. In addition, directionality is satisfied under such semantics. The local inconsistency property is, however, violated.

Finally, in our third proposal, we only require transitivity of the domain relation, which is a condition weaker than compositional consistency. Under this semantics, subsumption only propagates along chains of onto-bridge rules. It fails to propagate along into-chains. On the other hand, directionality and local inconsistency, as well as all other widely acknowledged desired properties of DDL are satisfied.

For the latter of the three approaches, that is, for the semantics of DDL with transitive domain relations, we have developed a distributed tableaux reasoning algorithm that decides satisfiability of concepts and subsumption entailment. As in the case of the tableaux algorithm that is known for the original semantics (Serafini & Tamilin 2004), the newly introduced algorithm is also fully distributed and it permits the scenario where every local ontology is governed by an autonomous reasoning service and these services communicate by passing queries. The newly proposed algorithm introduces domain relation tracking into the tableau, and we believe it can possibly be adjusted also for different semantics with restricted domain relations, such as for instance the semantics with injectivity which we have studied in Chap. 4.

The algorithm that we have introduced, handles DDL knowledge bases over $\mathcal{ALC}$ with acyclic bridge graph. An extension of this algorithm towards more general languages, most importantly $\mathcal{SHIQ}$, and dealing with cyclic bridge graph is an open issue. As we have not yet been able to deal with chains of into-bridge rules, also further investigation of compositional consistency or any other variant of the semantics that would exhibit improved behaviour is highly relevant. Remaining research problems related to this work include practical evaluation of the algorithm by an implementation, complexity analysis for the decision problems, and combining, within an unified framework, the semantic adjustments that we proposed in this chapter with those proposed in Chap. 4.

Chapter 6

Relating Distributed Ontology Representation Frameworks

The past ten years have seen several proposals of logical knowledge representation frameworks which share the explicit goal to define a formal semantics for distributed and modular ontologies. While some of them are based on first order logic or modal logics (Subrahmanian 1994, Fagin et al. 1995, Ghidini & Serafini 1998, Giunchiglia & Ghidini 1998), the most influential formalisms for distributed and modular ontologies are those based on DL. In this chapter we present and compare four such formalisms:

Distributed Description Logics (DDL). A framework for combining DL ontologies by means of directed semantic mapping, originally introduced by Borgida & Serafini (2002);

$\mathcal{E}$-connections. A framework for combining non-overlapping ontologies encoded in DL but, possibly also other logics, by special inter-ontology roles, first introduced by Kutz et al. (2002);

Package-based Description Logics (P-DL). A framework for distributed ontologies that enables import of ontology entities between the modules, first introduced by Bao & Honavar (2004);

Integrated Distributed Description Logics (IDDL). Another framework that enables aligning DL ontologies by means of bidirectional semantic mapping, introduced by Zimmermann (2007).

The idea of distributed and modular ontologies opens a number of different issues, each of which deserves a specific investigation. Even if reductions between the formalisms are known (Kutz et al. 2004, Bao et al. 2006*a*, Cuenca Grau et al. 2007) and all of them are reducible into a regular monolithic DL ontology (Borgida & Serafini 2003, Cuenca Grau & Kutz 2007, Bao et al. 2009, Zimmermann & Le Duc 2008), it has to be remarked that each of the formalisms is focused on different modeling scenarios and is suited for different operations that have been enabled by the introduction of modular ontologies.

DDL and IDDL follow the mapping approach and are particularly motivated by scenarios where two or more ontology units cover the same domain or partially overlapping domains. They aim at combination of the ontology units and for resolving the modeling heterogeneity between the ontology units. Thus they facilitate knowledge reuse. While DDL work with directed mapping, IDDL make use of bidirectional mapping.

$\mathcal{E}$-connections are intended for modeling scenarios where the modeling domains of the ontology units are mutually disjoint – the overall modeling domain is split into disjoint parts. These domains are then combined with use of links. Complex concepts are possibly constructed by restrictions on links that involve also foreign concepts, and thus knowledge is reused. It is remarkable, that if we are able to identify such non-overlapping parts within a domain of a single ontology, the ontology may be partitioned into and represented by $\mathcal{E}$-connections (Cuenca Grau et al. 2005, Cuenca Grau, Parsia, Sirin & Kalyanpur 2006).

P-DL, in turn, aim at ontology importing, allowing ontology entities to be imported from one ontology unit to another. With these entities also semantic relations are imported, most notably concept subsumption, but possibly also other. This way the modeling domain of the importing ontology unit is enriched by new concepts originally part of the domain of the source ontology unit and thus knowledge is reused.

This chapter is structured as follows. Sect. 6.1 provides an abstract overview of these formalisms. We then proceed by introducing the four distributed ontology frameworks, DDL, $\mathcal{E}$-connections, P-DL and IDDL, in Sect. 6.2–6.5. In Sect. 6.6, we review the existing results and consecutively present our own results on comparison of these frameworks. The chapter is summarized in its final section.

6.1 Reference Distributed Ontology Framework

While different ontology frameworks have many distinctive features that make each of them suitable for different applications, in general an unifying view on these frameworks is possible. In this section we describe the features that are common to all the formalisms of our interest in this chapter. These basic logical components, defined by all the formalisms, are as follows:

Set of modules. Sometimes called local ontologies or knowledge sources, they constitute the modules that have to be integrated. This set is represented by a family of ontologies $\{O_i\}_{i\in I}$ indexed by a set of indexes I. The set of components is finite, and it is fixed and constant (i.e., once specified, there are no means to add, remove or merge components by means of logic). Each index uniquely identifies an ontology O_i which is to be integrated. One important parameter of such a component is the language in which each O_i is expressed, so called local language. For the comparison in this chapter we assume that the local language is always some language from the family of Description Logics, however also approaches that allow integration of DL with other local logics are known (Kutz et al. 2004).

Connection between the modules. This component represents the connection between the local modules within the framework. Each of the frameworks introduces different constructs to represents such a component. This

is indeed the core aspect that differentiates the approaches. As outlined in the introduction, there are three basic approaches: ontology mapping (DDL, IDDL), ontology linking ($\mathcal{E}$-connections) and ontology import (P-DL). Another important parameter is the type of objects that are connected. Some approaches allow to link only concepts, while other approaches allow to link also relations and individuals.

Semantics. Each of the approaches provides a formal semantics of the logic as an extension (or restriction) of the DL semantics. We distinguish two different philosophies: *centralized semantics*, there is a unique domain of interpretation in which each ontology component is interpreted; *distributed semantics* local ontologies are interpreted in a "private" domain, and the integration is reached by imposing some compatibility constraints on the possible combinations of local semantics.

Axiomatization and reasoning support. One of the main goals of this kind of research is to provide a sound and complete reasoning algorithm which computes the consequences of the integration, and to implement this algorithm into a reasoning engine. This goal is best approached by DDL and $\mathcal{E}$-connections which both have a reasoner available. For P-DL and IDDL reasoning algorithms are known. These algorithms are usually based on an extension of the tableaux based reasoning techniques that are available for DL. An important point is whether the algorithm performs distributed reasoning or not. For some of the approaches an axiomatization is known. Also here we have two philosophies: some approaches are limited and only consider the consequences of integration within the modules, while others are interested also in reasoning on the mapping component.

We will now return to DDL and introduce a generalization of this framework, reviewing the state of the art and discussing also the known extensions thereof. Consequently we review also the other frameworks and finally we present some novel reductions between these frameworks.

6.2 Distributed Description Logics

In Chap. 3 we have introduced a version of DDL build on top of $\mathcal{SHIQ}$ that enables basic bridge rules between concepts. This version has been most frequently investigated in the literature (Borgida & Serafini 2003, Serafini & Tamilin 2004, Serafini et al. 2005), and we have also rooted our research, presented in Chap. 4 and 5, in this version of the framework. As we have already noted in Sect. 3.4, the DDL framework has been further enriched by consecutive research (Serafini & Tamilin 2006, 2007, Ghidini & Serafini 2006*a*, Ghidini et al. 2007, 2008). In addition, an extension of OWL, that introduces semantic mapping into OWL and with semantics based on DDL has been introduced under the name Contextual OWL, or C-OWL (Bouquet et al. 2003).

In order to compare DDL with other distributed and modular ontology formalisms, we will need a richer DDL framework with greater expressive power, which we formally introduce below. Out of the possible extensions that have been investigated we will leave out only the heterogeneous bridge rules (Ghidini et al. 2007, 2008). DDL with heterogeneous bridge rules constitute a unique ontology representation framework with no relation to the other formalism.

6.2.1 Formalization

In the literature, local logics as expressive as $\mathcal{SHIQ}$ and $\mathcal{ALCQIb}$, have been used as local languages for DDL. The least common super language of these two is $\mathcal{SHIQb}$, which we will use in this section, in order to build a generalized version of DDL on top of it. To our best knowledge, the decidability of $\mathcal{SHIQb}$ is still an open problem. This is not a constraint for us, however, as we will assume that for a particular application one chooses a sublanguage which is decidable.

Definition 83 (Syntax of DDL)**.** *Assume a non-empty index set I, a family of sets of atomic concept names $N_{\mathrm{C}} = \{N_{\mathrm{C}i}\}_{i\in I}$, a family of sets of role names $N_{\mathrm{R}} = \{N_{\mathrm{R}i}\}_{i\in I}$ and a family of sets of individual names $N_{\mathrm{I}} = \{N_{\mathrm{I}i}\}_{i\in I}$. Let $\mathcal{L}_i$ be a sublanguage of $\mathcal{SHIQb}$, given any $i \in I$. DDL knowledge bases are constructed as follows:*

1. *a distributed TBox $\mathfrak{T} = \{\mathcal{T}_i\}_{i\in I}$ over I is a family of TBoxes, such that $\mathcal{T}_i$ is a TBox in the language $\mathcal{L}_i$, build over $N_{\mathrm{C}i}$ and $N_{\mathrm{R}i}$;*

2. *a distributed RBox $\mathfrak{R} = \{\mathcal{R}_i\}_{i\in I}$ over I is a family of RBoxes, where each $\mathcal{R}_i$ is an RBox in $\mathcal{L}_i$, build over $N_{\mathrm{R}i}$;*

3. *a distributed ABox $\mathfrak{A} = \{\mathcal{A}_i\}_{i\in I}$ over I is a family of ABoxes, where each $\mathcal{A}_i$ is an ABox in $\mathcal{L}_i$, build over $N_{\mathrm{C}i}$, $N_{\mathrm{R}i}$, and $N_{\mathrm{I}i}$;*

4. *a bridge rule from i to j is an expression of two possible forms (into-bridge rule on the left, onto-bridge rule on the right):*

$$i : X \xrightarrow{\sqsubseteq} j : Y \ , \qquad\qquad i : X \xrightarrow{\sqsupseteq} j : Y \ ,$$

   where X and Y are either two concepts or two atomic roles, each in the respective language;

5. *an individual correspondence from i to j is an expression of two possible forms (partial individual correspondence on the left, total individual correspondence on the right):*

$$i : a \longmapsto j : b \ , \qquad\qquad i : a \xmapsto{=} j : \{b_1, \ldots, b_n\} \ ,$$

   where $a \in N_{\mathrm{I}i}$ and $b, b_1, \ldots, b_n \in N_{\mathrm{I}j}$ are individuals. Individual correspondences are also called bridge rules between individuals;

6. *a set of bridge rules $\mathfrak{B}$ over I is any set of into- and onto-bridge rules, partial and total individuals correspondences. Any such $\mathfrak{B}$ decomposes into $\mathfrak{B} = \bigcup_{i,j\in I, i\neq j} \mathfrak{B}_{ij}$, such that each $\mathfrak{B}_{ij}$ contains only bridge rules and individual correspondences form i to j;*

7. *a DDL knowledge base over I is a quadruple $\langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ with all four components ranging over I.*

As usual, a shorthand expression $i : X \xrightarrow{\equiv} j : Y \in \mathfrak{B}$ represents the fact that both $i : X \xrightarrow{\sqsubseteq} j : Y$ and $i : X \xrightarrow{\sqsupseteq} j : Y$ belong to $\mathfrak{B}$. The local knowledge base residing at an index $i \in I$ will be denoted by $\mathcal{K}_i = \langle \mathcal{T}_i, \mathcal{R}_i, \mathcal{A}_i \rangle$. And, as

usual, by $i : \phi$ we denote any local axiom ϕ in the language $\mathcal{L}_i$ that belongs to the respective component of $\mathcal{K}_i$.

It should be remarked that the usual DDL notation is significantly changed by Definition 83. Most notably, a DDL knowledge base may no loner be referred to as a distributed TBox, instead distributed TBox is now a name for just one of the components thereof. This shift in the point of view reflects the fact that we are no longer solely interested in the terminological knowledge and relations between concepts, but we are equally interested into all types of ontological knowledge including the data. Also, we aim to compare DDL with different other formalisms and this improved notation will prove to be practical when it comes to do that.

Definition 84 (Semanitcs of DDL)**.** *Given a distributed knowledge base* $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ *with an index set* I *and given some distributed interpretation* $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i,j \in I, i \neq j} \rangle$ *over* I*, which is defined as usual,* $\mathfrak{I}$ *satisfies the components of* $\mathfrak{K}$ *(denoted by* $\mathfrak{I} \models_\epsilon \cdot$*) as follows:*

1. $\mathfrak{I} \models_\epsilon i : C \sqsubseteq D$ *if* $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.
2. $\mathfrak{I} \models_\epsilon \mathcal{T}_i$ *if* $\mathfrak{I} i : \phi$ *for each* $i : \phi \in \mathcal{T}_i$.
3. $\mathfrak{I} \models_\epsilon \mathfrak{T}$ *if* $\mathfrak{I} \models_\epsilon \mathcal{T}_i$ *for each* $i \in I$.
4. $\mathfrak{I} \models_\epsilon i : R \sqsubseteq S$ *if* $R^{\mathcal{I}_i} \subseteq S^{\mathcal{I}_i}$;
5. $\mathfrak{I} \models_\epsilon i : \mathrm{Trans}(R)$ *if* $R^{\mathcal{I}_i}$ *is a transitive relation;*
6. $\mathfrak{I} \models_\epsilon \mathcal{R}_i$ *if* $\mathfrak{I} \models_\epsilon i : \phi$ *for each* $i : \phi \in \mathcal{R}_i$;
7. $\mathfrak{I} \models_\epsilon \mathfrak{R}$ *if* $\mathfrak{I} \models_\epsilon \mathcal{R}_i$ *for each* $i \in I$;
8. $\mathfrak{I} \models_\epsilon i : C(a)$ *if* $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$;
9. $\mathfrak{I} \models_\epsilon i : R(a, b)$ *if* $\langle a^{\mathcal{I}_i}, b^{\mathcal{I}_i} \rangle \in R^{\mathcal{I}_i}$;
10. $\mathfrak{I} \models_\epsilon \mathcal{A}_i$ *if* $\mathfrak{I} \models_\epsilon \phi$ *for each* $\phi \in \mathcal{A}_i$;
11. $\mathfrak{I} \models_\epsilon \mathfrak{A}$ *if* $\mathfrak{I} \models_\epsilon \mathcal{A}_i$ *for each* $i \in I$;
12. $\mathfrak{I} \models_\epsilon i : X \xrightarrow{\sqsubseteq} j : Y$ *if* $r_{ij}\big(X^{\mathcal{I}_i}\big) \subseteq Y^{\mathcal{I}_j}$;
13. $\mathfrak{I} \models_\epsilon i : X \xrightarrow{\sqsupseteq} j : Y$ *if* $r_{ij}\big(X^{\mathcal{I}_i}\big) \supseteq Y^{\mathcal{I}_j}$;
14. $\mathfrak{I} \models_\epsilon i : a \longmapsto j : b$ *if* $b^{\mathcal{I}_j} \in r_{ij}\big(a^{\mathcal{I}_i}\big)$;
15. $\mathfrak{I} \models_\epsilon i : a \xmapsto{=} j : \{b_1, \ldots, b_n\}$ *if* $r_{ij}\big(a^{\mathcal{I}_i}\big) = \{{b_1}^{\mathcal{I}_j}, \ldots, {b_n}^{\mathcal{I}_j}\}$;
16. $\mathfrak{I} \models_\epsilon \mathfrak{B}$ *if* $\mathfrak{I} \models_\epsilon \phi$ *for all axioms* $\phi \in \mathfrak{B}$.

The standard semantics of DDL relies on the notion of holes which may possibly appear as local interpretations. Models under this semantics are called ϵ-models. As we have learned, besides for ϵ-models there are d-models, the former permitting holes, the latter not.

Definition 85 (ϵ-model)**.** $\mathfrak{I}$ *is a distributed model (*ϵ*-model) of* $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ *if* $\mathfrak{I} \models_\epsilon \mathfrak{T}$, $\mathfrak{I} \models_\epsilon \mathfrak{R}$, $\mathfrak{I} \models_\epsilon \mathfrak{A}$ *and* $\mathfrak{I} \models_\epsilon \mathfrak{B}$.

Definition 86 (d-model)**.** *An ϵ-model $\mathfrak{I}$ of $\mathfrak{K}$ is a d-model of $\mathfrak{K}$ if it is globally consistent.*

All the other DDL related notions, such as the bridge graph $G_{\mathfrak{K}}$ and the reduct $\mathfrak{K} \setminus\setminus J$, for $J \subseteq I$, generalize in a straightforward fashion and the reader is referred to Chap. 3 for the formal definitions. Also the two normal forms introduced in Chap. 3 both apply. We now shift our attention towards reasoning tasks. First task is the consistence of knowledge bases.

Definition 87 (Consitence)**.** *A DDL knowledge base $\mathfrak{K}$ is i-consistent, if there exists an ϵ-model of $\mathfrak{K}$ in which $\Delta^{\mathcal{I}_i}$ is nonempty. $\mathfrak{K}$ is globally consistent if it has a d-model.*

Next comes the reasoning task of concept satisfiability. As we are now used to it, we will have one notion corresponding to ϵ-models and another one derived from d-models.

Definition 88 (Concept satisfiabilty)**.** *An i-concept C is said to be satisfiable (ϵ-satisfiable) with respect to a DDL knowledge base $\mathfrak{K}$ if there exists an ϵ-model $\mathfrak{I}$ of $\mathfrak{K}$ such that $C^{\mathcal{I}_i} \neq \emptyset$. C is d-satisfiable with respect to $\mathfrak{K}$ if there exists a d-model with this property.*

Finally comes the notion of entailment. While typically only the entailment of concept subsumption is studied, it is possible to define entailment of any other formula. We will rely on this fact in order to relate DDL to IDDL, which considers a much larger set of entailment reasoning tasks.

Definition 89. *Given a DDL knowledge base $\mathfrak{K}$ over an index set I and an i-local formula ϕ (which is a concept subsumption, role inclusion, or an ABox assertion formula), we say that $\mathfrak{K}$ entails (ϵ-entails) ϕ with respect to the local index i (denoted by $\mathfrak{K} \models_\epsilon i : \phi$), if for every ϵ-model of $\mathfrak{K}$ we have $\mathfrak{I} \models_\epsilon i : \phi$. $\mathfrak{K}$ d-entails ϕ with respect to the local index i (denoted by $\mathfrak{K} \models_\mathrm{d} i : \phi$), if for every d-model of $\mathfrak{K}$ we have $\mathfrak{I} \models_\mathrm{d} i : \phi$.*

Since we consider very general DDL knowledge bases including ABoxes, the consistence of the knowledge base is the main reasoning task. Concept subsumption entailment, instance checking (entailment of concept assertion formulae) and satisfiability checking are reducible into the consistence problem in the usual way (Baader et al. 2003). The remaining reasoning tasks would be considered nonstandard also for monolithic DL.

The DDL reasoner DRAGO also handles reasoning with partial individual correspondences (Serafini & Tamilin 2006, 2007, Tamilin 2007). To the best of our knowledge, a specific reasoner for DDL that include bridge rules between roles is not known. However, Ghidini et al. (2007) and Ghidini et al. (2008), provide a set of inference rules which, when exhaustingly applied, propagate all knowledge between the ontology modules, and afterwards all consequences can be computed locally. Thus reasoning with this expressive bridge rules is possible whenever a reasoner for the local language is available.

6.2.2 Modeling with DDL

The new features that have been added to the DDL framework bring in new modeling possibilities, as we briefly illustrate in this section. Let us start a

running example with a rather basic DDL scenario involving mapping between concepts. Later on we will extend it with other kinds of mapping.

Example 21. *Let us build a small ontology that will track all kinds of professions in some IT Company and relations between this professions. In the TBox $\mathcal{T}_1$ we have among others the following axioms:*

$$\mathsf{ITStaff} \sqsubseteq \mathsf{Employee} \qquad \mathsf{SupportStaff} \sqsubseteq \mathsf{Employee}$$
$$\mathsf{CateringStaff} \sqsubseteq \mathsf{SupportStaff} \qquad \mathsf{AdministrativeStaff} \sqsubseteq \mathsf{SupportStaff}$$
$$\mathsf{Cook} \sqsubseteq \mathsf{CateringStaff} \qquad \mathsf{Chef} \sqsubseteq \mathsf{Cook}$$

Let us also have an individual johnSmith *who serves as a chef in the catering division. We assert this in the ABox $\mathcal{A}_1$:*

$$\mathsf{Chef}(\mathsf{johnSmith})$$

This simple modeling allows us to derive some entailed knowledge (for instance, $\mathsf{Cook}(\mathsf{johnSmith})$, $\mathsf{CateringStaff}(\mathsf{johnSmith})$, $\mathsf{Cook} \sqsubseteq \mathsf{Employee}$, *etc.). Suppose that one day we run into a readily available ontology $\mathcal{T}_2$ based upon the Escoffier's brigade de cuisine system (Dominé 1998). The excerpt of the ontology $\mathcal{T}_2$ is as follows:*

$$\mathsf{KitchenChef} \sqsubseteq \mathsf{Cook} \qquad \mathsf{Saucemaker} \sqsubseteq \mathsf{Cook}$$
$$\mathsf{PastryCook} \sqsubseteq \mathsf{Cook} \qquad \mathsf{Baker} \sqsubseteq \mathsf{PastryCook}$$

Suppose that we decide to reuse the knowledge of $\mathcal{T}_2$ within our organization. We will keep the ontology $\mathcal{T}_1$ cleaner and reuse knowledge of $\mathcal{T}_2$ whenever possible. First, we remove the two axioms $\mathsf{Cook} \sqsubseteq \mathsf{CateringStaff}$ *and* $\mathsf{Chef} \sqsubseteq \mathsf{Cook}$ *from $\mathcal{T}_1$ because they are redundant in light of $\mathcal{T}_2$. We do keep, however, the concepts* Cook *and* Cheff *in $\mathcal{T}_1$, as these are positions that some people occupy in our company (e.g.,* johnSmith*). In order to integrate the ontologies, we add the following bridge rules:*

$$2 : \mathsf{Cook} \xrightarrow{\sqsubseteq} 1 : \mathsf{CateringStaff} \qquad 2 : \mathsf{Cook} \xrightarrow{\equiv} 1 : \mathsf{Cook}$$
$$2 : \mathsf{KitchenChef} \xrightarrow{\sqsupseteq} 1 : \mathsf{Chef}$$

Thanks to the mapping expressed by the bridge rules, the two subsumptions that we have previously removed from the TBox $\mathcal{T}_1$ are now entailed (i.e., $\mathfrak{K} \models_\epsilon 1 : \mathsf{Chef} \sqsubseteq \mathsf{Cook}$ and $\mathfrak{K} \models_\epsilon 1 : \mathsf{Cook} \sqsubseteq \mathsf{CateringStaff}$). We are also equally able to derive $\mathfrak{K} \models_\epsilon 1 : \mathsf{Cook}(\mathsf{johnSmith})$, $\mathfrak{K} \models_\epsilon 1 : \mathsf{CateringStaff}(\mathsf{johnSmith})$ and $\mathfrak{K} \models_\epsilon 1 : \mathsf{Cook} \sqsubseteq \mathsf{Employee}$, etc.

With individual correspondence axioms we are able to encode mapping between individuals. We illustrate this by an example.

Example 22. *Let us extend the distributed ontology from Example 21. Suppose that the accounting department decides to create their own ontology where they model things in the company in a way that is more practical for the accountants. In the local TBox $\mathcal{T}_3$ we have:*

$$\mathsf{Accountant} \sqsubseteq \mathsf{Employee} \qquad \mathsf{Director} \sqsubseteq \mathsf{Employee}$$
$$\mathsf{SeniorExecutive} \sqsubseteq \mathsf{Employee} \qquad \mathsf{JuniorExecutive} \sqsubseteq \mathsf{Employee}$$
$$\mathsf{BasicStaff} \sqsubseteq \mathsf{Employee} \qquad \mathsf{SupportStaff} \sqsubseteq \mathsf{Employee}$$

This ontology then makes reuse of the knowledge already existing within the distributed knowledge base, employing the bridge rules:

$$1 : \mathsf{CateringStaff} \xrightarrow{\sqsubseteq} 3 : \mathsf{SupportStaff}$$

$$1 : \mathsf{ITStaff} \xrightarrow{\sqsubseteq} 3 : \mathsf{BasicStaff}$$

In addition the accounting department keeps its own evidence of all employees in $\mathcal{A}_3$, using individuals with special nomenclature relying on employee numbers (i.e., individuals such as $\mathsf{e006B3F}, \mathsf{eF9DB15}, \mathsf{eE23A28}$*, etc.). Using individual correspondence axioms, these individuals are matched with other representations of the same employee in the knowledge base, for instance:*

$$1 : \mathsf{johnSmith} \longmapsto 3 : \mathsf{e006B3F}$$

Now, even without incorporating into $\mathcal{T}_3$ concepts such as Cook *and* Chef *which are of little interest to company accountants and without explicit recording of all information about the precise working position of* e006B3F *in $\mathcal{T}_3$ and $\mathcal{A}_3$ we are able to derive $\mathfrak{T} \models_\epsilon 3 : \mathsf{SupportStaff}(\mathsf{e006B3F})$.*

In the generalized version of DDL it is also possible to bridge between roles. A practical modeling scenario that makes use of such bridge rules is shown below, by extending our running example.

Example 23. *In order to improve work efficiency, the accounting department has planned a reorganization of offices. Under the assumption that people do more work when they feel fine in the office, the accounting department asked the employees to create a simple friend or a foe ontology $\mathcal{A}_4$. See an excerpt from this ontology:*

$$\mathsf{friendOf}(\mathsf{johnSmith}, \mathsf{robertKay}) \qquad \mathsf{friendOf}(\mathsf{robertKay}, \mathsf{johnSmith})$$
$$\mathsf{foeOf}(\mathsf{robertKay}, \mathsf{stanCoda}) \qquad \mathsf{foeOf}(\mathsf{stanCoda}, \mathsf{markHoffer})$$

The following bridge rules are used to transfer knowledge that is relevant into the ontology of the accounting department:

$$4 : \mathsf{friendOf} \xrightarrow{\sqsubseteq} 3 : \mathsf{goodOfficeMateOf} \qquad 4 : \mathsf{foeOf} \xrightarrow{\sqsubseteq} 3 : \mathsf{dislikes}$$

$$1 : \mathsf{officeMateOf} \xrightarrow{\sqsubseteq} 3 : \mathsf{currentOfficeMateOf}$$

We are especially interested in employees which do not feel comfortable sharing the office with their current office mates, hence we define a new role that will be called unhappyOfficeMateOf *by an axiom in the RBox $\mathcal{R}_3$:*

$$\mathsf{currentOfficeMateOf} \sqcap \mathsf{dislikes} \sqsubseteq \mathsf{unhappyOfficeMateOf}$$

Now, by using individual correspondence axioms, the relation between the individuals representing the same employee in different modules is encoded:

$$1 : \mathsf{johnSmith} \longmapsto 3 : \mathsf{e006B3F} \qquad 4 : \mathsf{johnSmith} \longmapsto 3 : \mathsf{e006B3F}$$

and so on for the other individuals. Making use of all this knowledge, we are finally able to derive in $\mathcal{T}_3$ who of the employees is located in the same office with an unfriendly coworker and we are also able to determine which office mates would be more suitable for such employees.

6.2.3 DDL with Restricted Semantics

While the classic semantics of DDL (Borgida & Serafini 2003) places no restrictions on the domain relation, in this thesis we have also studied DDL semantics in which further conditions are required. We have investigated DDL with injective domain relations in Chap. 4 and DDL with transitive domain relations and DDL with compositionally consistent domain relations in Chap. 5. These restrictions secured certain desired properties of the framework, but as we have seen, sometimes other desired properties were violated. Hypothetically, it is possible to place many different conditions on the domain relation, and derive even more specific versions of DDL. Later on in this chapter we will make use of some of such semantics in order to compare different distributed ontology formalisms and DDL. The following definition formally establishes the considerable semantics.

Definition 90 (Restricted semantics of DDL)**.** *The following restricted semantics of DDL are possible:*

- *under DDL semantics with* injectivity*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j \in I)\ (\forall x, y \in \Delta^{\mathcal{I}_i})\ r_{ij}(x) \cap r_{ij}(y) = \emptyset$*;*
- *under DDL semantics with* functionality*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j \in I)\ (\forall x \in \Delta^{\mathcal{I}_i})\ y_1 \in r_{ij}(x) \wedge y_2 \in r_{ij}(x) \implies y_1 = y_2$*;*
- *under DDL semantics with* totality*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j \in I)\ (\forall x \in \Delta^{\mathcal{I}_i})\ r_{ij}(x) \neq \emptyset$*;*
- *under DDL semantics with* transitivity*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j, k \in I)\ r_{ij} \circ r_{jk} \subseteq r_{ik}$*;*
- *under DDL semantics with* compositional consistency*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j, k \in I)\ r_{ij} \circ r_{jk} = r_{ik}$*;*
- *under DDL semantics with* restricted compositionality*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j, k \in I)\ r_{ij} \circ r_{jk} = r_{ik}$*, if there are two directed paths in* $\mathrm{G}_{\mathfrak{K}}$*, the first from i to j and the second from j to k;*
- *under DDL semantics with* role preservation*, the domain relation r in each distributed model is required to satisfy:* $(\forall i, j \in I)\ (\forall\ role\ R)\ \langle x, y\rangle \in r_{ij} \wedge \langle x, x'\rangle \in R^{\mathcal{I}_i} \implies r_{ij}(x') \neq \emptyset$*.*

All these semantics are stronger than the original semantics and each exhibits some desirable properties. There is trade-off however. The semantics with injectivity beaks directionality and the semantics with compositional consistency breaks both directionality and local inconsistency properties. To our best knowledge, the remaining semantics listed in Definition 90 were not investigated yet. They correspond to restrictions employed by other distributed ontology formalisms such such as P-DL and IDDL (see Sect. 6.4 and 6.5).

6.3 $\mathcal{E}$-connections

The framework of $\mathcal{E}$-connections has been introduced by Kutz et al. (2002) with the motivation of combining different logics each of which represents one aspect of a complex system (Kutz et al. 2004, Kutz 2004). For this reason, $\mathcal{E}$-connections were defined over so called Abstract Description Systems, a common generalization of DL, modal logics, logics of time and space and some other logical formalisms, as introduced by Baader et al. (2002). $\mathcal{E}$-connections pioneered the idea of ontology linking: ontology modules are interconnected with links, which act as inter-ontology properties. As much as roles are properties that associate instances within the same ontology module, links allow to associate instances from two distinct ontology modules. Links are allowed in restrictions and thus they allow the local ontologies to be connected. In contrast with DDL and some other distributed ontology frameworks, links are not intended to represent semantic mappings between ontologies.

A distinctive feature of $\mathcal{E}$-connections, different from other modular ontology frameworks presented in this chapter, is that each ontology module is supposed to model a portion of the domain that is complementary and non-overlapping with respect to the other ontology modules (e.g., the two domains of people and vehicles are non-overlapping). Hence, in $\mathcal{E}$-connections it is not possible to have a concept in some ontology module that has subconcepts or instances in some other ontology module. For further illustration of non-overlapping modeling domains see Example 24 below.

Properties of $\mathcal{E}$-connections over Description Logics have been studied by Kutz et al. (2003), Cuenca Grau et al. (2004*b*,*a*), and Parsia & Cuenca Grau (2005). $\mathcal{E}$-connections of OWL ontologies were also introduced by Cuenca Grau, Parsia & Sirin (2006).

6.3.1 Formalization

The two most commonly explored flavours of $\mathcal{E}$-connections of Description Logics are $\mathcal{C}^{\epsilon}_{\mathcal{HQ}}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ and $\mathcal{C}^{\epsilon}_{\mathcal{HI}}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ (Cuenca Grau et al. 2004*a*,*b*, Cuenca Grau, Parsia & Sirin 2006). These allow $\mathcal{SHIQ}$, $\mathcal{SHOQ}$, and $\mathcal{SHIO}$ local ontologies to be connected with links, the former allowing link hierarchies and qualified number restrictions on links, the latter link hierarchies and inverse links. It has been noted that more general frameworks such as $\mathcal{C}^{\epsilon}_{\mathcal{HIQ}}(\mathcal{SHOIQ})$ and even $\mathcal{C}^{\epsilon}_{\mathcal{HIQ}}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ lead to undesired behaviour such as projection of nominals from one local model to another (Cuenca Grau, Parsia & Sirin 2006, Kutz et al. 2004).

In addition, two extended frameworks named $\mathcal{C}^{\epsilon}_{\mathcal{HQ}+}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ and $\mathcal{C}^{\epsilon}_{\mathcal{HI}+}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ (Parsia & Cuenca Grau 2005, Cuenca Grau et al. 2004*a*) have also been studied. These extend the formalism with several tweaks useful from the modeling perspective: same property name is freely reusable as a role name (within a module) and as well as a link name (between the modules), transitive links are allowed and transitivity of roles and links is now controlled by a new general transitivity axiom which enables switching the transitivity on and off based on the context of modules within/between which the role/link is used. For sake of simplicity we introduce a slightly more general $\mathcal{C}^{\epsilon}_{\mathcal{HIQ}+}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$, of which $\mathcal{C}^{\epsilon}_{\mathcal{HQ}+}(\mathcal{SHIQ},\mathcal{SHOQ},\mathcal{SHIO})$ is the

sublanguage that does not allow inverse links and $\mathcal{C}^{\epsilon}_{\mathcal{HI}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ is the sublanguage that does not allow links in number restrictions.

Definition 91 (Syntax of $\mathcal{C}^{\epsilon}_{\mathcal{HIQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$)**.** *Given a finite index set I, for $i \in I$ let m_i be a constant either equal to $\mathcal{SHIQ}$, $\mathcal{SHOQ}$ or $\mathcal{SHIO}$ that serves to identify the local language of a component knowledge base. For $i \in I$ let $N_{\mathrm{C}i}^{m_i}$ and $N_{\mathrm{I}i}^{m_i}$ be pairwise disjoint sets of concept names and individual names respectively. For $i, j \in I$, i and j not necessarily distinct, let $\epsilon_{ij}^{m_i}$ be sets of properties, not necessarily mutually disjoint, but disjoint with respect to $N_{\mathrm{C}k}^{m_k}$ and $N_{\mathrm{I}k}^{m_k}$ for any $k \in I$, and let ρ_{ij} be sets of ij-properties defined as follows:*

- *if $i = j$ and $m_i = \mathcal{SHOQ}$ then $\rho_{ij} = \epsilon_{ij}^{m_i}$;*
- *otherwise $\rho_{ij} = \epsilon_{ij}^{m_i} \cup \{P^- \mid P \in \epsilon_{ji}^{m_j}\}$;*

Given two properties $P_1, P_2 \in \rho_{ij}$, an ij-property axiom is an assertion of the form $P_1 \sqsubseteq P_2$. A general transitivity axiom is an expression of the form $\mathrm{Trans}(P; (i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n))$, where $i_k \in I$, and $P \in \epsilon_{i_k i_{k+1}}^{m_{i_k}}$, for $1 \leq k < n$. An ij-property box $\mathcal{R}_{ij}$ is a finite set of ij-property axioms. The combined property box $\mathcal{R}$ contains all the property boxes for each $i, j \in I$ (not necessarily distinct) and also all transitivity axioms. Let us denote by $\sqsubseteq^*$ the transitive-reflexive closure on $\sqsubseteq$. A property P is said to be transitive in (i, j), if $\mathrm{Trans}(P; (i_1, i_2), \ldots, (i_{n-1}, i_n)) \in \mathcal{R}$ such that $i_k = i$ and $i_{k+1} = j$, for some $1 \leq k < n$. An ij-property P is called simple if there is no S that is transitive in (i, j) and $S \sqsubseteq^* P$.

Given some $i \in I$ the set of i-concepts is defined inductively, as the smallest set such that:

- *each atomic concept $A \in N_{\mathrm{C}i}^{m_i}$ and two special symbols $\top_i$ and $\bot_i$ are i-concepts;*
- *given i-concepts C and D, a j-concept Z, and $P \in \rho_{ij}$, also $\neg C$, $C \sqcap D$, $C \sqcup D$, $\exists P.Z$, and $\forall P.Z$ are i-concepts;*
- *if $m_i \in \{\mathcal{SHOQ}, \mathcal{SHIO}\}$ and $o \in N_{\mathrm{I}i}$ then $\{o\}$ is an i-concept;*
- *given a j-concept Z, a natural number n, and a simple property $S \in \rho_{ij}$ such that if $i = j$ then $m_i \in \{\mathcal{SHIQ}, \mathcal{SHOQ}\}$, also $\geqslant n\, S.Z$ and $\leqslant n\, S.Z$ are i-concepts*[1]*.*

Given two individuals $a, b \in N_{\mathrm{I}i}$, an i-concept C and some role $R \in \rho_{ii}$, the expression $C(a)$ is called an i-local concept assertion and the expression $R(a, b)$ is called an i-local role assertion. In addition, given two individuals $a \in N_{\mathrm{I}i}$ and $b \in N_{\mathrm{I}j}$ such that $i \neq j$, and some link property $E \in \rho_{ij}$, an object assertion is an axiom of the form:

$$a \cdot E \cdot b \ .$$

A combined TBox is a tuple $\mathcal{K} = \{\mathcal{K}_i\}_{i \in I}$ where each $\mathcal{K}_i$ is a set of i-local GCI each of the form $C \sqsubseteq D$, where C and D are i-concepts. A combined ABox

[1] Note that even if the local language of the i-th module is $\mathcal{SHIO}$ (i.e., $m_i = \mathcal{SHIO}$) for some index $i \in I$, number restrictions may still appear in i-concepts as long as they are specified on link properties (Parsia & Cuenca Grau 2005, Cuenca Grau et al. 2004*a*).

$\mathcal{A} = \{\mathcal{A}_i\}_{i\in I} \cup \mathcal{A}_\mathcal{E}$ is a set containing local ABoxes[2] $\mathcal{A}_i$, each comprising of a finite number of i-local concept and role assertions, and a finite number of object assertions $\phi \in \mathcal{A}_\mathcal{E}$.

Finally, a combined knowledge base $\Sigma = \langle \mathcal{K}, \mathcal{R}, \mathcal{A}\rangle$ is composed of a combined TBox, a combined property box, and a combined ABox, all of them defined over same index set I.

Semantics of $\mathcal{E}$-connections employs so called combined interpretations which are similar to distributed interpretations of DDL. A combined interpretation $\mathcal{I}$ consists of local domains and interpretation functions. Local domains $\Delta^{\mathcal{I}_i}$ are pairwise disjoint and unlike in DDL they are strictly non-empty. There is no distinctive domain relation as in DDL, instead there are special interpretation functions for links which assign to each link $E \in \rho_{ij}$ a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ thus effectively turning them into inter-ontology properties.

Definition 92 (Semantics of $\mathcal{C}^{\epsilon}_{\mathcal{HIQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$)**.** *Let us assume a combined knowledge base $\Sigma = \langle \mathcal{K}, \mathcal{R}, \mathcal{A}\rangle$ with some index set I. A combined interpretation is a triple $\mathcal{I} = \langle \{\Delta^{\mathcal{I}_i}\}_{i\in I}, \{\cdot^{\mathcal{I}_i}\}_{i\in I}, \{\cdot^{\mathcal{I}_{ij}}\}_{i,j\in I}\rangle$, where for each $i \in I$, $\Delta^{\mathcal{I}_i} \neq \emptyset$ and for each $i, j \in I$ such that $i \neq j$ we have $\Delta^{\mathcal{I}_i} \cap \Delta^{\mathcal{I}_j} = \emptyset$. Interpretation functions $\cdot^{\mathcal{I}_i}$ provide denotation for i-concepts and interpretation functions $\cdot^{\mathcal{I}_{ij}}$ are employed for sake of denotation of ij-properties.*

Each ij-property $P \in \rho_{ij}$ is interpreted by $P^{\mathcal{I}_{ij}}$, a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$. If $P \in \rho_{ij}$ is an inverse of another property, say $P = Q^-$, $Q \in \rho_{ji}$, then $P^{\mathcal{I}_{ij}} = \{(x, y) \in \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j} \mid (y, x) \in Q^{\mathcal{I}_{ji}}\}$. A combined interpretation $\mathcal{I}$ satisfies an ij-property axiom $P_1 \sqsubseteq P_2$ (denoted by $\mathcal{I} \models P_1 \sqsubseteq P_2$) if $P_1{}^{\mathcal{I}_{ij}} \subseteq P_2{}^{\mathcal{I}_{ij}}$. It satisfies a transitivity axiom $\phi = \mathrm{Trans}(P; (i_1, i_2), \ldots, (i_{n-1}, i_n))$ (denoted by $\mathcal{I} \models \phi$) if both of the following conditions are true:[3]

1. *for each $1 \leq k < n$, $P^{\mathcal{I}_{i_k i_{k+1}}}$ is a transitive relation;*
2. *for each $1 \leq k < h < n$, $P^{\mathcal{I}_{i_k i_h}} = P^{\mathcal{I}_{i_k i_{k+1}}} \circ \cdots \circ P^{\mathcal{I}_{i_h i_{h+1}}}$.*

A combined interpretation $\mathcal{I}$ satisfies an ij-property box $\mathcal{R}_{ij}$ (denoted by $\mathcal{I} \models \mathcal{R}_{ij}$) if it satisfies all ij-property axioms of $\mathcal{R}_{ij}$; it satisfies a combined property box $\mathcal{R}$ (denoted by $\mathcal{I} \models \mathcal{R}$) if it satisfies each ij-property box and each transitivity axiom contained in $\mathcal{R}$.

Each i-concept C is interpreted by some subset $C^{\mathcal{I}_i}$ of $\Delta^{\mathcal{I}_i}$. For special symbols we have $\top_i = \Delta^{\mathcal{I}_i}$ and $\bot_i = \emptyset$. In addition, denotation of complex i-concepts must satisfy the constraints as given in Table 6.1. Combined interpretation $\mathcal{I}$ satisfies an i-local GCI $C \sqsubseteq D$ (denoted by $\mathcal{I} \models C \sqsubseteq D$) if $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$; it satisfies a local TBox $\mathcal{K}_i$ (denoted $\mathcal{I} \models \mathcal{K}_i$) if it satisfies each i-local GCI thereof;

[2] Local ABoxes are not included in $\mathcal{C}^{\epsilon}_{\mathcal{HQ}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, $\mathcal{C}^{\epsilon}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, $\mathcal{C}^{\epsilon}_{\mathcal{HQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, $\mathcal{C}^{\epsilon}_{\mathcal{HI}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ $\mathcal{E}$-connections as defined by Cuenca Grau et al. (2004*a*,*b*) and Cuenca Grau, Parsia & Sirin (2006) nor as given by Parsia & Cuenca Grau (2005). But they are present in the previous work of Kutz et al. (2003, 2004). Since they do not constitute any problem semantically we add them for sake of comparison. Note also that even if the inter-module object assertions are possibly expressed using the same syntax as role assertions, we favour a syntax similar to the one of Cuenca Grau & Kutz (2007) in order to make a clear distinction here.

[3] This definition of the semantics of general transitivity axioms differs from that originally given by Cuenca Grau et al. (2004*a*) and Parsia & Cuenca Grau (2005), however, we believe that this new definition actually suits the intuition behind the general transitivity axiom as it is explained in these works.

and it satisfies a combined TBox $\mathcal{K}$ over I (denoted $\mathcal{I} \models \mathcal{K}$) if it satisfies $\mathcal{K}_i$ for each $i \in I$.

A combined interpretation $\mathcal{I}$ satisfies an i-local concept assertion $C(a)$ (denoted by $\mathcal{I} \models C(a)$), if $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$; it satisfies an i-local role assertion $R(a,b)$ (denoted by $\mathcal{I} \models R(a,b)$), if $\langle a^{\mathcal{I}_i}, b^{\mathcal{I}_i} \rangle \in R^{\mathcal{I}_{ii}}$; $\mathcal{I}$ satisfies an object assertion $a \cdot E \cdot b$ (denoted by $\mathcal{I} \models a \cdot E \cdot b$), $a \in N_{\mathrm{I}i}$, $b \in N_{\mathrm{I}j}$, $E \in \rho_{ij}$, if $\langle a^{\mathcal{I}_i}, b^{\mathcal{I}_j} \rangle \in E^{\mathcal{I}_{ij}}$. Putting this together, $\mathcal{I}$ satisfies a combined ABox $\mathcal{A}$ (denoted by $\mathcal{I} \models \mathcal{A}$) if it satisfied each ABox assertion of each $\mathcal{A}_i \in \mathcal{A}$ and as well each object assertion contained in $\mathcal{A}$.

Finally, a combined interpretation $\mathcal{I}$ over some index set I is a model of a combined knowledge base $\Sigma = \langle \mathcal{K}, \mathcal{R}, \mathcal{A} \rangle$ defined over I (denoted by $\mathcal{I} \models \Sigma$) if $\mathcal{I}$ satisfies all three $\mathcal{K}$, $\mathcal{R}$ and $\mathcal{A}$.

Construct	Condition
$\neg C$	$(\neg C)^{\mathcal{I}_i} = \Delta^{\mathcal{I}_i} \setminus C^{\mathcal{I}_i}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}_i} = C^{\mathcal{I}_i} \cap D^{\mathcal{I}_i}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}_i} = C^{\mathcal{I}_i} \cup D^{\mathcal{I}_i}$
$\{o\}$	$\{o\}^{\mathcal{I}_i} = \{o^{\mathcal{I}_i}\}$
$\forall P.Z$	$(\forall P.Z)^{\mathcal{I}_i} = \{x \in \Delta^{\mathcal{I}_i} \mid (\forall y \in \Delta^{\mathcal{I}_j})\ (x,y) \in P^{\mathcal{I}_{ij}} \implies y \in Z^{\mathcal{I}_j}\}$
$\exists P.Z$	$(\exists P.Z)^{\mathcal{I}_i} = \{x \in \Delta^{\mathcal{I}_i} \mid (\exists y \in \Delta^{\mathcal{I}_j})\ (x,y) \in P^{\mathcal{I}_{ij}} \wedge y \in Z^{\mathcal{I}_j}\}$
$\geqslant n\, S.Z$	$(\geqslant n\, S.Z)^{\mathcal{I}_i} = \{x \in \Delta^{\mathcal{I}_i} \mid \sharp\{y \in \Delta^{\mathcal{I}_j} \mid (x,y) \in S^{\mathcal{I}_{ij}}\} \geq n\}$
$\leqslant n\, S.Z$	$(\leqslant n\, S.Z)^{\mathcal{I}_i} = \{x \in \Delta^{\mathcal{I}_i} \mid \sharp\{y \in \Delta^{\mathcal{I}_j} \mid (x,y) \in S^{\mathcal{I}_{ij}}\} \leq n\}$

Table 6.1: Semantic constraints for complex i-concepts in $\mathcal{E}$-connections. C and D are i-concepts, Z is a j-concept, P and S are ij-properties (i.e., either roles or links), S is simple

6.3.2 Reasoning with $\mathcal{E}$-connections

A key reasoning task for $\mathcal{E}$-connections is the satisfiability of i-concepts with respect to a combined knowledge base. Other reasoning tasks such as entailment of subsumption between i-concepts are reducible.

Definition 93 (Reasoning tasks for $\mathcal{E}$-connections)**.** *Given a combined knowledge base Σ over some index set I and $i \in I$, an i-concept C is satisfiable with respect to Σ, if there exists a combined interpretation $\mathcal{I}$ of Σ that is a model of Σ and $C^{\mathcal{I}_i} \neq \emptyset$. Given two i-concepts C and D, it is said that Σ entails the subsumption $C \sqsubseteq D$ (denoted $\Sigma \models C \sqsubseteq D$), if in each model $\mathcal{I}$ of Σ we have $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.*

As for regular DL, entailment of subsumption is reducible into concept satisfiability (Kutz et al. 2004, Cuenca Grau et al. 2004*a*).

Theorem 92. *Given a combined knowledge base Σ over some index set I, $i \in I$ and two i-concepts C and D, the subsumption formula $C \sqsubseteq D$ is entailed by Σ if and only if the complex i-concept $C \sqcap \neg D$ is unsatisfiable with respect to Σ.*

Another decision problem that has been studied for $\mathcal{E}$-connections is the problem of entailment of ABox knowledge (Kutz et al. 2004).

Definition 94 (Entailment of ABox formulae)**.** *A combined knowledge base* Σ *entails an* i*-local expression* $C(a)$*, if in each model* $\mathcal{I}$ *of* Σ *we have* $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$*. A combined knowledge base* Σ *entails an* i*-local expression* $R(a,b)$*, if in each model* $\mathcal{I}$ *of* Σ *we have* $\langle a^{\mathcal{I}_i}, b^{\mathcal{I}_i} \rangle \in R^{\mathcal{I}_{ii}}$*.*

Tableaux reasoning algorithms for the $\mathcal{E}$-connections languages $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$, and $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ have been introduced by Cuenca Grau et al. (2004*a*) and Parsia & Cuenca Grau (2005). For each of these algorithms, the complexity of deciding the satisfiability of an i-concept C with respect to a combined knowledge base Σ is 2NExpTime in the size of C and Σ in the worst case. The Pellet reasoner supports $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ and $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ (Cuenca Grau, Parsia & Sirin 2006, Cuenca Grau et al. 2004*a*); Pellet's extension towards $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ and also $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ is planned (Parsia & Cuenca Grau 2005).

6.3.3 Modeling with $\mathcal{E}$-connections

We will now have a closer look at how one models with $\mathcal{E}$-connections. The examples in this section approximate, in part, the modeling scenario we have done with DDL in Sect. 6.2.2. However, this approximation is not intended to be complete, above all our aim is to show the possibilities we have with $\mathcal{E}$-connections. Again, we will model a running example focused on people and their occupation. Since $\mathcal{E}$-connections require strictly split modeling domains, we will first split the modeling domain that we will use.

Example 24. *We will build a combined knowledge base* Σ*, split into four local ontologies. We will use the index set* $I = \{1, 2, 3, 4\}$*. Each local ontology deals with a separated segment of the modeling domain. The first three local ontologies are (index of each ontology in parentheses):*

- People *(*$i = 1$*), containing individuals such as* johnSmith, rickDeckard, *etc., and concepts such as* Man, Woman, Adult, Parent, *etc.;*
- Organizations *(*$i = 2$*), with individuals* googleInc, operaSA, teatroAllaScala, *etc., and concepts such as* Enterprise, Theatre, ITCompany, OperaHouse, NorwegianEnterprise, EuropeanEnterprise, *etc.;*
- Locations *(*$i = 3$*), that will contain concepts such as* Milan, MountainView, Oslo, Norway, Italy, Lesotho, EU, USA, Africa, Europe, *etc.*

These three local domains are clearly separated: people are disjoint from institutions because no individual will ever be an instance of some concept from People *and some other concept from* Organizations*. The same holds for* Locations*.*

The fourth local ontology will be:

- Professions *(*$i = 4$*), that contains concepts* Chef, Engineer, Pilot, Policeman, Researcher, *etc.*

Here we ought to be cautious, however. We need to think twice about how do we really want to use these concepts: given an individual that represents a person, say johnSmith*, that comes from the ontology* People*, do we want this individual to possibly become an instance of one of the concepts in* Professions*? This kind of*

modeling is not possible in $\mathcal{E}$-connections. In fact, as shown below, we will be able to keep professions in a separate local ontology, but under certain restrictions.

Let us now show how links are used in $\mathcal{E}$-connections and how they make propagation of knowledge possible.

Example 25. *Assume that in the* Locations *ontology module we have information such as:*

$$\mathsf{Oslo} \sqsubseteq \mathsf{Norway}\ , \qquad \mathsf{Milan} \sqsubseteq \mathsf{Italy}\ ,$$
$$\mathsf{Norway} \sqsubseteq \mathsf{Europe}\ , \qquad \mathsf{Lesotho} \sqsubseteq \mathsf{Africa}\ ,$$

and so on. Now, using the link $\mathsf{locatedIn} \in \epsilon_{23}^{m_2}$ *we introduce two complex concepts within the* Organizations*:*

$$\mathsf{NorwegianEnterprise} \doteq \mathsf{Enterprise} \sqcap \exists \mathsf{locatedIn}.\mathsf{Norway}\ ,$$
$$\mathsf{EuropeanEnterprise} \doteq \mathsf{Enterprise} \sqcap \exists \mathsf{locatedIn}.\mathsf{Europe}\ .$$

It is implied that $\mathsf{NorwegianEnterprise} \sqsubseteq \mathsf{EuropeanEnterprise}$ *by the semantics of $\mathcal{E}$-connections. If $x \in \Delta^{\mathcal{I}_2}$ is an instance of* $\mathsf{Enterprise} \sqcap \exists \mathsf{locatedIn}.\mathsf{Norway}$*, we know from the semantics that it is an instance of* Enterprise *and it has a* locatedIn*-successor $y \in \Delta^{\mathcal{I}_3}$ that is an instance of* Norway*. However, due to the axiom* $\mathsf{Norway} \sqsubseteq \mathsf{Europe}$*, we know that y is also an instance of* Europe *and hence x is an instance of* $\mathsf{Enterprise} \sqcap \exists \mathsf{locatedIn}.\mathsf{Europe}$*.*

We will now continue the running example and demonstrate modeling with individuals.

Example 26. *Assume the* Organizations *module contains the following knowledge:*

$$\mathsf{ITCompany} \sqsubseteq \mathsf{Enterprise}\ , \qquad \mathsf{ITCompany}(\mathsf{operaSA})\ ,$$
$$\mathsf{OperaHouse} \sqsubseteq \mathsf{Theatre}\ , \qquad \mathsf{ITCompany}(\mathsf{googleInc})\ .$$

To relate individuals from different ontologies, we use object assertions. We will use the link worksAt *between the* People *ontology module and the* Organizations *module to indicate employment. Similarly, we will use the link* locatedIn *between* Organizations *and* Locations *to indicate where the companies are located:*

$$\mathsf{johnSmith} \cdot \mathsf{worksAt} \cdot \mathsf{googleInc}\ , \qquad \mathsf{googleInc} \cdot \mathsf{locatedIn} \cdot \mathsf{MountainView}\ ,$$
$$\mathsf{rickDeckard} \cdot \mathsf{worksAt} \cdot \mathsf{operaSA}\ , \qquad \mathsf{operaSA} \cdot \mathsf{locatedIn} \cdot \mathsf{Oslo}\ .$$

In addition, we employ links in restrictions in order to derive new knowledge within People *in form of complex concepts, for instance by including the following GCI axiom:*

$$\exists \mathsf{worksAt}.\mathsf{Enterprise} \sqsubseteq \mathsf{IndustryEmployee}\ .$$

Using the $\mathcal{E}$-connections semantics, one is able to derive that both johnSmith *and* rickDeckard *are in fact industry employees. The proof is very similar to the previous example.*

As mentioned above, $\mathcal{E}$-connections require strict separation of modeling domains between modules. Hence, if we want to introduce a new local ontology Professions, that will contain concepts such as Chef or Pilot, this comes at some cost. Particularly, we will not be able to assert in the ABox of People that some of its individuals belongs to any of the concepts of Professions. Also, we will not be able to relate directly concepts such as Adult of People with say Policeman of Professions by means of subsumption. In part these issues are overcome by using links instead of class membership, as we show in the following example. Even if inter-ontology class membership is ruled out by this approach, in many cases it yields satisfactory modeling.

Example 27. *Let us now include a new local ontology* Professions *containing concepts such as* Chef, Pilot, Researcher, Policeman, *etc. In order to express that some people belong to certain profession we use the link* hasProfession *between* People *and* Professions*:*

$$\mathsf{johnSmith} \cdot \mathsf{hasProfession} \cdot \mathsf{Chef} \ ,$$
$$\mathsf{rickDeckard} \cdot \mathsf{hasProfession} \cdot \mathsf{Researcher} \ .$$

The two more expressive $\mathcal{E}$-connections frameworks $\mathcal{C}^{\mathcal{E}}_{\mathcal{HQ}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ and $\mathcal{C}^{\mathcal{E}}_{\mathcal{HI}+}(\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHIO})$ also include the general transitivity axiom (Cuenca Grau et al. 2004*a*, Parsia & Cuenca Grau 2005). Taking advantage of the fact that names are freely allowed to be reused for role and link properties, this allows for flexible transitivity control. In order to demonstrate this feature, we borrow an example that originally appears in these works.

Example 28. *Let us add some more knowledge into our running example. First, we add to* Organizations *(indexed by* 2*) further structure for the Google company:*

$$\mathsf{partOf}(\mathsf{gCateringDpt}, \mathsf{gEmployeeServicesDiv}) \ ,$$
$$\mathsf{partOf}(\mathsf{gEmployeeServicesDiv}, \mathsf{googleInc}) \ .$$

In addition, for whatever reason we decide to also include knowledge about human body parts. In accordance, we add the following axiom into People *(indexed by* 1*):*

$$\mathsf{partOf}(\mathsf{finger47}, \mathsf{johnSmith}) \ .$$

Finally, we are also free to use the property partOf *as a link, and so we add the following object assertion between these two component ontologies:*

$$\mathsf{johnSmith} \cdot \mathsf{partOf} \cdot \mathsf{gCateringDpt} \ .$$

With previous flavours of $\mathcal{E}$-connections we would need to use a separate role name for each component ontology and for each linkbox between each two component modules. In this case we would be forced to use three names.

The real utility of this feature only comes in combination with general transitivity axioms. Before these axioms were introduced, $\mathcal{E}$-connections only included the possibility to declare transitive roles inside each component RBox.

In addition to the explicitly stated knowledge, this would allow us to derive $\mathcal{K} \models \mathsf{partOf}(\mathsf{gCateringDpt}, \mathsf{googleInc})$ *in our case, if we have previously declared the role* partOf *transitive in the* Organizations *component, but not anything more. The new general transitivity axioms allow us to assert transitivity on the combined relation resulting into the composition of all* partOf *properties that are involved, whether roles or links, e.g., by the following axiom:*

$$\mathrm{Trans}(\mathsf{partOf}; (1,1), (1,2), (2,2)) \ .$$

By including this axiom in $\mathcal{K}$ *we derive that* $\mathcal{K} \models \mathsf{johnSmith} \cdot \mathsf{partOf} \cdot \mathsf{googleInc}$ *which is intuitively justified but as well* $\mathcal{K} \models \mathsf{finger47} \cdot \mathsf{partOf} \cdot \mathsf{googleInc}$ *which may not be justified for each modeling scenario. However, the transitivity axiom is versatile enough, and we easily rule out the second consequence, if undesired, by replacing the above transitivity axiom by:*

$$\mathrm{Trans}(\mathsf{partOf}; (1,2), (2,2)) \ .$$

$\mathcal{E}$-connections allow to connect several ontologies under the assumption that their modeling domains are separated. Knowledge from one local ontology is reused in other ontologies thanks to use of link properties. Such a framework is of demand and is easily applicable if one is about to build a new ontology in a modular way, and from the very beginning clearly perceives how to split the modeling domain into modules. Highly developed reasoning algorithms and readily available reasoning engines make $\mathcal{E}$-connections also available for practical use. A tool that allows to automatically partition a monolithic ontology is also available (Cuenca Grau, Parsia, Sirin & Kalyanpur 2006). On the other hand, the separate domains assumption is a serious obstacle if one wishes to combine several already existing ontologies. In such a case we consider the assumption of separate domains too strict, as with fair probability the modeling domains covered by these existing ontologies overlap to some extent.

6.4 Package-based Description Logics

Package-based Description Logics (P-DL) were originally introduced by Bao & Honavar (2004) under the name Package-extended Ontologies. Later the framework was developed by Bao et al. (2006*d*,*a*,*c*, 2009). P-DL fall under the paradigm of ontology imports. The semantics of ontology imports has also been investigated by Pan et al. (2006). Within this framework, a modular ontology is composed of several packages. Each ontology term (namely, each individual, concept and role) is assigned its home package – the package it primarily belongs to. Given a package P_i, besides for its home terms, also other terms are free to appear in P_i, we say that they are imported into P_i. When terms are imported, also the semantic relations that exit amongst these terms in their home package are imported.

P-DL also include several other modularization features inspired by modular software engineering (Bao et al. 2006*d*). Package nesting is possible, by virtue of which packages are organized into a hierarchy. The package hierarchy in a P-DL ontology constitutes an organisational structure, in contrast to the semantic structure imposed by ontology axioms and importing. In addition P-DL include so called scope limitation modifiers, which allow ontology engineers to limit

the visibility of certain terms from the perspective of packages other than their home package. Most typical examples of these are the three predefined modifiers public, protected and private, however P-DL allow users to introduce their own modifiers.

6.4.1 Formalization

We now proceed by introducing the P-DL framework formally. The review in this section is based on the most recent publication of Bao et al. to date (Bao et al. 2009), where P-DL are built over the local language $\mathcal{SHOIQ}$ resulting into the P-DL language $\mathcal{SHOIQP}$.

Definition 95 (Syntax of $\mathcal{SHOIQP}$ PD-L)**.** *A package based ontology is any $\mathcal{SHOIQ}$ ontology $\mathcal{P}$, which partitions into a finite set of packages $\{P_i\}_{i\in I}$, where I is some finite index set. Each P_i uses its own alphabet of terms $N_{\mathrm{C}i} \uplus N_{\mathrm{R}i} \uplus N_{\mathrm{I}i}$ (concept, role and individual names, respectively). The alphabets are not mutually disjoint, but for any term t there is a unique home package of t, denoted by* $\mathrm{home}(t)$*. In addition:*

- *the set of home terms of a package $P_i \in \mathcal{P}$ is denoted by Δ_{S_i};*
- *term t occurring in P_i is a local term in P_i if* $\mathrm{home}(t) = P_i$*, otherwise it is said to be a foreign term in P_i;*
- *if t is used a as foreign term in P_j and its home package is P_i, then we say that t is imported into P_j from P_i and we indicate it by $P_i \xrightarrow{t} P_j$;*
- *$P_i \to P_j$ means that $P_i \xrightarrow{t} P_j$ for some t;*
- *$P_i \xrightarrow{*} P_j$ if $P_i = P_{i_1} \to P_{i_2} \ldots P_{i_{n-1}} \to P_{i_n} = P_j$ for some $i_1, \ldots, i_n \in I$ and $n > 1$ (i.e., the relation $\xrightarrow{*}$ is a transitive closure of $\to$);*
- *$P_j^* = P_j \cup \{P_i | P_i \xrightarrow{*} P_j\}$.*

When a package P_j imports another package P_i, it imports also the domain of P_i. In order to denote in P_j the domain of an imported package P_i, the language of P-DL introduces the concept symbol $\top_i$. Within the local model of P_j, the denotation of $\top_i$ represents the domain of P_i. P-DL also introduce so called contextualized negation $\neg_i$, which is a concept constructor applicable within the package P_i and the packages that import P_i. When it occurs in P_i, $\neg_i$ stands for normal concept complement. However, if $\neg_i C$ occurs in P_j, it is the complement of C with respect to the domain of P_i as imported into P_j (i.e., within P_j we have $\neg_i C \equiv \top_i \sqcap \neg_j C$).

The P-DL semantics is reminiscent of the DDL semantics in that, sense that it also relies on a domain relation which projects the interpretation of terms from one package to another. Unlike to the original DDL semantics, strict constraints are placed on the domain relation in P-DL.

Definition 96 (Semantics of $\mathcal{SHOIQP}$ P-DL)**.** *Given a package-based ontology $\mathcal{P} = \{P_i\}_{i\in I}$, a distributed interpretation of $\mathcal{P}$ is a pair $\mathcal{I} = \left\langle \{\mathcal{I}_i\}_{i\in I}, \{r_{ij}\}_{P_i \xrightarrow{*} P_j} \right\rangle$ such that each $\mathcal{I}_i = \left\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i} \right\rangle$ is an interpretation of the local package P_i and each $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ is a domain relation between $\Delta^{\mathcal{I}_i}$ and $\Delta^{\mathcal{I}_j}$. A distributed interpretation $\mathcal{I}$ is a model of $\{P_i\}_{i\in I}$ if the following conditions hold:*

1. *there is at least one* $i \in I$ *such that* $\Delta^{\mathcal{I}_i} \neq \emptyset$;
2. $\mathcal{I}_i \models P_i$, *for each* $i \in I$;
3. r_{ij} *is an injective partial function, for any* $i, j \in I$ *such that* $i \neq j$, *and* r_{ii} *is the identity function on* $\Delta^{\mathcal{I}_i}$, *for each* $i \in I$;
4. *the domain relation is compositionally consistent, that is, if* $P_i \xrightarrow{*} P_j$ *and* $P_j \xrightarrow{*} P_k$, *then* $r_{ik} = r_{ij} \circ r_{jk}$, *for any* $i, j, k \in I$;
5. *if* $P_i \xrightarrow{t} P_j$, *then* $r_{ij}(t^{\mathcal{I}_i}) = t^{\mathcal{I}_j}$, *for any* $i, j \in I$ *and for any term* t;
6. *the domain relation is role preserving, that is, if* $P_i \xrightarrow{R} P_j$ *and* $(x, y) \in R^{\mathcal{I}_i}$, *then* $r_{ij}(x) \neq \emptyset \implies r_{ij}(y) \neq \emptyset$, *for any* $i, j \in I$, $i \neq j$, *and for any role* R.

In a nutshell, the P-DL semantics is characterized as follows. If two terms t_1 and t_2 appear within some package P_i and they are related by means of axioms in P_i, then this relation is propagated into any other package P_j where both these terms also appear, as long as P_j directly or indirectly imports P_i.

6.4.2 Reasoning with P-DL

The three main reasoning tasks for P-DL are consistency of knowledge bases, concept satisfiability and concept subsumption entailment with respect to a knowledge base. These decision problems are always defined with respect to a so called witness package P_w of the knowledge base $\mathcal{P}$. When answering the decision problems we only care about the importing closure P_w^*, the remaining packages do not influence the answer. Given two different packages P_1 and P_2, the answer to these decision problems may be different when witnessed by P_1 as it is when witnessed by P_2.

Definition 97 (Reasoning tasks for P-DL)**.** *A package-based ontology* $\mathcal{P}$ *is consistent as witnessed by a package* P_w *of* $\mathcal{P}$ *(we also say that* $\mathcal{P}$ *is w-consistent), if there exists a model* $\mathcal{I}$ *of* P_w^* *such that* $\Delta^{\mathcal{I}_w} \neq \emptyset$.

A concept C *is satisfiable as witnessed by a package* P_w *of* $\mathcal{P}$ *(we also say that* C *is w-satisfiable with respect to* $\mathcal{P}$*), if there exists a model* $\mathcal{I}$ *of* P_w^* *such that* $C^{\mathcal{I}_w} \neq \emptyset$.

A subsumption formula $C \sqsubseteq D$ *is valid as witnessed by a package* P_w *of* $\mathcal{P}$ *(denoted by* $\mathcal{P} \models C \sqsubseteq_w D$*; we also say that* $C \sqsubseteq D$ *is w-entailed by* $\mathcal{P}$*, if for every model* $\mathcal{I}$ *of* P_w^* *we have* $C^{\mathcal{I}_w} \subseteq D^{\mathcal{I}_w}$.

While not formally introduced in by Bao et al. (2009), we analogously define also ABox formulae entailment, as follows.

Definition 98 (ABox reasoning for P-DL)**.** *A formula* $C(a)$ *is valid as witnessed by a package* P_w *of* $\mathcal{P}$*, if for every model* $\mathcal{I}$ *of* P_w^* *we have* $a^{\mathcal{I}_w} \in C^{\mathcal{I}_w}$. *A formula* $R(a, b)$ *is valid as witnessed by a package* P_w *of* $\mathcal{P}$*, if for every model* $\mathcal{I}$ *of* P_w^* *we have* $\langle a^{\mathcal{I}_w}, b^{\mathcal{I}_w} \rangle \in R^{\mathcal{I}_w}$.

A distributed reasoning algorithm for the P-DL language $\mathcal{ALCP}_{\mathcal{C}}$, a sublanguage of $\mathcal{SHOIQP}$ which is built on top of the local DL $\mathcal{ALC}$ and only allows for importing of concepts, has been given by Bao et al. (2006*c*). To our best

knowledge, no implementation is known. The decidability and the computational complexity of reasoning for the full expressive P-DL language $\mathcal{SHOIQP}$ are shown via a reduction of a package-based ontology into a standard DL knowledge base. $\mathcal{SHOIQP}$ is decidable and the computational complexity for the decision problems is NExpTime-complete (Bao et al. 2009).

6.4.3 Modeling with P-DL

The $\mathcal{E}$-connections users will find P-DL familiar. In fact, one is able to model in P-DL in a very similar fashion. In order to illustrate this, we remodel Example 26 in P-DL.

Example 29. *Let us build a package based ontology* $\{P_1, P_2, P_3\}$ *where* P_1 *will contain knowledge about people,* P_2 *about organizations, and* P_3 *about locations. Let us fix the home terms for these packages as follows:* $\Delta_{S_1} = \{$Person, Child, Adult, Man, Woman, IndustryEmployee, johnSmith, rickDeckard, enderWiggin, hasChild, worksAt$\}$, $\Delta_{S_2} = \{$ITCompany, Enterprise, Theatre, googleInc, operaSA, locatedIn$\}$, $\Delta_{S_3} = \{$Oslo, Milan, MountainView, Norway, Italy, EU, USA$\}$.

It shows that we are able to mimic the conceptual modeling that we have done in case of $\mathcal{E}$*-connections. In this respect, we put into* P_1 *the following axioms:*

$$\begin{aligned} \mathsf{Child} &\sqsubseteq \neg\mathsf{Adult} \ , \\ \mathsf{Woman} &\sqsubseteq \neg\mathsf{Man} \ , \\ \mathsf{Parent} &\sqsubseteq \exists\mathsf{hasChild}.\mathsf{Person} \ , \\ \exists\mathsf{worksAt}.\mathsf{Enterprise} &\sqsubseteq \mathsf{IndustryEmployee} \ . \end{aligned}$$

Since worksAt *now turns into a regular role with its home package being* P_1*, role assertions are used to express where our* P_1*-individuals work. In accordance we add the following axioms into* P_1*:*

$$\mathsf{worksAt}(\mathsf{johnSmith}, \mathsf{googleInc}) \ , \qquad \mathsf{worksAt}(\mathsf{rickDeckard}, \mathsf{operaSA}) \ .$$

In the TBox of P_2 *we put:*

$$\mathsf{ITCompany} \sqsubseteq \mathsf{Enterprise} \ .$$

And in the ABox of P_2 *we put:*

$$\begin{aligned} &\mathsf{ITCompany}(\mathsf{operaSA}) \ , && \mathsf{locatedIn}(\mathsf{operaSA}, \mathsf{Oslo}) \ , \\ &\mathsf{ITCompany}(\mathsf{googleInc}) \ , && \mathsf{locatedIn}(\mathsf{googleInc}, \mathsf{MountainView}) \ . \end{aligned}$$

Altogether we have done the following imports:

$$\begin{aligned} &2 \xrightarrow{\mathsf{operaSA}} 1 \ , && 3 \xrightarrow{\mathsf{Oslo}} 2 \ , \\ &2 \xrightarrow{\mathsf{googleInc}} 1 \ , && 3 \xrightarrow{\mathsf{MountainView}} 2 \ , \\ &2 \xrightarrow{\mathsf{Enterprise}} 1 \ . \end{aligned}$$

Note that in the last axiom of P_1 *we have build the* 1*-concept* $\exists$worksAt.Enterprise *in a very similar fashion as we would do in* $\mathcal{E}$*-connections. The only foreign term that comes into play here is* Enterprise *and the local* 1*-role serves as* worksAt *instead of a link that we would use in* $\mathcal{E}$*-connections.*

We will now continue with the running example and show how the semantics of importing works and how knowledge propagates from one ontology module to another.

Example 30. *Given the knowledge base and the importing relation of the previous example, we will show that the P-DL semantics implies that the individuals* johnSmith *and* rickDeckard *are instances of the concept* IndustryEmployee*, as witnessed by the package P_1. This does not follow from P_1 directly, distributed reasoning is required.*

First we need to realize that both, the individual googleInc *and the concept* Enterprise*, are imported to P_1 from P_2. In P_2,* googleInc *is an instance of the concept* ITCompany *and hence also of its superconcept* Enterprise*. The concept* ITCompany *is not imported to P_1, however the relation between* googleInc *and* Enterprise *is. This works as follows: if the individual* googleInc *exists in the interpretation domain of the package where it is imported, that is, there is some $x = \mathsf{googleInc}^{\mathcal{I}_1} \in \Delta^{\mathcal{I}_1}$, then it also exists in its home package, and so we have some $y = \mathsf{googleInc}^{\mathcal{I}_2} \in \Delta^{\mathcal{I}_2}$. Moreover, $x = r_{12}(y)$. Now the semantics directly implies that for each concept imported from 2 to 1 of which y is an instance in $\mathcal{I}_2$, x must be an instance in $\mathcal{I}_1$. And hence $x \in \mathsf{Enterprise}^{\mathcal{I}_1}$.*

The rest is local reasoning in P_1. Let us consider johnSmith*. We know, that in every model* johnSmith *has* googleInc *as a* worksAt*-successor and that* googleInc *is an instance of* Enterprise*. However, then* johnSmith *is an instance of* IndustryEmployee *which is defined exactly as a set of individuals with at least one such successor. The case of* rickDeckard *is analogous.*

P-DL offer more freedom in modeling however. Let us now extend Example 29 in order to demonstrate this. We will introduce a new package P_4 that will cover the domain of professions. However, even if we have decided to split the modeling domain into the domain of persons in P_1 and the domain of professions in P_4, we will still be able to relate the concepts Policeman and Adult by a GCI, even if each of them belongs to a different home package. Moreover, we will be allowed to assert membership of individuals of P_1 in concepts imported from P_4. Recall from our conclusion from Sect. 6.3, that such modeling is not possible in $\mathcal{E}$-connections.

Example 31. *Let us now enrich the P-DL knowledge base from Example 29 with new package P_4 that will deal with professions. The home terms of P_4 are $\Delta_{S_4} = \{\mathsf{Chef}, \mathsf{Pilot}, \mathsf{Researcher}, \mathsf{Policeman}\}$.*

We record conceptual knowledge about professions in P_4, such as, for instance, for some professions only adults are eligible. For this reason we import the concept Adult *from P_1. Hence, in the TBox of P_4, we have:*

$$\mathsf{Policeman} \sqsubseteq \mathsf{Adult} \ , \qquad \mathsf{Pilot} \sqsubseteq \mathsf{Adult} \ .$$

And, in the ABox of P_1, we indicate:

$$\mathsf{Pilot}(\mathsf{enderWiggin}) \ , \qquad \mathsf{Chef}(\mathsf{johnSmith}) \ ,$$
$$\mathsf{Child}(\mathsf{enderWiggin}) \ .$$

Altogether, the importing relation is enriched by the following imports:

$$4 \xrightarrow{\mathsf{Chef}} 1 \ , \qquad 1 \xrightarrow{\mathsf{Adult}} 4 \ ,$$
$$4 \xrightarrow{\mathsf{Pilot}} 1 \ .$$

Expectedly, the P-DL semantics renders the knowledge base inconsistent, witnessed by the package P_1. This is because the individual enderWiggin *is an instance of both* Child *and* Pilot*, the second concept being imported from P_4. The semantics of importing assures that also in P_1 the imported concept* Pilot *is a subconcept of* Adult*; this subsumption relation is imported from P_4. This in turn implies that the individual* enderWiggin *is also an instance of* Adult*, a concept disjoint with* Child *to which* enderWiggin *must belong because it was explicitly asserted in the ABox of P_1.*

As we have seen, the prominent feature of P-DL is term importing which allows us to reuse in any local ontology also terms defined in some other part of the system together with the knowledge associated with these terms. This makes P-DL an attractive formalism for modeling distributed ontologies. The reasoning algorithm is, however, known only for a rather limited case of package-based $\mathcal{ALC}$ with only concept imports allowed. Also, no implementation of a P-DL reasoner is available, a serious obstacle towards its practical use.

6.5 Integrated Distributed Description Logics

Integrated Distributed Description Logics (IDDL) have been introduced by Zimmermann (2007) with the main motivation of overcoming some of the limitations of the already existing formalisms for ontology mapping, with special focus on reasoning about mappings and mapping composition. Thus, IDDL fall under the ontology mapping paradigm. The basic intuition of IDDL is to represent mappings in a separate logical language, on which it is possible to define a calculus. A distinguishing feature is that IDDL allow us to determine logical consequence between mappings.

One important motivation lies in the observation that in DDL, $\mathcal{E}$-connections and also in P-DL, the two ontologies between which knowledge is transferred are treated differently. There are always two perspectives of the source/foreign/home and the target/local/importing ontology, and these two perspectives are different. In the vision of IDDL, in contrast with this observation, mapping represents semantic relations between entities from different ontologies that are stated from an external perspective, and so it is not distinguished between the source and the target ontology. This approach is further justified by the fact that most of the currently available ontology matching tools (Euzenat & Shvaiko 2007) typically produce mappings which are bidirectional.

Such an approach is reminiscent of peer-to-peer information integration, as introduced by Ullman (1997) and further formalized by Calvanese et al. (2004), in which mappings are represented as implication formulae between the two data sources that are to be integrated.

6.5.1 Formalization

Theoretically, IDDL are build on top of an underlying DL that extends $\mathcal{ALC}$ with qualified number restrictions, nominals, role complement, union, intersection and composition, inverse roles and the transitive-reflexive closure role constructor (Zimmermann 2007, Zimmermann & Le Duc 2008). This language extends both $\mathcal{SROIQ}$ and $\mathcal{ALCQI}b$, introduced in Sect. 2.5 with composition of roles and role transitive-reflexive closure constructors. Such an expressive DL is

known to be undecidable (Baader & Sattler 1996). It has been shown, however, that reasoning in IDDL is decidable as long as each local knowledge base uses a sublanguage which is decidable (Zimmermann & Le Duc 2008).

Definition 99 (Syntax of IDDL). *An IDDL knowledge base, called distributed system (DS), is a pair* $\langle \mathbf{O}, \mathbf{A} \rangle$ *where* $\mathbf{O} = \{O_i\}_{i \in I}$ *is a set of DL ontologies and* $\mathbf{A} = \{A_{ij}\}_{i,j \in I}$ *is a family of alignments, each* A_{ij} *consisting of correspondence axioms of the following six possible forms:*

$$i : C \xleftarrow{\sqsubseteq} j : D \ , \qquad i : R \xleftarrow{\sqsubseteq} j : S \ ,$$
$$i : C \xleftarrow{\bot} j : D \ , \qquad i : R \xleftarrow{\bot} j : S \ ,$$
$$i : a \xleftarrow{\in} j : C \ , \qquad i : a \xleftarrow{=} j : b \ ,$$

where C, D *are concepts,* R, S *are roles,* a, b *are individuals, and* $k : \phi$ *means that the expression* ϕ *belongs to the local ontology* O_i *and conforms to its local language. From left to right and from top to bottom the above correspondence axioms are called cross-ontology concept subsumption, cross-ontology role subsumption, cross-ontology concept disjointness, cross-ontology role disjointness, cross-ontology membership and cross-ontology identity.*

In order to simplify the notation we add a syntactic shorthand $i : X \xleftarrow{\equiv} j : Y$, representing the pair of cross-ontology subsumptions $i : X \xleftarrow{\sqsubseteq} j : Y$, $j : Y \xleftarrow{\sqsubseteq} i : X$, where X, Y are either both concepts or both roles.

The semantics of IDDL works with a separate local domain and a local interpretation for each of the ontologies in the distributed system, however it adds one additional domain, called equalizing domain. The mapping is not interpreted by domain relation between the local ontologies, instead it is interpreted by so called equalizing functions ε_i that project each of the local domains into the equalizing domain.

Definition 100 (Semantics of IDDL). *Given a distributed system* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *over an index set* I, *a distributed interpretation of* S *is a pair* $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$, *where* $\mathbf{I} = \{\mathcal{I}_i\}_{i \in I}$ *is a family of local interpretations such that each* $\mathcal{I}_i = \left\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}} \right\rangle$ *is an interpretation of* $\mathcal{O}_i$ *in its own language with a non-empty local domain* $\Delta^{\mathcal{I}_i}$, *and the equalizing function* $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i \in I}$ *is a family of functions* $\varepsilon_i : \Delta^{\mathcal{I}_i} \to \Delta_\varepsilon$ *that map all elements of each local domain* $\Delta^{\mathcal{I}_i}$ *into a single global domain* Δ_ε[4].

A distributed interpretation $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ *satisfies a local axiom* $i : \phi \in O_i$ *(denoted by* $\mathcal{I} \models_{\mathrm{d}} i : \phi$*), if* ϕ *is satisfied by* $\mathcal{I}_i$ *in accordance to its local DL language. A local ontology* $\mathcal{O}_i$ *is satisfied by* $\mathcal{I}$ *(denoted* $\mathcal{I} \models_{\mathrm{d}} O_i$*), if each its axiom* $i : \phi \in O_i$ *is satisfied by* $\mathcal{I}$. *A correspondence axiom* ψ *is satisfied by* $\mathcal{I}$ *depending on its type as follows:*

1. $\mathcal{I} \models_{\mathrm{d}} i : C \xleftarrow{\sqsubseteq} j : D$, *if* $\varepsilon_i\left(C^{\mathcal{I}_i}\right) \subseteq \varepsilon_j\left(D^{\mathcal{I}_j}\right)$;
2. $\mathcal{I} \models_{\mathrm{d}} i : R \xleftarrow{\sqsubseteq} j : S$, *if* $\varepsilon_i\left(R^{\mathcal{I}_i}\right) \subseteq \varepsilon_j\left(S^{\mathcal{I}_j}\right)$;
3. $\mathcal{I} \models_{\mathrm{d}} i : C \xleftarrow{\bot} j : D$, *if* $\varepsilon_i\left(C^{\mathcal{I}_i}\right) \cap \varepsilon_j\left(D^{\mathcal{I}_j}\right) = \emptyset$;

[4] Note that also Δ_ε is nonempty, thanks to the totality of the equalizing functions and the fact that $\Delta^{\mathcal{I}_i}$ is nonempty for each $i \in I$.

4. $\mathcal{I} \models_{\mathrm{d}} i : R \xleftrightarrow{\perp} j : S$, *if* $\varepsilon_i\left(R^{\mathcal{I}_i}\right) \cap \varepsilon_j\left(S^{\mathcal{I}_j}\right) = \emptyset$;

5. $\mathcal{I} \models_{\mathrm{d}} i : a \xleftrightarrow{\in} j : C$, *if* $\varepsilon_i\left(a^{\mathcal{I}_i}\right) \in \varepsilon_j\left(D^{\mathcal{I}_j}\right)$;

6. $\mathcal{I} \models_{\mathrm{d}} i : a \xleftrightarrow{=} j : b$, *if* $\varepsilon_i\left(a^{\mathcal{I}_i}\right) = \varepsilon_j\left(b^{\mathcal{I}_j}\right)$.

An alignment A_{ij} is satisfied by $\mathcal{I}$ (denoted $\mathcal{I} \models_{\mathrm{d}} A_{ij}$), if for each correspondence axiom $\psi \in A_{ij}$ we have $\mathcal{I} \models_{\mathrm{d}} \psi$.

Given an index set I, a distributed system $S = \langle \mathbf{O}, \mathbf{A} \rangle$ and a distributed interpretation $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ both over I, we say that $\mathcal{I}$ is a model of S (denoted by $\mathcal{I} \models_{\mathrm{d}} S$), if for each $i \in I$ we have $\mathcal{I} \models_{\mathrm{d}} O_i$ and for each pair $i, j \in I$ we have $\mathcal{I} \models_{\mathrm{d}} A_{ij}$.

6.5.2 Reasoning with IDDL

The main reasoning task described for IDDL is consistency checking for distributed systems. Since the definition of a consistent distributed system is not given by Zimmermann (2007) nor by Zimmermann & Le Duc (2008), we assume the usual definition, which is as follows.

Definition 101 (DS consistency in IDDL)**.** *A distributed system S is consistent if there exists a distributed interpretation $\mathcal{I}$ that is a model of S.*

The other reasoning tasks for IDDL are local formula entailment and also correspondence formula entailment.

Definition 102 (Other reasoning tasks in IDDL)**.** *A formula ϕ (either a local formula or a correspondence formula) is entailed by S (denoted $S \models_{\mathrm{d}} \phi$), if for each distributed interpretation $\mathcal{I}$ that is a model of S it holds that $\mathcal{I} \models_{\mathrm{d}} \phi$.*

All the other reasoning tasks are reducible into distributed system consistency checking (Zimmermann & Le Duc 2008). A sound and complete algorithm for consistency checking of IDDL distributed systems with alignments limited to cross-ontology concept subsumption, disjointness and role subsumption has been given by Zimmermann & Le Duc (2008). This reasoning algorithm is based on a reduction of a distributed system into a set of independent ontologies which are all consistent if and only if the original distributed system is consistent. Worst-case complexity of this procedure is 3ExpTime (Zimmermann & Le Duc 2008).

An attempt to provide an axiomatization of the effects of the correspondences in terms of propagation rules has been presented by Zimmermann (2007). The completeness of these rules is still an open problem.

6.5.3 Modeling with IDDL

IDDL are particularly intended for reasoning about ontology mapping. In IDDL, the mapping is not primarily viewed as a mean of knowledge transfer between ontologies, instead it is viewed as an integrating element that forms the global semantics of a distributed system on top of the local semantics of each of the components thereof. As such, the mapping is bidirectional – a unique feature amongst the distributed and modular ontology frameworks that we have encountered so far. This fact poses some implications on the way how one models

with IDDL. In order to demonstrate this, we remodel, in part, the scenario created in Examples 21–23.

Example 32. *In our example company, three ontologies are used. The first one is O_1, the ontology of employees. An excerpt:*

$$\begin{aligned} \mathsf{ITStaff} &\sqsubseteq \mathsf{Employee} \ , & \mathsf{SupportStaff} &\sqsubseteq \mathsf{Employee} \ , \\ \mathsf{CateringStaff} &\sqsubseteq \mathsf{SupportStaff} \ , & \mathsf{AdministrativeStaff} &\sqsubseteq \mathsf{SupportStaff} \ , \\ \mathsf{Cook} &\sqsubseteq \mathsf{CateringStaff} \ , & \mathsf{Chef} &\sqsubseteq \mathsf{Cook} \ . \end{aligned}$$

There is also O_2, the brigade de cuisine based ontology of different responsibilities in the kitchen. An excerpt:

$$\begin{aligned} \mathsf{KitchenChef} &\sqsubseteq \mathsf{Cook} \ , & \mathsf{Saucemaker} &\sqsubseteq \mathsf{Cook} \ , \\ \mathsf{PastryCook} &\sqsubseteq \mathsf{Cook} \ , & \mathsf{Baker} &\sqsubseteq \mathsf{PastryCook} \ . \end{aligned}$$

And in addition the accounting department prefers to model things according to their own taste, so there is O_3, the accounting ontology. An excerpt:

$$\begin{aligned} \mathsf{Accountant} &\sqsubseteq \mathsf{Employee} \ , & \mathsf{Director} &\sqsubseteq \mathsf{Employee} \ , \\ \mathsf{SeniorExecutive} &\sqsubseteq \mathsf{Employee} \ , & \mathsf{JuniorExecutive} &\sqsubseteq \mathsf{Employee} \ , \\ \mathsf{BasicStaff} &\sqsubseteq \mathsf{Employee} \ , & \mathsf{SupportStaff} &\sqsubseteq \mathsf{Employee} \ . \end{aligned}$$

With IDDL we are able to integrate these three ontologies into the distributed system $S = \langle \mathbf{O} = \{O_1, O_2, O_3\}, \mathbf{A} \rangle$ using the alignment $\mathbf{A}$ as follows:

$$\begin{aligned} 2 : \mathsf{KitchenChef} &\xleftrightarrow{\sqsubseteq} 1 : \mathsf{Chef} \ , & 2 : \mathsf{Cook} &\xleftrightarrow{\equiv} 1 : \mathsf{Cook} \ , \\ 3 : \mathsf{Accountant} &\xleftrightarrow{\sqsubseteq} 1 : \mathsf{AdministrativeStaff} \ , & 3 : \mathsf{Employee} &\xleftrightarrow{\equiv} 1 : \mathsf{Employee} \ , \\ 3 : \mathsf{SupportStaff} &\xleftrightarrow{\equiv} 1 : \mathsf{SupportStaff} \ , & 3 : \mathsf{BasicStaff} &\xleftrightarrow{\sqsubseteq} 1 : \mathsf{ITStaff} \ . \end{aligned}$$

In the distributed system S we now derive for instance $S \models_{\mathrm{d}} 2 : \mathsf{Baker} \xleftrightarrow{\sqsubseteq} 1 : \mathsf{CateringStaff}$ and as well $S \models_{\mathrm{d}} 2 : \mathsf{Baker} \xleftrightarrow{\sqsubseteq} 3 : \mathsf{SupportStaff}$. In fact, the entailed relation $S \models_{\mathrm{d}} i : X \xleftrightarrow{\sqsubseteq} j : Y$ provide us with a global view, i.e., ontological organization of all concepts regardless of what ontology they originally belonged to.

Besides for mapping between concepts, IDDL allow us to map between roles, in a very analogous fashion. We move on, and in the following example we demonstrate the modeling possibilities with individuals offered by IDDL.

Example 33. *Let us now extend the distributed system S from the previous example by asserting in O_1 the following axiom:*

$$\mathsf{Cook}(\mathsf{johnSmith}) \ ,$$

and asserting the following correspondence:

$$1 : \mathsf{johnSmith} \xleftrightarrow{=} 3 : \mathsf{e006B3F} \ .$$

We are immediately able to derive as a consequence the cross-ontology membership $S \models_{\mathrm{d}} 3 : \mathsf{e006B3F} \xleftrightarrow{\in} 1 : \mathsf{CateringStaff}$. In addition, if we want to further

indicate the specialization of this employee (a 1*-individual), it is not necessary to copy the desired target concept from* O_2 *to* O_1*, instead we use the cross-ontology membership correspondence as an axiom. We add:*

$$1 : \mathsf{johnSmith} \xleftarrow{\in} 2 : \mathsf{Baker} \ .$$

Sometimes it is handy to indicate the disjointness of concepts (also roles) across ontologies. IDDL provides us with means to do it. Let us continue the running example, as follows.

Example 34. *Since the company does not specialize in catering services but in IT instead, we add into our distributed system* S *the following cross-ontology disjointness:*

$$1 : \mathsf{CateringStaff} \xleftarrow{\bot} 3 : \mathsf{Director} \ .$$

We are immediately able to derive both $S \models_{\mathrm{d}} 1 : \mathsf{johnSmith} \xleftarrow{\in} 3 : \neg\mathsf{Director}$ *and* $S \models_{\mathrm{d}} 3 : \mathsf{e006B3F} \xleftarrow{\in} 3 : \neg\mathsf{Director}$ *which are consequences on the global semantics.*

Above we have motivated the IDDL semantics by the idea of a new treatment of mapping. Instead of propagating/importing knowledge from one ontology to another, IDDL aim at providing a global view on top of a collection of ontologies. The newly inferred knowledge is at the level of mapping, that is, at the level of the global semantics. Remarkably, this is not always exclusively the case, sometimes addition of alignments may indeed result into new formula entailments also at the local level.

Example 35. *Let us return once again to our running example and reconsider the distributed system* S*, as introduced above. In Example 34 we have proven that in* S *the following alignment is entailed:* $S \models_{\mathrm{d}} 3 : \mathsf{e006B3F} \xleftarrow{\in} 3 : \neg\mathsf{Director}$*. In other words, this means that in any model* $\mathcal{I}$ *of* S *we always have* $\varepsilon_3\left(\mathsf{e006B3F}^{\mathcal{I}_3}\right) \notin \varepsilon_3\left(\mathsf{Director}^{\mathcal{I}_3}\right)$*. However, since* $\varepsilon_3(\cdot)$ *is a total function, this implies that in* $\Delta^{\mathcal{I}_3}$ $\mathsf{e006B3F}^{\mathcal{I}_3} \notin \mathsf{Director}^{\mathcal{I}_3}$*, and since this holds in every model, it follows that* $s \models_{\mathrm{d}} 3 : \neg\mathsf{Director}(\mathsf{e006B3F})$.

Given the motivation explained above, such a behaviour must be viewed at least as an obstacle of the current IDDL semantics. It should be kept in mind when modeling with IDDL.

As we have seen, IDDL are especially suitable for ontology integration, representation of mapping and reasoning about mapping. In contrast, the other approaches concentrate on reuse of knowledge, for instance, when building a new ontology one may reuse some ontologies that already exist. DDL, $\mathcal{E}$-connections and P-DL are more suitable for this task than IDDL because of two reasons; first, in IDDL the mapping is bidirectional, and hence if ontology O_1 wishes to reuse ontology O_2 also O_2 is affected by O_1; and second, IDDL create a global view which is rather centralized and not distributed.

In contrast, a typical application scenario for IDDL is for instance reasoning about mapping computed by one of the ontology matching algorithms (Euzenat & Shvaiko 2007).

6.6 Comparison

A number of comparisons of the different approaches in distributed and modular ontology representation frameworks has been published in the literature (Kutz et al. 2004, Bao et al. 2006*a*, Cuenca Grau & Kutz 2007, Wang et al. 2007). Some of these approaches concentrate mostly on qualitative features (Wang et al. 2007) or lack the necessary formality (Bao et al. 2006*a*). Formal comparative studies of the expressive power are less frequent (Kutz et al. 2004, Cuenca Grau & Kutz 2007).

In the reminder of this chapter, we concentrate on the formal comparison of the expressive power of the different frameworks. We summarize the existing results (comparison of DDL and $\mathcal{E}$-connections of Kutz et al. (2004)) and we further extend them (comparisons of P-DL/IDDL with DDL).

It has to be further noted, that besides for the approach that we pursuit in this chapter, that is, that of a direct translation between the formalisms, another possible approach is to translate the formalisms into a common language that is capable of accommodating their expressive power. Of course, two such languages that are particularly considerable for this purpose are plain DL and first order logic. Reduction into a plain DL is known for all of the frameworks that we have investigated above (Borgida & Serafini 2003, Cuenca Grau & Kutz 2007, Bao et al. 2009, Zimmermann & Le Duc 2008). Relating their expressive power relying on this reductions is an interesting open issue.

6.6.1 DDL vs. $\mathcal{E}$-connections

As showed by Kutz et al. (2004) and Cuenca Grau & Kutz (2007), under specific assumptions, DDL are reducible into $\mathcal{E}$-connections. Let us rewrite the result using the notation that we have used to introduce $\mathcal{E}$-connections in Sect 6.3.

Theorem 93. *Let $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ be a DDL knowledge base in the disjoint names normal form, defined over an index set I and $\mathcal{SHIQ}$ as the local language, that permits only bridge rules between concepts and partial individual correspondences.*

Let us first construct an alphabet of symbols, that we will use: let ${N_\mathrm{C}}' = N_\mathrm{C}$; ${N_\mathrm{I}}' = N_\mathrm{I}$; $\epsilon_{ii}^{m_i\prime} = N_{\mathrm{R}i}$, for each $i \in I$; $\epsilon_{ij}^{m_i\prime} = \{E_{ij}\}$, for any $i, j \in I$ such that $i \neq j$, and E_{ij} is a new symbol.

On top of this alphabet we now construct an $\mathcal{E}$-connections knowledge base $\Sigma' = \langle \mathcal{K}', \mathcal{R}', \mathcal{A}' \rangle$, that uses the same index set I.

- $\mathcal{K}'_j = \mathcal{T}_j \cup \{\exists E_{ji}.C \sqsubseteq G \mid i : C \xrightarrow{\sqsubseteq} j : G \in \mathfrak{B}\} \cup \{H \sqsubseteq \exists E_{ji}.D \mid i : D \xrightarrow{\sqsubseteq} j : H \in \mathfrak{B}\}$, *for each $j \in I$;*
- $\mathcal{R}'_j = \mathcal{R}_j$, *for each $j \in I$;*
- $\mathcal{A}'_j = \mathcal{A}_j$, *for each $j \in I$;*
- $\mathcal{A}'_\mathcal{E} = \{a \cdot E_{ij} \cdot b \mid i : a \longmapsto j : b \in \mathfrak{B}\}$.

It follows that $\mathfrak{K}$ is globally consistent if and only if Σ' is consistent (i.e., there exists a d-model of $\mathfrak{K}$ if and only if there exists a model of Σ').

The reduction works by simulating the domain relation r_{ij} by the interpretation of the link E_{ij}, for each $i, j \in I$. As a straightforward consequence of the

theorem, also the remaining reasoning tasks related to global consistency are reducible.

Corollary 94. *Deciding d-satisfiability and d-entailment of subsumption with respect to a DDL knowledge base reduces into deciding the consistence of a knowledge base in $\mathcal{E}$-connections.*

This reduction especially shows that there is some similarity between bridge rules in DDL and links in $\mathcal{E}$-connections. However it cannot be claimed that $\mathcal{E}$-connections are more expressive than DDL based on this result. This is for two reasons:

- the reduction presented in Theorem 93 is presented for the DDL semantics based on d-models. However, "the semantics" of DDL is currently the one based ϵ-models, which allows DDL to cope with situations in which some of the local ontologies are inconsistent. $\mathcal{E}$-connections offer no such features and once a single local ontology is inconsistent the whole distributed ontology is inconsistent as well;
- to our best knowledge, no reduction is known for the richer flavours of DDL that include either bridge rules between roles or heterogeneous bridge rules. Also, by putting further restrictions on the domain relation, as we have investigated in Chap. 4 and 5, one derives semantics which are further more distant, and for which no reduction to $\mathcal{E}$-connections is known.

These results and this interpretation thereof are well in line with the different purpose for which $\mathcal{E}$-connections and DDL have been designed. $\mathcal{E}$-connections work with carefully crafted local modules with nonoverlapping modeling domains, while DDL allow also for integration of overlapping ontologies in which part of the terminology is modeled on both sides although possibly differently.

6.6.2 P-DL vs. DDL

Bao et al. have claimed that P-DL are more expressive than DDL (Bao et al. 2006*a*). We formally show in this section, that this is not true, at least not when we consider the standard DDL semantics. We will show that P-DL are interreducible with DDL with semantics with functionality, injectivity, restricted compositionality and role preservation (let us denote this semantics of DDL by DDL$_\epsilon$(F,I,RC,RP)). Recall, that we have studied a weaker variant of the DDL semantics in Sect. 5.2.2.

Both these semantics are different from the original DDL semantics, and hence also P-DL is, strictly speaking, different from DDL. On the other hand, the very same reductions also show, that even if the semantics of the formalisms partly differ, the underlaying concepts of semantic mapping (represented by bridge rules in DDL) and semantic importing (represented by the import relation in P-DL) are strongly related. Thus, among the four distributed ontology frameworks, DDL and P-DL are probably the two most tightly related.

Theorem 95. *Assume a contextualized negation-free $\mathcal{SHIQP}$ ontology $\mathcal{P} = \{P_i\}_{i\in I}$, with the importing relation $P_i \xrightarrow{t} P_j$. Let us construct a DDL knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ over the index set I. For each $i \in I$ the local ontology $\mathcal{K}_i$ of $\mathfrak{K}$ is constructed as follows:*

- $\mathcal{T}_i := \{\phi \mid \phi \in P_i \textit{ is a GCI axiom}\} \cup \{\top^i \equiv \top\}$;
- $\mathcal{R}_i := \{\phi \mid \phi \in P_i \textit{ is a RIA axiom}\}$;
- $\mathcal{A}_i := \{\phi \mid \phi \in P_i \textit{ is an ABox assertion}\}$.

For each $i, j \in I$ the set of bridge rules from i to j is constructed as follows:

- $\mathfrak{B}_{ij} := \left\{ i : C \xrightarrow{\equiv} j : C \;\middle|\; P_i \xrightarrow{C} P_j \right\} \cup \left\{ i : R \xrightarrow{\equiv} j : R \;\middle|\; P_i \xrightarrow{R} P_j \right\} \cup \left\{ i : a \longmapsto j : a \;\middle|\; P_i \xrightarrow{a} P_j \right\}$.

Then given any $i \in I$, $\mathcal{P}$ is i-consistent if and only if $\mathfrak{K}$ is i-consistent under the DDL$_\epsilon$(F,I,RC,RP) semantics.

Proof. The *only if* part. Assume that there exists a distributed interpretation $\mathcal{I}$ of $\mathcal{P}$ with $\Delta^{\mathcal{I}_i} \neq \emptyset$ such that $\mathcal{I} \models \mathcal{P}$. We show that $\mathcal{I}$ is also a distributed model of $\mathfrak{K}$:

1. given any $i \in I$, $\mathcal{I}_i$ satisfies the axiom $\top^i \equiv \top \in \mathcal{T}_i$, because in P-DL we have ${\top_i}^{\mathcal{I}_i} = \top^{\mathcal{I}_i}$ by definition, and it satisfies all remaining axioms of $\mathcal{T}_i$ since they also belong to P_i and $\mathcal{I}_i \models P_i$ as we have assumed. Also, all axioms of $\mathcal{R}_i$ and $\mathcal{A}_i$ are satisfied by $\mathcal{I}_i$ since they all appear also in P_i;
2. we ought to prove that for any $i, j \in I$, $i \neq j$, each bridge rule ϕ of $\mathfrak{B}_{ij}$ is satisfied by $\mathcal{I}$. We distinguish two cases:
   (a) in case that $\phi = i : X \xrightarrow{\equiv} j : X$, where X is either a concept or a role, we know from the construction that $P_i \xrightarrow{X} P_j$. In this case the semantics of P-DL requires $r_{ij}\left(X^{\mathcal{I}_i}\right) = X^{\mathcal{I}_j}$ and hence $\mathcal{I}$ satisfies ϕ;
   (b) in case that $\phi = i : a \longmapsto j : a$, we know from the construction that $P_i \xrightarrow{a} P_j$. Again, the semantics of P-DL requires $r_{ij}\left(a^{\mathcal{I}_i}\right) = a^{\mathcal{I}_j}$ and hence $\mathcal{I}$ satisfies ϕ;
3. from the semantics of P-DL we have that the domain relation r is an injective partial function, it satisfies restricted compositionality and it is role-preserving, exactly as required by Definition 90. This means that the domain relation complies to the DDL$_\epsilon$(F,I,RC,RP) semantics. And hence $\mathcal{I}$ is a distributed model of $\mathfrak{K}$ under the DDL$_\epsilon$(F,I,RC,RP) semantics.

The *if* part. Assume $\mathcal{I} \models_\epsilon \mathfrak{K}$ and that $\mathfrak{K}$ is i-consistent, that is $\mathcal{I}_i \neq \emptyset$. In order to prove that $\mathcal{I}$ is also a model of $\mathcal{P}$ we need to show the following:

1. $\bigcup_{j \in I} \Delta^{\mathcal{I}_j} \neq \emptyset$ – this follows from the fact that $\mathcal{I}$ is i-consistent;
2. for all $i, j \in I$, r_{ij} is a injective partial function, and it is compositionally consistent and role preserving as required by Definition 96. This is because $\mathcal{I}$ is a model of $\mathfrak{K}$ under the DDL$_\epsilon$(F,I,RC,RP) semantics;
3. $r_{ij}\left(t^{\mathcal{I}_i}\right) = t^{\mathcal{I}_j}$ for each term t such that $P_i \xrightarrow{t} P_j$ – if t is a concept or a role, then the construction implies the existence of the bridge rule $i : t \xrightarrow{\equiv} j : t \in \mathfrak{B}_{ij}$ which in turn implies $r_{ij}\left(t^{\mathcal{I}_j}\right) = t^{\mathcal{I}_i}$. Similarly, in case t is an individual, the construction implies $i : t \longmapsto j : t \in \mathfrak{B}_{ij}$ which implies $t^{\mathcal{I}_i} \in r_{ij}\left(t^{\mathcal{I}_j}\right)$. But since $r_{ij}(\cdot)$ is functional, it follows that $r_{ij}\left(t^{\mathcal{I}_j}\right) = \{t^{\mathcal{I}_i}\}$ which we also write as $r_{ij}\left(t^{\mathcal{I}_j}\right) = t^{\mathcal{I}_i}$;

4. $\mathcal{I}_i \models P_i$ – $\mathcal{I}$ is a distributed model of $\mathfrak{K}$, and so $\mathcal{I}_i \models \mathcal{T}_i$, $\mathcal{I}_i \models \mathcal{R}_i$ and $\mathcal{I}_i \models \mathcal{A}_i$, and so also $\mathcal{I}_i \models P_i = \mathcal{T}_i \cup \mathcal{R}_i \cup \mathcal{A}_i$.

□

It trivially follows that also the decision problems of satisfiability and subsumption entailment are reducible, since they are reducible into i-consistence.

Corollary 96. *Deciding satisfiability and subsumption entailment with respect to a P-DL knowledge base reduces into deciding i-consistence of a DDL knowledge base under the DDL$_\epsilon$(F,I,RC,RP) semantics.*

We will now show, that under this strong semantics of DDL, also a reduction in the opposite direction is possible.

Theorem 97. *Let $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ be a DDL knowledge base over some index set I, in atomic names normal form and disjoint names normal form. We construct a package-based ontology $\mathcal{P} = \{P_i\}_{i \in I}$. Each package P_j contains a union of the following components:*

1. $\mathcal{T}_j \cup \{C \sqsubseteq G \mid i : C \xrightarrow{\sqsubseteq} j : G \in \mathfrak{B}\} \cup \{G \sqsubseteq C \mid i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}\}$;
2. $\mathcal{R}_j \cup \{R \sqsubseteq S \mid i : R \xrightarrow{\sqsubseteq} j : S \in \mathfrak{B}\} \cup \{S \sqsubseteq R \mid i : R \xrightarrow{\sqsupseteq} j : S \in \mathfrak{B}\}$;
3. $\mathcal{A}_j \cup \{a = b \mid i : a \longmapsto j : b \in \mathfrak{B}\}$.

In addition, $\mathcal{P}$ uses the following imports:

- $P_i \xrightarrow{C} P_j$ *if either* $i : C \xrightarrow{\sqsubseteq} j : G \in \mathfrak{B}$ *or* $i : C \xrightarrow{\sqsupseteq} j : G \in \mathfrak{B}$, *for any* $i, j \in I$, *for any* i*-concept* C *and for any* j*-concept* G;
- $P_i \xrightarrow{R} P_j$ *if either* $i : R \xrightarrow{\sqsubseteq} j : S \in \mathfrak{B}$ *or* $i : R \xrightarrow{\sqsupseteq} j : S \in \mathfrak{B}$, *for any* $i, j \in I$, *for any* i*-role* R *and for any* j*-role* S;
- $P_i \xrightarrow{a} P_j$ *if* $i : a \longmapsto j : b \in \mathfrak{B}$ *for any* $i, j \in I$, *for any* i*-individual* a *and for any* j*-individual* b.

Given any $i \in I$, $\mathcal{P}$ is i-consistent if and only if $\mathfrak{K}$ is i-consistent under the DDL$_\epsilon$(F,I,RC,RP) semantics.

Proof. The *only if* part. Given an model $\mathcal{I}$ of $\mathcal{P}$ with $\Delta^{\mathcal{I}_i} \neq \emptyset$, let $\mathfrak{I}' = \{\mathcal{I}'_i\}_{i \in I}$ be obtained from $\mathcal{I}$ as follows:

- $\mathcal{I}'_j := \mathcal{I}_j$, for all $j \in I$ such that $P_j \in P_i^*$;
- $\mathcal{I}'_j := \mathcal{I}^\epsilon$, for all $j \in I$ such that $P_j \in \mathcal{P} \setminus P_i^*$;
- $r'_{kl} := r_{kl}$, for all $k, l \in I$ such that $P_k \in P_i^*$ and $P_l \in P_i^*$;
- $r'_{kl} := \emptyset$, for all $k, l \in I$ such that $P_k \in \mathcal{P} \setminus P_i^*$ or $P_l \in \mathcal{P} \setminus P_i^*$.

We will show that $\mathfrak{I}'$ is a distributed model of $\mathfrak{K}$:

1. $\mathfrak{I}' \models_\epsilon \mathcal{K}_j$, for any $j \in I$. If $P_j \notin P_i^*$, this follows from the facts that $\mathcal{T}_j \subseteq P_j$, $\mathcal{R}_j \subseteq P_j$, $\mathcal{A}_j \subseteq P_j$, and $\mathcal{I}'_j = \mathcal{I}_j \models P_j$. If $P_j \notin P_i^*$ this trivially holds, since $\mathcal{I}'_j = \mathcal{I}^\epsilon$ satisfies any axiom;

2. $\mathfrak{I}' \models_\epsilon \phi$, for any $\phi = k : C \xrightarrow{\sqsubseteq} l : G \in \mathfrak{B}$. We need to distinguish three cases:
   (a) if $P_l \notin P_i^*$, then $r_{kl}\big(C^{\mathcal{I}_k}\big) = G^{\mathcal{I}_l} = \emptyset$ and hence $\mathfrak{I}' \models_\epsilon \phi$;
   (b) the case when $P_l \in P_i^*$ and $P_k \notin P_i^*$, is not possible, because due to the construction of $\mathcal{P}$ the existence of ϕ implies $P_k \to P_l$ and hence $P_k \notin P_i^*$ implies $P_l \notin P_i^*$;
   (c) if $P_l \in P_i^*$ and $P_k \in P_i^*$, then since $P_k \xrightarrow{C} P_l$ and $C \sqsubseteq G \in P_i$ we have $r'_{kl}\left(C^{\mathcal{I}'_k}\right) = r_{kl}\big(C^{\mathcal{I}_k}\big) = C^{\mathcal{I}_l} \subseteq G^{\mathcal{I}_l} = G^{\mathcal{I}'_l}$, and hence $\mathfrak{I}' \models_\epsilon \phi$;
3. $\mathfrak{I}' \models_\epsilon \phi$, for any $\phi = k : C \xrightarrow{\sqsupseteq} l : G \in \mathfrak{B}$. In case that $P_l \in P_i^*$ and $P_k \in P_i^*$, we know that $P_k \xrightarrow{C} P_l$ and $G \sqsubseteq C \in P_i$, and hence $r'_{kl}\left(C^{\mathcal{I}'_k}\right) = r_{kl}\big(C^{\mathcal{I}_k}\big) = C^{\mathcal{I}_l} \supseteq G^{\mathcal{I}_l} = G^{\mathcal{I}'_l}$, that is, $\mathfrak{I}' \models_\epsilon \phi$. In any other case the proof is analogous, as above for into-bridge rules;
4. $\mathfrak{I}' \models_\epsilon \phi$, for any bridge rule $\phi \in \mathfrak{B}$ between two roles R and S. This case is proven analogously to the previous two, depending on the type of the bridge rule ϕ;
5. $\mathfrak{I}' \models_\epsilon \phi$, for any $\phi = k : a \longmapsto l : b \in \mathfrak{B}$. In case that $l \in P_i^*$ and $k \in P_i^*$, we know that $P_k \xrightarrow{a} P_l$ and $a = b \in P_i$, and hence $r'_{kl}\left(a^{\mathcal{I}'_k}\right) = r_{kl}\big(a^{\mathcal{I}_k}\big) = a^{\mathcal{I}_l} = b^{\mathcal{I}_l} = b^{\mathcal{I}'_l}$, that is, $\mathfrak{I}' \models_\epsilon \phi$. In any other case the proof is analogous, as above for into-bridge rules;
6. from the construction in directly follows that r' is an injective partial function. It is also role preserving, since for any $k, l \in I$, if $(x, y) \in r'_{kl}$ then both $P_k \in P_i^*$ and $P_l \in P_i^*$, but in this case $r'_{kl} = r_{kl}$ and r is role preserving;
7. in order to prove that r' satisfies restricted compositionality, we must show that for any distinct $j, k, l \in I$ such that there is a path from j to k and also from k to l in $\mathrm{G}_{\mathfrak{K}}$ the condition $r'_{jk} \circ r'_{kl} = r'_{jl}$ is true. Let us consider four cases:
   (a) $P_j \in P_i^*$, $P_k \in P_i^*$ and $P_l \in P_i^*$. In this case $r'_{jk} = r_{jk}$, $r'_{kl} = r_{kl}$ and $r'_{jl} = r_{jl}$ and we know that $r_{jk} \circ r_{kl} = r_{jl}$;
   (b) $P_j \in \mathcal{P} \setminus P_i^*$. From the construction $r'_{jk} = \emptyset$ and $r'_{jl} = \emptyset$ and hence the condition holds;
   (c) $P_l \in \mathcal{P} \setminus P_l^*$. From the construction $r'_{jl} = \emptyset$ and $r'_{kl} = \emptyset$ and hence the condition holds;
   (d) $P_k \in \mathcal{P} \setminus P_k^*$. We will focus on the case when $P_j \in P_i^*$ and $P_l \in P_i^*$, as, any other case is already proven. In this case, however, there is no path from k to l in $\mathrm{G}_{\mathfrak{K}}$, because this would imply that $P_k \in P_i^*$. Hence the condition is not required to hold.

We have proven, that $\mathfrak{I}'$ is a distributed model of $\mathfrak{K}$. In addition we know that $\Delta^{\mathcal{I}'_i} = \Delta^{\mathcal{I}_i} \neq \emptyset$ and hence $\mathfrak{K}$ is i-consistent.

The *if* part. Let $\mathfrak{I}$ be a distributed model of $\mathfrak{K}$ with $\mathcal{I}_i \neq \mathcal{I}^\epsilon$, let us construct $\mathcal{I}'$ by five consecutive steps:

1. $\mathcal{I}'_j := \mathcal{I}_j$, for each $j \in I$ such that $P_j \in P_i^*$;
2. set $C^{\mathcal{I}'_l} = r_{kl}\big(C^{\mathcal{I}_k}\big)$, for each C and for each $k, l \in I$, such that either $k : C \xrightarrow{\sqsubseteq} l : G \in \mathfrak{B}$ or $k : C \xrightarrow{\sqsupseteq} l : G \in \mathfrak{B}$, and $P_k, P_l \in P_i^*$;
3. set $R^{\mathcal{I}'_l} = r_{kl}\big(R^{\mathcal{I}_k}\big)$, for each R and for each $k, l \in I$, such that either $k : R \xrightarrow{\sqsubseteq} l : S \in \mathfrak{B}$ or $k : R \xrightarrow{\sqsupseteq} l : S \in \mathfrak{B}$, and $P_k, P_l \in P_i^*$;
4. set $a^{\mathcal{I}'_l} = r_{kl}\big(a^{\mathcal{I}_k}\big)$, for each R and for each $k, l \in I$, such that there is $k : a \longmapsto l : b \in \mathfrak{B}$ and $P_k, P_l \in P_i^*$;
5. $r'_{kl} := r_{kl}$, for all $k, l \in I$, $k \neq l$;
6. set r'_{jj} to identity on $\Delta^{\mathcal{I}_j}$, for all $j \in I$.

We will show that $\mathcal{I}'$ is a model of $\mathcal{P}$:

1. $\Delta^{\mathcal{I}'_i} = \Delta^{\mathcal{I}_i} \neq \emptyset$, since $\mathfrak{K}$ is i-consistent;
2. $\mathcal{I}'_j \models P_j$, for all $j \in P_i^*$. For any axiom $\phi \in \mathcal{T}_j \cup \mathcal{R}_j \cup \mathcal{A}_j$ we have that $\mathcal{I}_j \models \phi$ because $\mathcal{I}'_j = \mathcal{I}_j \models \mathcal{K}_j$. For any $\phi \in P_j \setminus (\mathcal{T}_j \cup \mathcal{R}_j \cup \mathcal{A}_j)$ we distinguish three cases based on the type of ϕ:
   (a) $\phi = C \sqsubseteq G$: due to construction, either $k : C \xrightarrow{\sqsubseteq} j : G \in \mathfrak{B}$ or $k : G \xrightarrow{\sqsupseteq} j : C \in \mathfrak{B}$. In the first of these two cases, we get $C^{\mathcal{I}'_j} = r_{kj}\big(C^{\mathcal{I}_k}\big) \subseteq G^{\mathcal{I}_j} = G^{\mathcal{I}'_j}$, where the first equation follows from the construction of $\mathcal{I}'$ and the inclusion is due to the bridge rule. In the latter case, we analogously verify that $C^{\mathcal{I}'_j} = C^{\mathcal{I}_j} \subseteq r_{kj}\big(G^{\mathcal{I}_k}\big) = G^{\mathcal{I}'_j}$.
   (b) $\phi = R \sqsubseteq S$: this case is analogous to the previous one;
   (c) ϕ is $a = b$: this case is also analogous;
3. r'_{kl} is an injective partial function, for any $k, l \in I$, because it is either equal to r_{kl} which is an injective partial function due to our choice of the particular variant of DDL semantics, or it is an identity, or it is empty, which are again injective partial functions;
4. $r'_{jl} = r'_{jk} \circ r'_{kl}$, for any three $j, k, l \in I$ such that $P_j \xrightarrow{*} P_l$ and $P_l \xrightarrow{*} P_k$. In this case, however, there must also exist two directed paths, one from j to k and another one from k to l, in $\mathrm{G}_{\mathfrak{K}}$ and hence $r_{jl} = r_{jk} \circ r_{kl}$ because $\mathfrak{I}$ satisfies restricted compositionality. This implies $r'_{jl} = r'_{jk} \circ r'_{kl}$ because $r'_{jl} = r_{jl}$, $r'_{jk} = r_{jk}$ and $r'_{kl} = r_{kl}$.
5. if $P_k \xrightarrow{t} P_l$, then $r'_{kl}\left(t^{\mathcal{I}'_k}\right) = t^{\mathcal{I}'_l}$. This follows from steps 2–4 of construction of $\mathcal{I}'$ and from the fact, that we only have an import $P_k \xrightarrow{t} P_l$ in $\mathcal{P}$ if there is some bridge rule/individual correspondence between $k : t$ and some $l : t'$ in $\mathfrak{K}$;
6. finally, r' is role preserving, because r is role preserving and $r'_{kl} = r_{kl}$ for any $k, l \in I$, $k \neq l$. Note that r'_{jj} is role preserving, for any $j \in I$, because any identity is role preserving.

And so we have shown that $\mathcal{I}'$ is a model of $\mathcal{P}$, and since in addition we know that $\Delta^{\mathcal{I}'_i} \neq \emptyset$, it follows that $\mathcal{P}$ is i-consistent. □

Also the remaining decision problems are reducible, because they are reducible into i-consistence in DDL.

Corollary 98. *Deciding satisfiability of concepts and subsumption entailment in DDL under the semantics DDL$_\epsilon$(F,I,RC,RP), reduces into deciding i-consistence in P-DL.*

As we see, P-DL are closely related to DDL, under the DDL$_\epsilon$(F,I,RC,RP) semantics. More precisely, these results show, that it is possible to implement importing with bridge rules, and vice versa, it is possible to simulate bridge rules with imports and additional local axioms, *if the requirements placed on domain relations are the same.*

These results do not imply, however, that the two frameworks have the same expressivity, for two reasons. First, in terms of expressive power, none of the reductions is complete. DDL does not handle nominals, hence P-DL ontologies with nominals cannot be reduced. On the other hand, we did not provide any reduction for DDL ontologies with heterogeneous bridge rules, and as these type of bridge rules essentially require domain relations that are not functional, it is hard to imagine that any reduction is possible.

The second reason is the fact that the DDL semantics DDL$_\epsilon$(F,I,RC,RP), used by the reductions, is considerably stronger than what is commonly understood as appropriate DDL semantics. This semantics is a weaker version of DDL with restricted compositionality, that we have studied in Sect. 5.2.2. Recall, that we have shown that this semantics does not satisfy the local inconsistency property. The problem also occurs with DDL$_\epsilon$(F,I,RC,RP), and, as we show, it also occurs in P-DL.

Example 36. *Consider a package-based ontology $\mathcal{P}$ consisting of three packages P_1, P_2 and P_3 with the following imports:*

$$P_1 \xrightarrow{C} P_2 \ , \qquad P_2 \xrightarrow{D} P_3 \ , \qquad P_1 \xrightarrow{E} P_3 \ .$$

This is a very basic P-DL setting, each of the three packages imports one concept. Furthermore, let us assume that all tree packages are empty (i.e., $P_1 = P_2 = P_3 = \emptyset$). This means, that the three concepts C, D and E are unrelated. It is easy to show, that if P_2 becomes inconsistent, for some reason, then the imported concept D becomes unsatisfiable in P_3. This is quite intuitive, since D is imported from P_2. However, as we will show, if P_2 becomes inconsistent, also E becomes unsatisfiable in P_3 which we consider rather unintuitive, since this concept is imported from P_1 and is unrelated to D.

Let P_2 be inconsistent. For simplicity, let us assign $P_2 := \{\top \sqsubseteq \bot\}$. Due to the first two imports we get that $P_1 \xrightarrow{*} P_2$ and $P_2 \xrightarrow{*} P_3$, and so the semantics implies that in any model of $\mathcal{P}$, it must be the case that $r_{13} = r_{12} \circ r_{23}$. However, since P_2 is inconsistent, $\Delta^{\mathcal{I}_2} = \emptyset$ in every model, and hence $r_{12} \circ r_{23} = \emptyset$, and hence also $r_{13} = \emptyset$. And so, we also get $E^{\mathcal{I}_3} = r_{13}(E^{\mathcal{I}_1}) = \emptyset$.

The original DDL semantics does not exhibit this problem, as showed by Serafini et al. (2005) (see Sect. 3.2). On the other hand, we have also learned in this thesis, that the original semantics has problems with transitive propagation of the effects of bridge rules (Sect. 5.1), which is not a problem for the semantics with restricted compositionality, and nor for DDL$_\epsilon$(F,I,RC,RP) (Sect. 5.2.2). In

the P-DL setting this reformulates as the problem of transitive propagation of the imported semantic relations, and due to correspondence between P-DL and the DDL under the DDL_ϵ(F,I,RC,RP) semantics, we now have a formal proof that it never appears in P-DL. This clearly marks the difference between the two approaches.

An interesting open issue is put forward by this reduction. If the DDL reasoning algorithm could be adjusted to deal with DDL_ϵ(F,I,RC,RP), a reasoner for P-DL would be obtained. This might be of some value, since implementation of a practical reasoning in P-DL is still an open issue (Bao et al. 2009).

6.6.3 IDDL vs. DDL

Besides for the fact, that DDL work with directed and IDDL with bidirectional mapping, which is the obvious difference, there are some similarities between the two frameworks that one might not guess. We will show a reduction from IDDL into DDL. This is done by reducing the alignment in the IDDL knowledge base into an additional local ontology and linking this local ontology to the rest of the system using bridge rules. Note that the reduction relies on the DDL semantics with domain relations restricted to total functions. This is a rather severe restriction resulting to a strengthening of the original DDL semantics. Later on we discuss whether this restriction is necessary.

Theorem 99. *Assume an IDDL distributed system $S = \langle \mathbf{O}, \mathbf{A} \rangle$ over some index set I such that each local ontology $O_i = \langle T_i, R_i, A_i \rangle$ is in $\mathcal{SHIQ}b$ or some of its sublanguages, and it employs symbols from $N_{\mathrm{C}i}$, $N_{\mathrm{R}i}$ and $N_{\mathrm{I}i}$. Let $I' = I \cup \{\mathbf{a}\}$ where $\mathbf{a}$ is a new index symbol such that $\mathbf{a} \notin I$. Let $N_{\mathrm{C}\mathbf{a}} = \{A_i \mid i \in I \wedge A \in N_{\mathrm{C}i}\}$, $N_{\mathrm{R}\mathbf{a}} = \{R_i \mid i \in I \wedge R \in N_{\mathrm{R}i}\}$ and $N_{\mathrm{I}\mathbf{a}} = \{a_i \mid i \in I \wedge a \in N_{\mathrm{I}i}\}$.*

Let $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ be a DDL knowledge base over I' that uses $\mathcal{SHIQ}b$ with equality as the local language, and is constructed as follows:

- $\mathcal{T}_i := T_i$ *for each* $i \in I$;
- $\mathcal{T}_{\mathbf{a}} := \{C_i \sqsubseteq D_j \mid i : C \xleftarrow{\sqsubseteq} j : D \in A_{ij} \wedge i, j \in I\} \cup \{C_i \sqsubseteq \neg D_j \mid i : C \xleftarrow{\perp} j : D \in A_{ij} \wedge i, j \in I\}$;
- $\mathcal{R}_i := R_i$ *for each* $i \in I$;
- $\mathcal{R}_{\mathbf{a}} := \{R_i \sqsubseteq S_j \mid i : R \xleftarrow{\sqsubseteq} j : S \in A_{ij} \wedge i, j \in I\} \cup \{R_i \sqsubseteq \neg S_j \mid i : R \xleftarrow{\perp} j : S \in A_{ij} \wedge i, j \in I\}$;
- $\mathcal{A}_i := A_i$ *for each* $i \in I$;
- $\mathcal{A}_{\mathbf{a}} := \{C_j(a_i) \mid i : a \xleftarrow{\in} j : C \in \mathcal{A}_{ij} \wedge i, j \in I\} \cup \{a_i = b_j \mid i : a \xleftarrow{=} j : b \in A_{ij} \wedge i, j \in I\}$;
- $\mathfrak{B}_{ij} = \emptyset$ *for* $i, j \in I$;
- *and let us construct the sets $\mathfrak{B}_{k\mathbf{a}}$, for each $k \in I$, starting from empty sets as follows: for each $i, j \in I$ and for each correspondence axiom $i : X \xleftarrow{*} j : Y \in A_{ij}$ of any type, we add $i : X \xrightarrow{\equiv} \mathbf{a} : X_i$ into $\mathfrak{B}_{i\mathbf{a}}$ if X is a concept or a role, or we add $i : X \longmapsto \mathbf{a} : X_i$ into $\mathfrak{B}_{i\mathbf{a}}$ if X is an individual. Analogously, we add $j : Y \xrightarrow{\equiv} \mathbf{a} : Y_j$ into $\mathfrak{B}_{j\mathbf{a}}$ if Y is a concept or a role, or we add $j : Y \longmapsto \mathbf{a} : Y_j$ into $\mathfrak{B}_{j\mathbf{a}}$ if Y is an individual.*

In the following we will assume a semantics of DDL in which the domain relation is restricted to be a total function. It follows that S is consistent if and only if $\mathfrak{K}$ is globally consistent.

Given some $i \in I$ and an i-local formula ϕ, we have:

$$S \models \phi \iff \mathcal{T}_i \models \phi \ .$$

The entailment of cross-ontology formulae is reduced as follows:

- $S \models_{\mathrm{d}} i : C \xleftarrow{\sqsubseteq} j : D \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : C_i \sqsubseteq D_j$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : C \xrightarrow{\equiv} \mathbf{a} : C_i,\ j : D \xrightarrow{\equiv} \mathbf{a} : D_j \}$;
- $S \models_{\mathrm{d}} i : C \xleftarrow{\bot} j : D \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : C_i \sqsubseteq \neg D_j$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : C \xrightarrow{\equiv} \mathbf{a} : C_i,\ j : D \xrightarrow{\equiv} \mathbf{a} : D_j \}$;
- $S \models_{\mathrm{d}} i : R \xleftarrow{\sqsubseteq} j : S \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : R_i \sqsubseteq S_j$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : R \xrightarrow{\equiv} \mathbf{a} : R_i,\ j : S \xrightarrow{\equiv} \mathbf{a} : S_j \}$;
- $S \models_{\mathrm{d}} i : R \xleftarrow{\bot} j : S \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : R_i \sqsubseteq \neg S_j$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : R \xrightarrow{\equiv} \mathbf{a} : R_i,\ j : S \xrightarrow{\equiv} \mathbf{a} : S_j \}$;
- $S \models_{\mathrm{d}} i : a \xleftarrow{\in} j : C \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : C_j(a_i)$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : a \longmapsto \mathbf{a} : a_i,\ j : C \xrightarrow{\equiv} \mathbf{a} : C_j \}$;
- $S \models_{\mathrm{d}} i : a \xleftarrow{=} j : b \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : a_i = b_j$, *where* $\mathfrak{K}' = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B}' \rangle$, $\mathfrak{B}' = \mathfrak{B} \cup \{ i : a \longmapsto \mathbf{a} : a_i,\ j : b \longmapsto \mathbf{a} : b_j \}$.

Proof. Let S be an arbitrary IDDL knowledge base and let $\mathfrak{K}$ be a DDL knowledge base resulting from the construction as given in the theorem. We start the proof by establishing a correspondence between models of S and d-models of $\mathfrak{K}$. Let $\mathfrak{I}$ be any d-model of $\mathfrak{K}$ under the assumed semantics (i.e., the domain relation r of $\mathfrak{I}$ is a function). Let us construct $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$, a distributed interpretation of S, as follows:

- $\mathbf{I} := \{\mathcal{I}_i\}_{i \in I}$ where $\mathcal{I}_i$ is the i-local interpretation of $\mathfrak{I}$;
- $\Delta_\varepsilon := \Delta^{\mathcal{I}_{\mathbf{a}}}$;
- $\varepsilon_i := r_{i\mathbf{a}}$, for each $i \in I$;
- $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i \in I}$.

Let us prove that $\mathcal{I}$ is a model of S:

- indeed $\mathcal{I}$ is a distributed interpretation: $\mathcal{I}_i$ is a local interpretation of the respective local knowledge base, with a non-empty domain, and ε_i is a function, for each $i \in I$;
- $\mathcal{I}_i$ is a model of the local ontology O_i, for each $i \in I$, because it is a model of the local ontology $\mathcal{K}_i = \{\mathcal{T}_i, \mathcal{R}_i, \mathcal{A}_i\}$ which is identical to O_i;
- for each $\phi = i : C \xleftarrow{\sqsubseteq} j : D \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: we have $\varepsilon_i(C^{\mathcal{I}_i}) = r_{i\mathbf{a}}(C^{\mathcal{I}_i}) = C_i{}^{\mathcal{I}_{\mathbf{a}}} \subseteq D_j{}^{\mathcal{I}_{\mathbf{a}}} = r_{j\mathbf{a}}(D^{\mathcal{I}_j}) = \varepsilon_j(D^{\mathcal{I}_j})$ because $\varepsilon_i = r_{i\mathbf{a}}$, $i : C \xrightarrow{\equiv} \mathbf{a} : C_i \in \mathfrak{B}$, $C_i \sqsubseteq D_j \in \mathcal{T}_{\mathbf{a}}$, $j : D \xrightarrow{\equiv} \mathbf{a} : D_j \in \mathfrak{B}$ and $\varepsilon_j = r_{j\mathbf{a}}$;

- for each $\phi = i : R \xleftrightarrow{\sqsubseteq} j : S \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: this case is completely analogous to the previous one;

- for each $\phi = i : C \xleftrightarrow{\bot} j : D \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: we have $\varepsilon_i(C^{\mathcal{I}_i}) = r_{i\mathbf{a}}(C^{\mathcal{I}_i}) = {C_i}^{\mathcal{I}_\mathbf{a}}$ and ${D_j}^{\mathcal{I}_\mathbf{a}} = r_{j\mathbf{a}}(D^{\mathcal{I}_j}) = \varepsilon_j(D^{\mathcal{I}_j})$ because $\varepsilon_i = r_{i\mathbf{a}}$, $i : C \xrightarrow{\equiv} \mathbf{a} : C_i \in \mathfrak{B}$, $j : D \xrightarrow{\equiv} \mathbf{a} : D_j \in \mathfrak{B}$ and $\varepsilon_j = r_{j\mathbf{a}}$. However, since $C_i \sqsubseteq \neg D_j \in \mathcal{T}_\mathbf{a}$, it follows that ${C_i}^{\mathcal{I}_\mathbf{a}} \cap {D_j}^{\mathcal{I}_\mathbf{a}} = \emptyset$, which in turn implies $\varepsilon_i(C^{\mathcal{I}_i}) \cap \varepsilon_j(D^{\mathcal{I}_j}) = \emptyset$;

- for each $\phi = i : R \xleftrightarrow{\bot} j : S \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: again, this case is completely analogous to the previous one;

- for each $\phi = i : a \xleftrightarrow{\in} j : C \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: first of all, note that since the domain relation is functional, then if the axiom $i : a \longmapsto \mathbf{a} : a_i \in \mathfrak{B}$ is satisfied, we have ${a_i}^{\mathcal{I}_\mathbf{a}} \in r_{i\mathbf{a}}(a^{\mathcal{I}_i})$, but this in fact implies $r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = \{{a_i}^{\mathcal{I}_\mathbf{a}}\}$, which we also write as $r_{j\mathbf{a}}(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_\mathbf{a}}$.[5] In light of this observation, we have $\varepsilon_i(a^{\mathcal{I}_i}) = r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_\mathbf{a}} \in {C_j}^{\mathcal{I}_\mathbf{a}} = r_{j\mathbf{a}}(C^{\mathcal{I}_j}) = \varepsilon_j(C^{\mathcal{I}_j})$ because $\varepsilon_i = r_{i\mathbf{a}}$, $i : a \longmapsto \mathbf{a} : a_i \in \mathfrak{B}$, $C_j(a_i) \in \mathcal{A}_\mathbf{a}$, $j : C \xrightarrow{\equiv} \mathbf{a} : C_j \in \mathfrak{B}$ and $\varepsilon_j = r_{j\mathbf{a}}$;

- for each $\phi = i : a \xleftrightarrow{=} j : b \in A_{ij}$, for each $i, j \in I$, we have $\mathcal{I} \models_{\mathrm{d}} \phi$: we have $\varepsilon_i(a^{\mathcal{I}_i}) = r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_\mathbf{a}} = {b_j}^{\mathcal{I}_\mathbf{a}} = r_{j\mathbf{a}}(b^{\mathcal{I}_j}) = \varepsilon_j(b^{\mathcal{I}_j})$ because $\varepsilon_i = r_{i\mathbf{a}}$, $i : a \longmapsto \mathbf{a} : a_i \in \mathfrak{B}$, $a_i = b_j \in \mathcal{A}_\mathbf{a}$, $j : b \longmapsto \mathbf{a} : b_j \in \mathfrak{B}$, $\varepsilon_j = r_{j\mathbf{a}}$, and from the fact that the domain relation is functional.

Conversely, given a model $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ of S let us construct a distributed interpretation $\mathfrak{I}$ of $\mathfrak{K}$ as follows:

- the local interpretation $\mathcal{I}_i$ is the same as $\mathcal{I}_i \in \mathbf{I}$, for each $i \in I$;

- $\Delta^{\mathcal{I}_\mathbf{a}} := \Delta_\varepsilon$;

- ${X_i}^{\mathcal{I}_\mathbf{a}} := \varepsilon_i(X^{\mathcal{I}_i})$, for each i-individual, i-role, or i-concept X, for each $i \in I$;

- $r_{i\mathbf{a}} := \varepsilon_i$, for each $i \in I$;

- $r_{ij} = \emptyset$, for each $i, j \in I$.

We will show that $\mathfrak{I}$ is a d-model of $\mathfrak{K}$:

- apparently, $\mathfrak{I}$ is a distributed interpretation of $\mathfrak{K}$, and for each $k \in I'$ we have $\Delta^{\mathcal{I}_k} \neq \emptyset$;

- $\mathcal{I}_i$ is a model of the local ontology $\mathcal{K}_i$, for each $i \in I$, because it is a model of the local ontology O_i which is identical to $\mathcal{K}_i$;

- $\mathcal{I}_\mathbf{a}$ is a model of $\mathcal{K}_\mathbf{a}$, as we prove:

[5] This means that the partial individual correspondence axiom which we employ here effectively performs as total individual correspondence thanks to the functionality of the domain relation. We will frequently make use of this fact in this proof.

  - for every $\phi = C_i \sqsubseteq D_i \in \mathcal{T}_\mathrm{a}$ it follows from the construction that either there is $i : C \overset{\sqsubseteq}{\longleftrightarrow} j : D \in A_{ij}$ (case 1) or there is $i : C \overset{\bot}{\longleftrightarrow} j : \neg D \in A_{ij}$ (case 2). In the first case, we get ${C_i}^{\mathcal{I}_\mathbf{a}} = \varepsilon_i(C^{\mathcal{I}_i}) \subseteq \varepsilon_j(D^{\mathcal{I}_j}) = {D_j}^{\mathcal{I}_\mathbf{a}}$, and hence $\mathcal{I}_a \models \phi$. In the other case, we get the two sets ${C_i}^{\mathcal{I}_\mathbf{a}} = \varepsilon_i(C^{\mathcal{I}_i})$ and $\neg {D_j}^{\mathcal{I}_\mathbf{a}} = \varepsilon_j(\neg D^{\mathcal{I}_j})$ are disjoint, which implies ${C_i}^{\mathcal{I}_\mathbf{a}} \subseteq {D_j}^{\mathcal{I}_\mathbf{a}}$, and hence $\mathcal{I}_a \models \phi$;
  - for every $\phi = R_i \sqsubseteq S_i \in \mathcal{R}_\mathrm{a}$ the proof is completely analogous to the previous case;
  - for every $\phi = C_j(a_i) \in \mathcal{A}_\mathbf{a}$ we have $i : a \overset{\in}{\longrightarrow} j : C \in A_{ij}$ and hence ${a_i}^{\mathcal{I}_\mathbf{a}} = \varepsilon_i(a^{\mathcal{I}_i}) \in \varepsilon_j(C^{\mathcal{I}_j}) = {C_j}^{\mathcal{I}_\mathbf{a}}$, that is $\mathcal{I}_a \models \phi$;
  - for every $\phi = a_i = b_j \in \mathcal{A}_\mathbf{a}$ we have $i : a \overset{=}{\longrightarrow} j : b \in A_{ij}$ and hence ${a_i}^{\mathcal{I}_\mathbf{a}} = \varepsilon_i(a^{\mathcal{I}_i}) = \varepsilon_j(b^{\mathcal{I}_j}) = {b_j}^{\mathcal{I}_\mathbf{a}}$, that is $\mathcal{I}_a \models \phi$;

- for each $\phi = i : X \overset{\equiv}{\longrightarrow} \mathbf{a} : X_i \in \mathfrak{B}$, for each $i \in I$, we know from the construction of $\mathfrak{I}$ that $r_{i\mathbf{a}}(X^{\mathcal{I}_i}) = \varepsilon_i(X^{\mathcal{I}_i}) = {X_i}^{\mathcal{I}_\mathbf{a}}$, and hence $\mathfrak{I} \models \phi$;
- for each $\phi = i : a \longmapsto \mathbf{a} : a_i \in \mathfrak{B}$, for each $i \in I$, we analogously get $r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = \varepsilon_i(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_\mathbf{a}}$, and hence $\mathfrak{I} \models \phi$.

Since there are no other formulae in $\mathfrak{K}$, we have just proven that $\mathfrak{I}$ is a d-model of $\mathfrak{K}$. Moreover, we have also proven that S is consistent if and only if $\mathfrak{K}$ is globally consistent: from the correspondence it follows that if S has a model, $\mathfrak{K}$ has a d-model and vice versa.

In the second part of the proof we prove the reduction of the entailment reasoning tasks:

- $S \models_\mathrm{d} i : C \overset{\sqsubseteq}{\longleftrightarrow} j : D \iff \mathfrak{K}' \models_\mathrm{d} \mathbf{a} : C_i \sqsubseteq D_j$:
  - the *only if* part: let $\mathfrak{I}$ be a d-model of $\mathfrak{K}'$. Let $\mathcal{I}$ be the corresponding distributed interpretation obtained by the construction above. Thanks to monotonicity of both formalisms, $\mathcal{I}$ is a model of S, because $\mathfrak{K} \subseteq \mathfrak{K}'$. Since $S \models_\mathrm{d} i : C \overset{\sqsubseteq}{\longleftrightarrow} j : D$, we have $\varepsilon_i(C^{\mathcal{I}_i}) \subseteq \varepsilon_j(D^{\mathcal{I}_j})$. Due to construction we have that $r_{i\mathbf{a}}(C^{\mathcal{I}_i}) = \varepsilon_i(C^{\mathcal{I}_i}) \subseteq \varepsilon_j(D^{\mathcal{I}_j}) = r_{j\mathbf{a}}(D^{\mathcal{I}_j})$. And finally, due to the two new bridge rules added to $\mathfrak{K}'$ we have ${C_i}^{\mathcal{I}_\mathbf{a}} = r_{i\mathbf{a}}(C^{\mathcal{I}_i}) = \varepsilon_i(C^{\mathcal{I}_i}) \subseteq \varepsilon_j(D^{\mathcal{I}_j}) = r_{j\mathbf{a}}(D^{\mathcal{I}_j}) = {D_j}^{\mathcal{I}_\mathbf{a}}$, which means that $\mathfrak{I} \models_\mathrm{d} \mathbf{a} : C_i \sqsubseteq D_j$;
  - the *if* part: let $\mathcal{I}$ be a model of S and let $\mathfrak{I}$ be the corresponding d-model of $\mathfrak{K}$. Let us extend $\mathfrak{I}$ by asserting ${C_i}^{\mathcal{I}_\mathbf{a}} := \varepsilon_i(C^{\mathcal{I}_i})$ and ${D_j}^{\mathcal{I}_\mathbf{a}} := \varepsilon_j(D^{\mathcal{I}_j})$ (please note that if ${C_i}^{\mathcal{I}_\mathbf{a}}$ and ${D_j}^{\mathcal{I}_\mathbf{a}}$ were already defined in $\mathfrak{I}$ they were defined exactly the same way, please refer to the construction of $\mathfrak{I}$). Now the extended $\mathfrak{I}$ is a d-model of $\mathfrak{K}'$: it satisfies every axiom of $\mathfrak{K}$ since we did not change the interpretation of any concept present already in $\mathfrak{K}$ and it now also satisfies the two newly added bridge rules. However, in this case ${C_i}^{\mathcal{I}_\mathbf{a}} \subseteq {D_j}^{\mathcal{I}_\mathbf{a}}$, because we have assumed that $\mathfrak{K}' \models_\mathrm{d} \mathbf{a} : C_i \sqsubseteq D_j$. Due to the construction it follows $\varepsilon_i(C^{\mathcal{I}_i}) = {C_i}^{\mathcal{I}_\mathbf{a}} \subseteq {D_j}^{\mathcal{I}_\mathbf{a}} = \varepsilon_j(D^{\mathcal{I}_j})$, and so also $\mathcal{I} \models_\mathrm{d} i : C \overset{\sqsubseteq}{\longleftrightarrow} j : D$;

- $S \models_\mathrm{d} i : C \overset{\bot}{\longleftrightarrow} j : D \iff \mathfrak{K}' \models_\mathrm{d} \mathbf{a} : C_i \sqsubseteq \neg D_j$:

  - the *only if* part: again, given a d-model $\mathfrak{I}$ of $\mathfrak{K}'$, the corresponding distributed interpretation $\mathcal{I}$ is a model of S. Since $S \models_{\mathrm{d}} i : C \overset{\perp}{\longleftrightarrow} j : D$, we have $\varepsilon_i(C^{\mathcal{I}_i}) \cap \varepsilon_j(D^{\mathcal{I}_j}) = \emptyset$. Due to the construction and due to the pair of bridge rules newly added to $\mathfrak{K}'$ we now have ${C_i}^{\mathcal{I}_{\mathbf{a}}} = r_{i\mathbf{a}}(C^{\mathcal{I}_i}) = \varepsilon_i(C^{\mathcal{I}_i})$ and ${D_j}^{\mathcal{I}_{\mathbf{a}}} = r_{j\mathbf{a}}(D^{\mathcal{I}_j}) = \varepsilon_j(D^{\mathcal{I}_j})$ are disjoint. This, however, implies ${C_i}^{\mathcal{I}_{\mathbf{a}}} \subseteq \neg {D_j}^{\mathcal{I}_{\mathbf{a}}}$ and hence also $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : C_i \sqsubseteq \neg D_j$;
  - the *if* part: let $\mathcal{I}$ be a model of S and let $\mathfrak{I}$ be the corresponding d-model of $\mathfrak{K}$. Let us again extend $\mathfrak{I}$ by asserting ${C_i}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_i(C^{\mathcal{I}_i})$ and ${D_j}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_j(D^{\mathcal{I}_j})$ which, as we already know leads to a d-model of $\mathfrak{K}'$. In this model, however, the two sets $\varepsilon_i(C^{\mathcal{I}_i}) = {C_i}^{\mathcal{I}_{\mathbf{a}}}$ and $\varepsilon_j(D^{\mathcal{I}_j}) = {D_j}^{\mathcal{I}_{\mathbf{a}}}$ are disjoint because $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : C_i \sqsubseteq \neg D_j$. It follows that $\mathcal{I} \models_{\mathrm{d}} i : C \overset{\perp}{\longleftrightarrow} j : D$;

- $S \models_{\mathrm{d}} i : R \overset{\sqsubseteq}{\longrightarrow} j : S \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : R_i \sqsubseteq S_j$: this case is analogous to the first case above;

- $S \models_{\mathrm{d}} i : R \overset{\perp}{\longleftrightarrow} j : S \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : R_i \sqsubseteq \neg S_j$: this case is analogous to the second case above;

- $S \models_{\mathrm{d}} i : a \overset{\in}{\longrightarrow} j : C \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : C_j(a_i)$:

  - the *only if* part: again, given a d-model $\mathfrak{I}$ of $\mathfrak{K}'$, the corresponding distributed interpretation $\mathcal{I}$ is a model of S. Since $S \models_{\mathrm{d}} i : a \overset{\in}{\longrightarrow} j : C$, we have $\varepsilon_i(a^{\mathcal{I}_i}) \in \varepsilon_j(C^{\mathcal{I}_j})$. Due to the construction, due to the pair of bridge rules newly added to $\mathfrak{K}'$, and due to the functionality of the domain relation, we now have ${a_i}^{\mathcal{I}_{\mathbf{a}}} = r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = \varepsilon_i(a^{\mathcal{I}_i}) \in \varepsilon_j(C^{\mathcal{I}_j}) = r_{j\mathbf{a}}(C^{\mathcal{I}_j}) = {C_j}^{\mathcal{I}_{\mathbf{a}}}$. Hence also $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : C_j(a_i)$;
  - the *if* part: let $\mathcal{I}$ be a model of S and let $\mathfrak{I}$ be the corresponding d-model of $\mathfrak{K}$. Let us again extend $\mathfrak{I}$ by asserting ${a_i}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_i(a^{\mathcal{I}_i})$ and ${C_j}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_j(C^{\mathcal{I}_j})$ which once again yields a d-model of $\mathfrak{K}'$. In this model, however, we have $\varepsilon_i(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_{\mathbf{a}}} \in {C_j}^{\mathcal{I}_{\mathbf{a}}} = \varepsilon_i(C^{\mathcal{I}_j})$ because $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : C_j(a_i)$, and hence it also holds $\mathcal{I} \models_{\mathrm{d}} i : a \overset{\in}{\longrightarrow} j : C$;

- $S \models_{\mathrm{d}} i : a \overset{=}{\longrightarrow} j : b \iff \mathfrak{K}' \models_{\mathrm{d}} \mathbf{a} : a_i = b_j$:

  - the *only if* part: again, given a d-model $\mathfrak{I}$ of $\mathfrak{K}'$, the corresponding distributed interpretation $\mathcal{I}$ is a model of S. Since $S \models_{\mathrm{d}} i : a \overset{=}{\longrightarrow} j : C$, we have $\varepsilon_i(a^{\mathcal{I}_i}) = \varepsilon_j(b^{\mathcal{I}_j})$. Due to the construction, due to the pair of bridge rules newly added to $\mathfrak{K}'$, and due to the functionality of the domain relation, we now have ${a_i}^{\mathcal{I}_{\mathbf{a}}} = r_{i\mathbf{a}}(a^{\mathcal{I}_i}) = \varepsilon_i(a^{\mathcal{I}_i}) = \varepsilon_j(b^{\mathcal{I}_j}) = r_{j\mathbf{a}}(b^{\mathcal{I}_j}) = {b_j}^{\mathcal{I}_{\mathbf{a}}}$. Hence also $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : a_i = b_j$;
  - the *if* part: let $\mathcal{I}$ be a model of S and let $\mathfrak{I}$ be the corresponding d-model of $\mathfrak{K}$. Let us again extend $\mathfrak{I}$ by asserting ${a_i}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_i(a^{\mathcal{I}_i})$ and ${b_j}^{\mathcal{I}_{\mathbf{a}}} := \varepsilon_j(b^{\mathcal{I}_j})$ which once again yields a d-model of $\mathfrak{K}'$. In this model, however, we have $\varepsilon_i(a^{\mathcal{I}_i}) = {a_i}^{\mathcal{I}_{\mathbf{a}}} = {b_j}^{\mathcal{I}_{\mathbf{a}}} = \varepsilon_j(b^{\mathcal{I}_j})$ because $\mathfrak{I} \models_{\mathrm{d}} \mathbf{a} : a_i = b_j$, and hence it also holds $\mathcal{I} \models_{\mathrm{d}} i : a \overset{=}{\longrightarrow} j : C$.

□

And so, we have shown that reasoning in IDDL is reducible into DDL, if we take d-satisfiability and d-entailment, and assuming the semantics of DDL with total and functional domain relations. While this reduction is not unconditional because both semantics are slightly different, it again shows the relation between the bidirectional alignments of IDDL and the directed bridge rules employed in DDL: IDDL alignments are possibly captured by a separate local ontology, and the binding between this ontology and the rest of the local knowledge bases is comfortably modeled with bridge rules.

The reduction does heavily rely on the restricted DDL semantics with functional domain relations. This is a rather exotic version of DDL therefore a natural question arises, whether a reduction to the unrestricted DDL is also possible. In the following example we show that the reduction that we have presented cannot be extended in a straightforward fashion to achieve this.

Example 37. *Assume the IDDL knowledge base S with two ontologies $\mathbf{O} = \{O_1, O_2\}$ and the alignment $\mathbf{A} = \{A_{12}\}$ between these two ontologies. Let there be the following two axioms in S:*

$$D \sqsubseteq \bot \in \mathcal{T}_2 \ ,$$
$$1 : C \xleftarrow{\sqsubseteq} 2 : D \in \mathcal{A}_{12} \ .$$

Clearly, also C is unsatisfiable with respect to S, which we state using the IDDL terms as $S \models_{\mathrm{d}} 1 : C \sqsubseteq \bot$. This is due to the fact that the two equalizing functions $\varepsilon_1, \varepsilon_2$ are functions in the true sense. In any model $\mathcal{I}$, in which $C^{\mathcal{I}_1} \neq \emptyset$, and hence there exists some $x \in C^{\mathcal{I}_1}$, there must also exist the ε_1-image x' of x in Δ_ε. That is, $\varepsilon_1(x) = x' \in \Delta_\varepsilon$. Due to the cross-ontology subsumption axiom we have $\varepsilon_1\left(C^{\mathcal{I}_1}\right) \subseteq \varepsilon_2\left(D^{\mathcal{I}_2}\right)$ which implies $x' \in \varepsilon_2\left(D^{\mathcal{I}_2}\right)$. Then there however must be a ε_2-preimage y of x' in $D^{\mathcal{I}_2}$, that is an element of a set which we assumed to be empty.

Applying the reduction of Theorem 99 on S results in a DDL knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{R}, \mathfrak{A}, \mathfrak{B} \rangle$ over index the set $I = \{1, 2, \mathbf{a}\}$ with the following axioms:

$$2 : D \sqsubseteq \bot \ , \qquad 1 : C \xrightarrow{\equiv} \mathbf{a} : C_1 \ ,$$
$$\mathbf{a} : C_1 \sqsubseteq D_2 \ , \qquad 2 : D \xrightarrow{\equiv} \mathbf{a} : D_2 \ .$$

Assuming the restricted DDL semantics in which $r_{ij}(\cdot)$ is a function $r_{ij} : \Delta^{\mathcal{I}_i} \to \Delta^{\mathcal{I}_j}$, we similarly derive that C is unsatisfiable with respect to $\mathfrak{K}$ (the proof is just as in the case of S, only now the domain relations $r_{1\mathbf{a}}$ and $r_{2\mathbf{a}}$ stand in place of ε_1 and ε_2).

Under the unrestricted semantics this is no longer true however. Consider the distributed interpretation $\mathfrak{I} = \langle \mathcal{I}, r \rangle$ with $\Delta^{\mathcal{I}_1} = \{x\}$, $\Delta^{\mathcal{I}_2} = \Delta^{\mathcal{I}_\mathbf{a}} = \emptyset$, $r_{1\mathbf{a}} = r_{2\mathrm{a}} = \emptyset$, which interprets the concepts as follows: $C^{\mathcal{I}_1} = \{x\}$, ${C_1}^{\mathcal{I}_\mathbf{a}} = {D_2}^{\mathcal{I}_\mathbf{a}} = D^{\mathcal{I}_2} = \emptyset$. Indeed $\mathfrak{I}$ is a model of $\mathfrak{K}$ under the unrestricted semantics. We have $r_{1\mathbf{a}}\left(C^{\mathcal{I}_1}\right) = \emptyset$, which is fine, as $r_{1\mathbf{a}}(\cdot)$ is no longer required to be a function. All axioms of $\mathfrak{K}$ are satisfied in $\mathfrak{I}$. Finally, $C^{\mathcal{I}_1}$ is nonempty, and so under this semantics C is satisfiable with respect to $\mathfrak{K}$.

This example shows, that if we want to use the reduction as defined in Theorem 99 functional domain relations are a strict requirement, otherwise the theorem would be falsified.

In this section, we have presented a formal reduction of IDDL into DDL. The DDL semantics required by the reduction is a rather nonstandard one, with domain relations restricted to total functions. Hence we cannot interpret this result as IDDL being a special case of the original DDL. Nevertheless, the reduction provides some valuable insight on the relation of the two mapping approaches that exist in the distributed ontology representation field. As it has been previously noted (Zimmermann & Le Duc 2008), the alignments, which are bidirectional, can be understood as a separate alignment ontology, that is somehow linked to the system formed by the remaining local ontologies in the knowledge base. Using exactly this metaphor we were able to link the alignment ontology to the rest of the system with use of bridge rules, although with an adjusted semantics.

Again, an interesting open issue put forward by this reduction is the question whether it is possible to adjust the DDL reasoner DRAGO (Serafini & Tamilin 2005) to cover this specific DDL semantics. This would yield a readily available reasoner for IDDL. To our best knowledge, no implementation of a native, distributed reasoner for IDDL is known, and the reasoning algorithm described by (Zimmermann & Le Duc 2008) is 3ExpTime-hard. Hence, any other competitive approach would be of some consideration.

6.7 Summary

This chapter gives a comparative presentation of the most important logical representation frameworks for distributed and modular ontologies: DDL, $\mathcal{E}$-connections, P-DL and IDDL. We have reviewed each of these formalisms, concentrating on their expressive power and modeling with these formalisms: by which practical features the modular ontology development is supported in each of the formalisms and how one is able to use these features in order to model ontologies in a modular fashion. We summarize our observation as follows:

- DDL allow for distributed ontologies in which the component modules are connected by directional semantic mapping. With this mapping, concepts, roles and individuals from distinct components are semantically associated. This allows for reuse of knowledge between the component ontologies. This reuse is directed, that is, if ontology O_1 reuses ontology O_2, the latter is not affected at all. DDL are able to cope with partially overlapping modeling domains and as well with inconsistency that appears locally in one of the component modules;

- $\mathcal{E}$-connections allow to combine several component ontology modules under the assumption that the local modeling domains are strictly disjoint. In $\mathcal{E}$-connections, one connects ontologies with links, practically inter-ontology roles. This framework is especially applicable when building a complex ontology in a modular fashion, and from the beginning it is understood how the local modeling domains should be separated. $\mathcal{E}$-connections also provide an elaborate transitivity control for combinations of local and inter-ontology roles;

- P-DL allow to combine several ontology modules by semantic importing of ontology entities, that is, concepts, roles and individuals. Semantically,

P-DL overcome some of the limitations of the other approaches. The greatest advantage of P-DL is the fact that ontology importing is intuitive and easily understood by the users;

- IDDL allow for integration of several ontologies with bidirectional semantic mapping, while maintaining both the local view (of each component ontology) and as well the global view (the result of integration). While the previous approaches aim at ontology combination and reuse, IDDL aim more at ontology integration. Typical application for IDDL is reasoning about automatically computed ontology mapping.

All four frameworks feature distributed reasoning algorithms, however, only for DDL and $\mathcal{E}$-connections the reasoning support is available in form of practical implementations (DDL reasoner DRAGO and $\mathcal{E}$-connections support in the reasoner Pellet). To our best knowledge, no practical implementations of reasoners for the other two frameworks are known, which leaves them, for the time being, at the level of theoretical proposals without practical applicability.

Later on in this chapter, our attention has diverted towards formal comparison of the expressive features of these frameworks. We have discussed three reductions between the frameworks, one is a known result and two are new:

- DDL are reducible into $\mathcal{E}$-connections, assuming the DDL semantics that relies on d-models and excluding bridge rules between roles and heterogeneous bridge rules (Kutz et al. 2004);
- P-DL are reducible into DDL, assuming the DDL semantics with functional, role preserving and compositional consistent domain relations;
- IDDL are reducible into DDL, assuming the DDL semantics with total and functional domain relations.

In the light of these results, we are able to conclude that there indeed are numerous similarities between the approaches. However, it cannot be claimed that one of the approaches is strictly more expressive as any other, as there are also important differences, which are also demonstrated by the very same results. This findings are well in line with the different purpose for which each of the frameworks has been designed.

Chapter 7

Conclusion and Future Work

In this thesis, we have concentrated on formal logical frameworks intended for modular and distributed ontology knowledge representation. Most of our attention has been captured by Distributed Description Logics (DDL), but we have also compared DDL with $\mathcal{E}$-connections, Package-based Description Logics (P-DL) and Integrated Distributed Description Logics (IDDL).

All of these frameworks employ a combined knowledge base, comprising of multiple ontology modules. There are three main approaches in ontology combination: ontology mapping, ontology linking and ontology importing. DDL fall under the paradigm of ontology mapping. In a DDL knowledge base, local ontologies are combined by directed semantic mapping expressed between ontology entities of various types. The mapping is exemplified by axioms called bridge rules. There are two basic types of bridge rules: into- and onto-bridge rules. In DDL, the mechanism of knowledge reuse is characterized by the effect of subsumption propagation: a subsumption, that is entailed in one local ontology is propagated into another ontology due to bridge rules.

In this thesis, we have conducted a study of subsumption propagation in DDL. Basic subsumption propagation patterns that occur in DDL have been characterized by Borgida & Serafini (2003) and Serafini et al. (2005). A complaint about a behaviour of DDL that appeared in the literature (Cuenca Grau et al. 2004*b*) leads us to study the impact of bridge rules on complex concepts that are constructed from more elementary concepts that are mapped between by bridge rules. We have learned, that subsumption propagation is sufficient in cases when the concept union constructor is involved, however, we have also learned, that subsumption propagation is not sufficient in situations when the concept intersection constructors and onto-bridge rules are involved.

In order to cope with the issue, we have introduced so called conjunctive onto-bridge rules. We have shown, that with this new form of onto-bridge rules, whose semantics is stronger, the problem does not occur. Concerned about the computational properties of DDL with conjunctive onto-bridge rules, we have considered a different approach in which the injectivity of domain relations in distributed models is required. We have shown, that this semantics is yet stronger than the one of conjunctive onto-bridge rules, and hence it also solves the problem. We conjecture, that this semantics may be possibly computationally less complex to handle, as, in a sense, it applies a single restriction globally, instead of applying multiple restrictions locally. In further evaluation we have

also shown, that all of the desired properties postulated for DDL (Borgida & Serafini 2003, Serafini et al. 2005) are satisfied by both these semantics.

Later on in this work, we have directed our attention towards the problem of subsumption propagation between so called remote ontologies in DDL. These are local ontologies within a DDL knowledge base, which are not connected directly, but instead they are connected by paths of two or more bridge rules. To our best knowledge, no prior study of subsumption propagation in such cases has appeared in the literature. We have identified scenarios, in which we consider the occurrence of such long-distance subsumption propagation to be natural and intuitive. These scenarios involve so called chains of bridge rules. Two types of chains have to be considered: into-chains, composed of into-bridge rules, and onto-chains, composed of onto-bridge rules.

We have learned, that under the original DDL semantics the amount of subsumption propagation along chains of bridge rules is rather limited. This leads us again to the study of different strengthenings of the semantics that could possibly exhibit an increased level of subsumption propagation between remote ontologies. We have borrowed an idea of compositionally consistent domain relations, which has been previously applied in P-DL (Bao et al. 2006*b*, 2009). Under compositional consistency, special attention is given to indirect domain mappings between local interpretations within a distributed model, and it is required that direct domain mappings are equal to any admissible composition of indirect domain mappings. We have consecutively studied three different semantic variants of the original DDL semantics, each derived by exploiting the application of the compositional consistency restriction.

We have started with the semantics, in which compositional consistency is required globally, in the whole distributed model. We have shown, that under this semantics, subsumption propagates along chains of into- and onto-bridge rules to a sufficient extent, however, the directionality and the local inconsistency properties, both considered important for DDL (Borgida & Serafini 2003, Serafini et al. 2005) are violated. Willing to retain these desired properties, we have investigated a different semantics, in which the compositional consistency requirement is applied more cautiously, only between ontologies that are in fact connected by bridge rules. In this case, as we have learned, subsumption propagation is at the same satisfactory level, and moreover directionality is satisfied. Unfortunately, the local inconsistency property is not. This has lead us to the third, further relaxed version of the semantics, in which only transitivity of the domain relations is required, which is a weaker condition compared to compositional consistency. As we have shown, all desired properties of DDL are satisfied under this semantics, including directionality and local inconsistency. There is some trade-off, however, since subsumption only propagates along chains of onto-bridge rules to a satisfactory extent, and it fails to propagate sufficiently along into-chains.

The local inconsistency property is important, in order to guarantee a reasonable behaviour of the semantics also in heterogeneous and dynamic knowledge environments, where occasional inconsistency cannot be excluded and therefore it should be properly dealt with. Therefore, we conclude that the DDL semantics with transitive domain relations is the most reasonable one, given the current state of the art. For this semantics we have also proposed a distributed tableaux reasoning algorithm. As a novel feature, this algorithm introduces domain relation tracking into the tableau, and we believe that it can possibly be

adjusted also for different semantics with restricted domain relations, such as for instance the semantics with injectivity which we have studied before in this thesis.

In the final part of this thesis, we have conducted a comparative study of DDL and three other distributed and modular ontology representation frameworks, including $\mathcal{E}$-connections, P-DL and IDDL. All these formalisms are similar in the sense, that they allow for combination of multiple ontologies, and reasoning over the system that results from the combination. What differentiates all these approaches are the particular constructs they employ for combination of local ontologies. There are three main approaches in ontology combination. In the ontology mapping approach, entities from distinct ontologies are related by special mapping axioms. Thanks to the mapping, knowledge possibly propagates from one ontology to another. This mapping can be directed (DDL) or bidirectional (IDDL). In the ontology linking approach ($\mathcal{E}$-connections), novel constructs called links are used, which semantically behave as inter-ontology roles. By creating a restriction on a link, foreign concepts are possibly used in a local ontology and thus knowledge reuse is facilitated. Ontology importing (P-DL) allows for a distinguished subset of ontology entities to be imported from one ontology to another where they are then used as any other entity. The semantics takes care of importing also the relations that hold amongst these entities in their source ontology and thus knowledge is reused.

At several occasions, relations between these frameworks have been studied (Kutz et al. 2004, Bao et al. 2006*a*, Cuenca Grau & Kutz 2007, Wang et al. 2007) with aim to compare their expressivity. It is a known result, that DDL, under a specific semantics, are reducible into $\mathcal{E}$-connection (Kutz et al. 2004). In this thesis, we have pushed this effort further forward, and we have proven two new reductions between these approaches. We have shown, that P-DL and DDL are interreducible, assuming a special DDL semantics which applies compositional consistency, and some other restrictions on the domain relations. We have also shown that IDDL are reducible into DDL, under a DDL semantics with total and functional domain relations.

In light of these results, we conclude that there are indeed many similarities between these frameworks. However, it cannot be claimed that one of the approaches is strictly more expressive than any other, since as we have shown, the reductions are not unconditional. These findings are well in line with the fact that each of the frameworks was originally designed for a slightly different purpose, and hence it is not surprising that there are also differences in the semantics they employ. In fact, the very same results are helpful in pointing out the actual differences between the frameworks and between the ontology combination paradigms. In this vision, directed mapping can be also implemented by links, but not all the kinds of mapping, for instance, for mapping between roles no reduction is known. Importing is possibly captured by directed mapping and vice versa, however the semantics has to be tailored, as for proper emulation of importing more restrictions have to be placed on the domain relation. Similarly, bidirectional mapping axioms of IDDL can be expressed by bridge rules in DDL, but again, stronger semantics is needed as the one typically used by DDL.

There are open issues put forward by the research conducted in this thesis. Among the problems that are closely related we see:

semantic issues: it is natural to ask, whether a variant of the DDL semantics

can be found, that would handle subsumption propagation along chains of into- and chains of onto-bridge rules, such that it would still satisfy all desirable properties previously postulated for DDL. We conjecture, that this problem can be resolved by further relaxing from compositional consistency, which as we have learned is too strong. It is however questionable, whether such a restriction can be found that is also sufficiently local, so that it will be possible to algorithmically verify it in a fully distributed fashion (i.e., each local reasoning service only needs to communicate with some of its directly adjacent neighbours in order to verify the restriction and not with distant reasoners);

algorithmic issues: a most important open problem is the extension of the distributed tableaux algorithm, that we have introduced, towards more complex DL, especially towards $\mathcal{SHIQ}$, and also the development of a distributed blocking strategy, so that the requirement of acyclic bridge graphs can be dropped. In addition, it would be much interesting to tailor this algorithm also for semantics with other restrictions put on the domain relations, apart from transitivity, which it currently handles. Thus, we could obtain decision support also for the DDL semantics with injectivity, which we have investigated in this thesis and we consider it important. It is worth noting, that by adjusting the algorithm to handle total, functional and role preserving domain relations and consecutive implementation of the algorithm, one would obtain a reasoner for IDDL and P-DL. To our best knowledge, no implementation of such reasoners has been described in the literature;

practical issues: in this thesis we have presented theoretical results with main focus on the semantic issues. Further practical evaluation of our proposals by complexity analysis, implementation and optimization is left for future work.

Besides for these issues, which can be seen as a direct continuation of the current work, there are many other interesting open issues related to DDL and to distributed and modular ontology knowledge representation in general. We pick several of the most interesting open issues. It would be interesting to extend DDL towards more general local languages, especially $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$, which are behind OWL. This is a challenging problem, as nominals are present in these languages. It would be also interesting to further investigate DDL with bridge rules between roles and with heterogeneous bridge rules as well. There are not so many reports published in this line of research and further evaluation of the behaviour of the semantics would be interesting. Finally, as distributed ontology representation frameworks increasingly make use of more and more complex local languages, it would be of some demand to investigate the computational properties and, if possible, to develop reasoning algorithms for logics like $\mathcal{ALCQHI}b$ and even $\mathcal{SHIQ}b$; to our best knowledge, these problems are still open.

Bibliography

Aristotle (circa 40BCE), 'Categoriae', Available from project Gutenberg, at `http://www.gutenberg.org/etext/2412`.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. & Patel-Schneider, P., eds (2003), *The Description Logic Handbook*, Cambridge University Press.

Baader, F., Lutz, C., Sturm, H. & Wolter, F. (2002), 'Fusions of description logics and abstract description systems', *Journal of Artificial Intelligence Research (JAIR)* **16**, 1–58.

Baader, F., Lutz, C. & Suntisrivaraporn, B. (2006), Efficient reasoning in $\mathcal{EL}^+$, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Baader, F. & Sattler, U. (1996), Number restrictions on complex roles in description logics, *in* 'Procs. of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)', Morgan Kaufmann.

Bao, J., Caragea, D. & Honavar, V. (2006*a*), On the semantics of linking and importing in modular ontologies, *in* 'Procs. of ISWC 2006', Vol. 4273 of *LNCS*, Springer.

Bao, J., Caragea, D. & Honavar, V. G. (2006*b*), A distributed tableau algorithm for package-based description logics, *in* 'Procs. of CRR 2006', Riva del Garda, Italy.

Bao, J., Caragea, D. & Honavar, V. G. (2006*c*), A tableau-based federated reasoning algorithm for modular ontologies, *in* 'Procs. of IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)', IEEE Press.

Bao, J., Caragea, D. & Honavar, V. G. (2006*d*), Towards collaborative environments for ontology construction and sharing, *in* 'Procs. of the 2006 International Symposium on Collaborative Technologies and Systems', Las Vegas, Nevada, USA.

Bao, J. & Honavar, V. (2004), Ontology language extensions to support collaborative ontology building, *in* '3rd International Semantic Web Conference (ISWC2004)', Vol. 3298 of *LNCS*, Springer.

Bao, J., Voutsadakis, G., Slutzki, G. & Honavar, V. (2009), Package-based description logics, *in* 'Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization', Vol. 5445 of *LNCS*, Springer, pp. 349–371.

Beckett, D., ed. (2004), *RDF Semantics Specification (Revised)*, W3C Recommendation. Available online, at `http://www.w3.org/TR/rdf-syntax-grammar/`.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001), 'The semantic web', *Scientific American* **284**(5), 34–43.

Borgida, A. & Serafini, L. (2002), Distributed description logics: First results, *in* I. Horrocks & S. Tessaris, eds, 'Procs. of the 2002 Intl. Workshop on Description Logics (DL2002)', CEUR-WS.

Borgida, A. & Serafini, L. (2003), 'Distributed description logics: Assimilating information from peer sources', *Journal of Data Semantics* **1**, 153–184.

Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L. & Stuckenschmidt, H. (2003), C-OWL: Contextualizing ontologies, *in* 'Procs. of the 2nd International Semantic Web Conference (ISWC2003)', pp. 164–179.

Brickley, D. & Guha, R. V., eds (2004), *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation. Available online, at `http://www.w3.org/TR/rdf-schema/`.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M. & Rosati, R. (2005), *DL-Lite*: Tractable description logics for ontologies., *in* 'Procs. of the The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference (AAAI 2005)', pp. 602–607.

Calvanese, D., Giacomo, G. D., Lenzerini, M. & Rosati, R. (2004), Logical foundations of peer-to-peer data integration, *in* 'PODS', pp. 241–251.

Cuenca Grau, B., Horrocks, I., Kazakov, Y. & Sattler, U. (2007), A logical framework for modularity of ontologies, *in* 'IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007', pp. 298–303.

Cuenca Grau, B., Horrocks, I., Kutz, O. & Sattler, U. (2006), Will my ontologies fit together?, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)', pp. 175–182.

Cuenca Grau, B. & Kutz, O. (2007), Modular ontology languages revisited, *in* 'Procs. of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition SWeCKa 2007'.

Cuenca Grau, B., Parsia, B. & Sirin, E. (2004*a*), Tableaux algorithms for $\mathcal{E}$-connections of description logics, Technical report, University of Maryland Institute for Advanced Computer Studies.

Cuenca Grau, B., Parsia, B. & Sirin, E. (2004*b*), Working with multiple ontologies on the semantic web., *in* 'Procs. of ISWC2004', Vol. 3298 of *LNCS*, Springer.

Cuenca Grau, B., Parsia, B. & Sirin, E. (2006), 'Combining OWL ontologies using $\mathcal{E}$-connections', *Journal of Web Semantics* **4**(1), 40–59.

Cuenca Grau, B., Parsia, B., Sirin, E. & Kalyanpur, A. (2005), Automatic partitioning of OWL ontologies using $\mathcal{E}$-connections, *in* 'Procs. of the 2005 International Workshop on Description Logics (DL'05)', Vol. 147 of *CEUR-WS*.

Cuenca Grau, B., Parsia, B., Sirin, E. & Kalyanpur, A. (2006), Modularity of web ontologies, *in* 'Proc. of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)', AAAI Press, pp. 198–209.

Ding, Y. & Haarslev, V. (2006), Tableau caching for description logics with inverse and transitive roles, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Dominé, A., ed. (1998), *Culinaria France*, Könemann Verlagsgesellschaft, Cologne.

Euzenat, J. & Shvaiko, P. (2007), *Ontology Matching*, Springer.

Fagin, R., Halpern, J. Y., Moses, Y. & Vardi, M. Y. (1995), *Reasoning About Knowledge*, MIT Press.

Fokoue, A., Kershenbaum, A. & Ma, L. (2006), $\mathcal{SHIN}$ ABox reduction, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Frank, J., Dziuban, V. & Homola, M. (2010), Sémantický web: niektoré aktuálne výzvy, *in* V. Kvasnička, J. Pospíchal, P. Návrat, P. Lacko & P. Trebatický, eds, 'Umelá inteligencia a kognitívna veda II', Vydavateľstvo STU, Bratislava.

Ghidini, C. & Serafini, L. (1998), Distributed first order logics, *in* 'Frontiers Of Combining Systems 2, Studies in Logic and Computation', Research Studies Press, pp. 121–140.

Ghidini, C. & Serafini, L. (2006*a*), Mapping properties of heterogeneous ontologies, *in* 'Procs. of the 1st International Workshop on Modular Ontologies (WoMo-06)', Vol. 232 of *CEUR WS*.

Ghidini, C. & Serafini, L. (2006*b*), Reconciling concepts and relations in heterogeneous ontologies, *in* 'Procs. of the 3rd European Semantic Web Conference (ESWC 2006)', Vol. 4011 of *LNCS*, Springer.

Ghidini, C., Serafini, L. & Tessaris, S. (2007), On relating heterogeneous elements from different ontologies, *in* 'Procs. of the Sixth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07)', Vol. 4635 of *LNCS*, Springer.

Ghidini, C., Serafini, L. & Tessaris, S. (2008), Complexity of reasoning with expressive ontology mappings, *in* 'Formal Ontology in Information Systems, Procs. of the Fifth International Conference, FOIS 2008', pp. 151–163.

Ghilardi, S., Lutz, C. & Wolter, F. (2006), Did I damage my ontology? A case for conservative extensions in description logics, *in* 'Procs. of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)', AAAI Press, pp. 187–197.

Giunchiglia, F. & Ghidini, C. (1998), Local models semantics, or contextual reasoning = locality + compatibility, *in* A. G. Cohn, L. K. Schubert & S. C. Shapiro, eds, 'Procs. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)', Morgan Kaufmann, pp. 282–291.

Gruber, T. (1993), 'A translation approach to portable ontology specifications', *Knowledge Acquisition* **5**(2), 199–220.

Haarslev, V. & Müller, R. (2001), RACER system description, *in* R. Goré, A. Leitsch & T. Nipkow, eds, 'Automated Reasoning First International Joint Conference, IJCAR 2001 Siena, Italy, June 18–22, 2001 Proceedings', Vol. 2083 of *LNCS*, Springer, pp. 701–706.

Hayes, P., ed. (2004), *RDF Semantics*, W3C Recommendation. Available online, at `http://www.w3.org/TR/rdf-mt/`.

Homola, M. (2007*a*), Alternative characterization of conjunctive bridge rules in distributed description logics (preliminary report), *in* 'Procs. of ISCAM 2007'.

Homola, M. (2007*b*), Distributed description logics revisited, *in* 'Procs. of the 20th International Workshop on Description Logics (DL-2007)', Vol. 250 of *CEUR-WS*.

Homola, M. (2008), Subsumption propagation between remote ontologies in distributed description logic, *in* 'Procs. of the 21st International Workshop on Description Logics (DL2008)', Vol. 353 of *CEUR-WS*.

Homola, M. & Serafini, L. (2008), Towards distributed tableaux reasoning procedure for DDL with increased subsumption propagation between remote ontologies, *in* 'Advances in Ontologies, Procs. of Knowledge Representation Ontology Workshop (KROW 2008)', Vol. 90 of *CRPIT*, Australian Computer Society, Sydney, Australia.

Homola, M. & Serafini, L. (2010*a*), 'Augmenting subsumption propagation in distributed description logics', *Applied Artificial Intelligence* **24**(1–2), 137–174.

Homola, M. & Serafini, L. (2010*b*), Towards formal comparison of ontology linking, mapping and importing, *in* 'Procs. of the 23rd International Workshop on Description Logics (DL2010)', Vol. 573 of *CEUR-WS*.

Horrocks, I. (1998), Using an expressive description logic: FaCT or fiction?, *in* 'Procs. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)', pp. 636–647.

Horrocks, I., Kutz, O. & Sattler, U. (2006), The even more irresistible $\mathcal{SROIQ}$, *in* 'Procs. of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)', AAAI Press, pp. 57–67.

Horrocks, I., Patel-Schneider, P. F. & van Harmelen, F. (2003), 'From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language', *Journal of Web Semantics* **1**(1), 7–26.

Horrocks, I. & Sattler, U. (1999), 'A description logic with transitive and inverse roles and role hierarchies', *Journal of Logic Computation* **9**(3), 385–410.

Horrocks, I. & Sattler, U. (2004), 'Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms', *Artificial Intelligence* **160**(1-2), 79–104.

Horrocks, I. & Sattler, U. (2005), A tableaux decision procedure for $\mathcal{SHOIQ}$, *in* 'Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)', pp. 448–453.

Horrocks, I. & Sattler, U. (2007), 'A tableaux decision procedure for shoiq', *Journal of Automated Reasoning* **39**(3).

Horrocks, I., Sattler, U. & Tobies, S. (1999), Practical reasoning for expressive description logics, *in* 'Procs. of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)', number 1705 *in* 'LNAI', Springer, pp. 161–180.

Horrocks, I., Sattler, U. & Tobies, S. (2000), 'Practical reasoning for very expressive description logics', *Journal of the Interest Group in Pure and Applied Logic* **8**(3), 239–264.

Hudek, A. K. & Weddell, G. (2006), Binary absorption in tableaux-based reasoning for description logics, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Hustadt, U., Motik, B. & Sattler, U. (2004), Reducing shiq-description logic to disjunctive datalog programs, *in* 'Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004'.

Kalfoglou, Y. & Schorlemmer, M. (2005), Ontology mapping: The state of the art, *in* 'Semantic Interoperability and Integration', number 04391 *in* 'Dagstuhl Seminar Proceedings'.

Kazakov, Y. (2008), $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$, *in* 'Procs. of the 11th Conference on Principles of Knowledge Representation and Reasoning (KR 2008)', AAAI Preass.

Kazakov, Y., Sattler, U. & Zolin, E. (2007), How many legs do i have? non-simple roles in number restrictions revisited, *in* 'Proc. of the 14th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'2007)', Vol. 4790 of *Lecture Notes in Computer Science*, Springer.

Kifer, M., Lausen, G. & Wu, J. (1995), 'Logical foundations of object-oriented and frame-based languages.', *Journal of the ACM* **42**(4), 741–843.

Kutz, O. (2004), $\mathcal{E}$-connections and Logics of Distance, PhD thesis, University of Liverpool, UK.

Kutz, O., Lutz, C., Wolter, F. & Zakharyaschev, M. (2003), $\mathcal{E}$-connections of description logics, *in* 'Procs. of the 2003 International Workshop on Description Logics (DL2003)', Vol. 81 of *CEUR-WS*.

Kutz, O., Lutz, C., Wolter, F. & Zakharyaschev, M. (2004), '$\mathcal{E}$-connections of abstract description systems', *Artificial Intelligence* **156**(1), 1–73.

Kutz, O., Wolter, F. & Zakharyaschev, M. (2002), Connecting abstract description systems, *in* 'Procs. of the 8th Conference on Principles of Knowledge Representation and Reasoning (KR 2002)', Morgan Kaufmann, pp. 215–226.

Laozi (circa 6th century BCE), 'Daodejing'. The excerpt cited herein in English is based on a novel translation of Daodejing by Marina Čarnogurská & Egon Bondy "Lao c' – Tao Te t'ing", published in Slovak language by Agentúra Fisher & Formát, 2005.

Loebe, F. (2006), Requirements for logical modules, *in* 'Procs. of the 1st International Workshop on Modular Ontologies (WoMO'06)', Vol. 232 of *CEUR-WS*.

Lutz, C., Walther, D. & Wolter, F. (2007), Conservative extensions in expressive description logics, *in* 'IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007', pp. 453–458.

McGuinness, D. L. & van Harmelen, F., eds (2004), *OWL Web Ontology Language Overview*, W3C Recommendation. Available online, `http://www.w3.org/TR/owl-features/`.

Motik, B., Shearer, R. & Horrocks, I. (2009), 'Hypertableau Reasoning for Description Logics', *Journal of Artificial Intelligence Research* **36**.

Möller, R., Haarslev, V. & Wessel, M. (2006), On the scalability of description logic instance retrieval, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Pan, J. Z., Serafini, L. & Zhao, Y. (2006), Semantic import: An approach for partial ontology reuse, *in* P. Haase, V. Honavar, O. Kutz, Y. Sure & A. Tamilin, eds, 'Procs. of the 1st International Workshop on Modular Ontologies (WoMo-06)', Vol. 232 of *CEUR WS*, Athens, Georgia, USA.

Papadimitriou, C. H. (1994), *Computational Complexity*, Addison Wesley.

Parsia, B. & Cuenca Grau, B. (2005), Generalized link properties for expressive e-connections of description logics, *in* 'Proc. of the Twentieth National Conference on Artificial Intelligence (AAAI-2005)', AAAI, Pittsburgh, Pennsylvania, USA.

Rector, A. L., Nowlan, W. A. & Glowinski, A. (1993), Goals for concept representation in the GALEN project, *in* 'Procs. of the Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC 93)', pp. 414–418.

Schild, K. (1991), A correspondence theory for terminological logics: Preliminary report, *in* 'Procs. of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)', pp. 466–471.

Schlicht, A. & Stuckenschmidt, H. (2008), Distributed resolution for $\mathcal{ALC}$, *in* 'Procs. of the 21st International Workshop on Description Logics (DL2008)', Vol. 353 of *CEUR-WS*.

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

Serafini, L., Borgida, A. & Tamilin, A. (2005), Aspects of distributed and modular ontology reasoning, *in* 'Procs. of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)', pp. 570–575.

Serafini, L. & Homola, M. (2010), Modular knowledge representation and reasoning in the Semantic Web, *in* R. D. Virgilio, F. Giunchiglia & L. Tanca, eds, 'Semantic Web Information Management: A Model-Based Perspective', Springer.

Serafini, L. & Tamilin, A. (2004), Local tableaux for reasoning in distributed description logics, *in* 'Procs. of the 2004 International Workshop on Description Logics (DL2004)', Vol. 104 of *CEUR-WS*.

Serafini, L. & Tamilin, A. (2005), DRAGO: Distributed reasoning architecture for the semantic web, *in* 'Proc. of the Second European Semantic Web Conference (ESWC'05)', Springer.

Serafini, L. & Tamilin, A. (2006), Distributed instance retrieval in heterogeneous ontologies, *in* 'Proc. of the 2nd Italian Semantic Web Workshop Semantic Web Applications and Perspectives (SWAP'05)', Trento, Italy.

Serafini, L. & Tamilin, A. (2007), Instance migration in heterogeneous ontology environments, *in* 'Proc. of the 6th International Semantic Web Conference (ISWC-07)', Busan, Korea.

Sirin, E. & Parsia, B. (2006), Optimizations for answering conjunctive ABox queries, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A. & Katz, Y. (2007), 'Pellet: A practical OWL-DL reasoner', *Journal of Web Semantics* **5**(2), 51–53.

Sowa, J. F. (1999), *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co. See also the book's homepage at `http://www.jfsowa.com/krbook/`.

Spackman, K., Campbell, K. & Cote, R. (1999), SNOMED RT: A reference terminology for health care, pp. 640–644. Fall Symposium Supplement.

Staab, S. & Studer, R., eds (2004), *Handbook on Ontologies*, Springer.

Subrahmanian, V. S. (1994), 'Amalgamating knowledge bases', *ACM Transactions on Database Systems* **19**(2), 291–331.

Tamilin, A. (2007), Distributed Ontological Reasoning: Theory, Algorithms, and Applications, PhD thesis, University of Trento.

Tobies, S. (2000), 'The complexity of reasoning with cardinality restrictions and nominals in expressive description logics', *Journal of Artificial Intelligence Research (JAIR)* **12**.

Tobies, S. (2001), Complexity Results and Practical Algorithms for Logics in Knowledge Representation, PhD thesis, RWTH Aachen, Germany.

Tsarkov, D. & Horrocks, I. (2006), Fact++ description logic reasoner: System description, *in* U. Furbach & N. Shankar, eds, 'Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006. Proceedings', Vol. 4130 of *LNAI*, Springer, pp. 292–297.

Ullman, J. D. (1997), Information integration using logical views, *in* F. N. Afrati & P. G. Kolaitis, eds, 'Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings', Vol. 1186 of *LNCS*, Springer, pp. 19–40.

W3C OWL Working Group, ed. (2009), *OWL 2 Web Ontology Language Document Overview*, W3C Recommendation. Available online, at `http://www.w3.org/TR/owl2-overview/`.

Wang, Y., Haase, P. & Bao, J. (2007), A survey of formalisms for modular ontologies, *in* 'Procs. of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition SWeCKa 2007'.

Zimmermann, A. (2007), Integrated distributed description logics, *in* 'Procs. of 20th International Workshop on Description Logics (DL-2007)', Vol. 250 of *CEUR WS*.

Zimmermann, A. & Le Duc, C. (2008), Reasoning on a Network of Aligned Ontologies, *in* D. Calvanese & G. Lausen, eds, 'Web Reasoning and Rule Systems, Second International Conference, RR 2008, Karlsruhe, Germany, October/November 2008, Proceedings', Vol. 5341 of *Lecture Notes in Computer Science*, Springer, pp. 43–57.

Zuo, M. & Haarslev, V. (2006), High performance absorption algorithms for terminological reasoning, *in* 'Procs. of the 2006 International Workshop on Description Logics (DL'06)'.

List of Figures

List of Tables