Web Semantics: Science, Services and Agents on the World Wide Web 35 (2015) 214–234

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

The description logic $\mathcal{SHIQ}$ with a flexible meta-modelling hierarchy

Regina Motz [a], Edelweis Rohrer [a], Paula Severi [b,*]

[a] Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay

[b] Department of Computer Science, University of Leicester, England, United Kingdom

ARTICLE INFO

Article history:
Received 2 November 2014
Received in revised form
30 April 2015
Accepted 10 May 2015
Available online 19 May 2015

Keywords:
Description logic
Meta-modelling
Meta-concepts
Well founded sets
Consistency
Decidability

ABSTRACT

This work is motivated by a real-world case study where it is necessary to integrate and relate existing ontologies through *meta-modelling*. For this, we introduce the Description Logic $\mathcal{SHIQM}$ which is obtained from $\mathcal{SHIQ}$ by adding statements that equate individuals to concepts in a knowledge base. In this new extension, concepts can be individuals of another concept (called *meta-concept*) which itself can be an individual of yet another concept (called *meta–meta-concept*) and so on. We define an algorithm that checks consistency of $\mathcal{SHIQM}$ by modifying the Tableau algorithm for $\mathcal{SHIQ}$. From the practical point of view, this has the advantage that we can reuse the code of existing OWL reasoners. From the theoretical point of view, it has a similar advantage of reuse. We make use of the existing results and proofs that lead to correctness of the algorithm for $\mathcal{SHIQ}$ in order to prove correctness of the algorithm for $\mathcal{SHIQM}$.

1. Introduction

Our extension of $\mathcal{SHIQ}$ is motivated by a real-world application on geographic objects that requires to reuse existing ontologies and relate them through meta-modelling [1].

Fig. 1 describes a simplified scenario of this application in order to illustrate the meta-modelling relationship. It shows two ontologies separated by a horizontal line. The two ontologies conceptualize the same entities at different levels of granularity. In the ontology above the horizontal line, rivers and lakes are formalized as individuals while in the one below the line they are concepts. If we want to integrate these ontologies into a single ontology (or into an ontology network) it is necessary to interpret the individual *river* and the concept *River* as the same real object. Similarly for *lake* and *Lake*.

Our solution consists in equating the individual *river* to the concept *River* and the individual *lake* to the concept *Lake*. These equalities are called *meta-modelling axioms* and in this case, we say that the ontologies are related through *meta-modelling*. In Fig. 1, meta-modelling axioms are represented by dashed edges. After adding the meta-modelling axioms for rivers and lakes, the concept *HydrographicObject* is now also a *meta-concept* because it is a concept that contains an individual which is also a concept.

The kind of meta-modelling we consider in this paper can be expressed in OWL Full but it cannot be expressed in OWL DL. The fact that it is expressed in OWL Full is not very useful since the meta-modelling provided by OWL Full is so expressive that leads to undecidability [2].

OWL 2 DL has a very restricted form of meta-modelling called *punning* where the same identifier can be used as an individual and as a concept [3]. These identifiers are treated as different objects by the reasoner and it is not possible to detect certain inconsistencies. We next illustrate two examples where OWL would not detect inconsistencies because the identifiers, though they look syntactically equal, are actually different.

Example 1. If we introduce an axiom expressing that *HydrographicObject* is a subclass of *River*, then OWL reasoner will not detect that the interpretation of *River* is not a well founded set (it is a set that belongs to itself).

Example 2. We add two axioms, the first one says that *river* and *lake* as individuals are equal and the second one says that the classes *River* and *Lake* are disjoint. Then OWL reasoner does not detect that there is a contradiction.

In this paper, we consider $\mathcal{SHIQ}$ ($\mathcal{ALCQ}$ with a role hierarchy, inverse and transitive roles) and extend it with *Mboxes*. An Mbox is a set of equalities of the form $a =_{\mathrm{m}} A$ where a is an individual and A is a concept. We call $\mathcal{SHIQM}$ such extension of $\mathcal{SHIQ}$. In our example, we have that $river =_{\mathrm{m}} River$ and these two identifiers are semantically equal, i.e., the interpretations of the individual *river* and the concept *River* are the same. The domain of an interpretation cannot longer consist of only basic objects. It cannot be an

* Corresponding author.

http://dx.doi.org/10.1016/j.websem.2015.05.002
1570-8268/

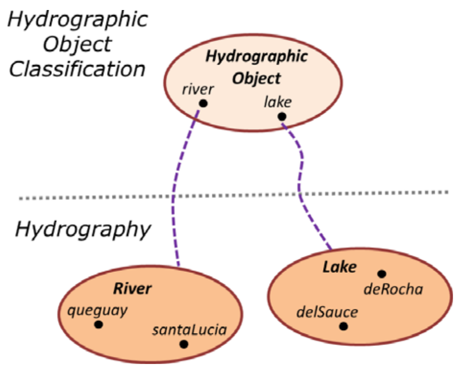

Fig. 1. Two ontologies on hydrography.

arbitrary set either. We require that the domain be a well-founded set. The reason for this is explained as follows. Suppose we have a domain $\Delta^{\mathcal{I}} = \{X\}$ where $X = \{X\}$ is a set that belongs to itself. Intuitively, X is the set

$\{\{\{\ldots\}\}\}$.

Clearly, a set like X should be excluded from our interpretation domain since it cannot represent any real object from our usual applications in Semantic Web (in other areas or aspects of Computer Science, representing such objects is useful [4]). The well-foundedness of our model is not ensured by means of fixing layers beforehand as in [5–8] but it is our reasoner which checks for circularities.

Our approach allows the user to have any number of levels or layers (meta-concepts, meta–meta-concepts and so on). The user does not have to write or know the layer of the concept because the reasoner will infer it for him. In this way, axioms can also naturally mix elements of different layers and the user has the flexibility of changing the status of an individual at any point without having to make any substantial change to the ontology. In a real scenario of evolving ontologies, that need to be integrated, not all individuals of a given concept need to have meta-modelling and hence, they do not have to belong to the same level in the hierarchy.

We define a tableau algorithm for checking consistency of an ontology in $\mathcal{SHIQM}$ by adding *new rules* and a *new condition* to the tableau algorithm for $\mathcal{SHIQ}$. The new rules deal with the equalities and inequalities between individuals with meta-modelling which need to be transferred to the level of concepts as equalities and inequalities between the corresponding concepts. The new condition deals with circularities (with respect to membership) avoiding non well-founded sets such as *River* in Example 1. From the practical point of view, extending tableau for $\mathcal{SHIQ}$ has the advantage that one can easily change and reuse the code of existing OWL reasoners. Moreover, the algorithm follows the same excellent "pay as you go" characteristics as the other DL extensions that provide the foundation of OWL. The "pay as you go" characteristic means that if only the expressiveness of $\mathcal{SHIQ}$ is used, the new algorithm just behaves like the tableau algorithm for $\mathcal{SHIQ}$. In other words, when the meta-modelling features are not needed, then the algorithm behaves just like the regular, first-order $\mathcal{SHIQ}$ algorithm. From the theoretical point of view, extending tableau for $\mathcal{SHIQ}$ allows us to "reuse" and invoke the results on soundness of the tableau algorithm for $\mathcal{SHIQ}$ [9]. This paper is an extension of [10] where we have studied the weaker logic $\mathcal{ALCQ}$ extended with Mboxes. One of the challenges of the present paper is the fact that $\mathcal{SHIQ}$ does not satisfy the finite model property. When the model is finite, it is clear that checking for well-foundedness with respect to the membership relation is decidable. But when the model is infinite this may not longer be true since a non-well founded set that is also infinite may have infinite descendent sequences of the form:

$X_1 \ni X_2 \ni X_3 \ni X_4 \ni \ldots.$

We will show that decidability is maintained for $\mathcal{SHIQM}$ due to the fact that the Mbox is finite. In this paper, we additionally study the problem of inferring the meta-modelling level of an ontology which was not done in [10].

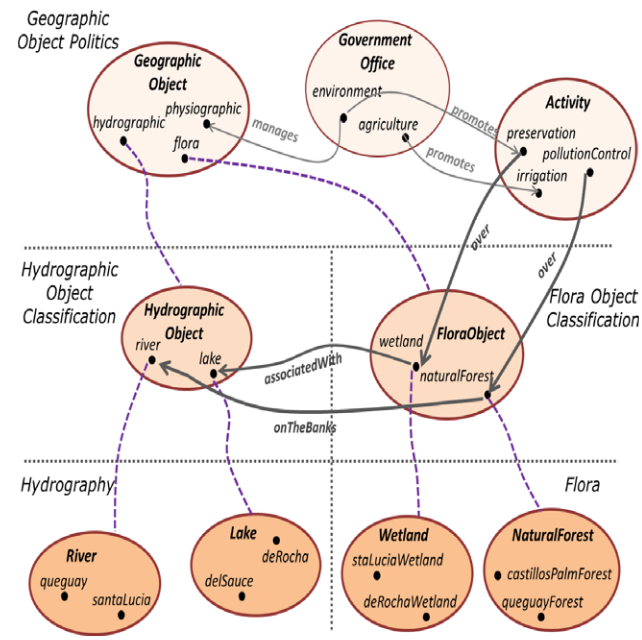

Fig. 2. Case study on geography.

Organization of the paper. The remainder of this paper is organized as follows. Section 2 shows a case study and explains the advantages of our approach. Section 3 recalls the logic $\mathcal{SHIQ}$ and the tableau algorithm for $\mathcal{SHIQ}$ [9]. Section 4 recalls the notions of well-founded relation, sets and the induction (recursion) principle. Section 5 defines the syntax and semantics of $\mathcal{SHIQM}$. Section 6 gives the tableau algorithm for checking consistency. Section 7 proves its correctness. Section 8 studies the problem of inferring the meta-modelling level of an ontology. Section 9 compares our approach with other approaches to meta-modelling in the literature. Finally, Section 10 summarizes the main contributions of this paper and Section 11 explains future work.

2. Case study on geography

In this section, we illustrate some important advantages of our approach through the real-world example on geographic objects presented in the introduction.

Fig. 2 extends the ontology network given in Fig. 1. Ontologies are delimited by two horizontal lines and one vertical line. Concepts are denoted by large ovals and individuals by bullets. Meta-modelling between ontologies is represented by dashed edges. Thinnest arrows denote roles within a single ontology while thickest arrows denote roles from one ontology to another ontology.

Fig. 2 has five separate ontologies. The ontology in the uppermost row conceptualizes the politics about geographic objects, defining *GeographicObject* as a meta–meta-concept, and *Activity* and *GovernmentOffice* as concepts. The ontology in the left middle describes hydrographic objects through the meta-concept *HydrographicObject* and the one in the right middle describes flora objects through the meta-concept *FloraObject*. The two remaining ontologies conceptualize the concrete natural resources at a lower level of granularity through the concepts *River*, *Lake*, *Wetland* and *NaturalForest*.

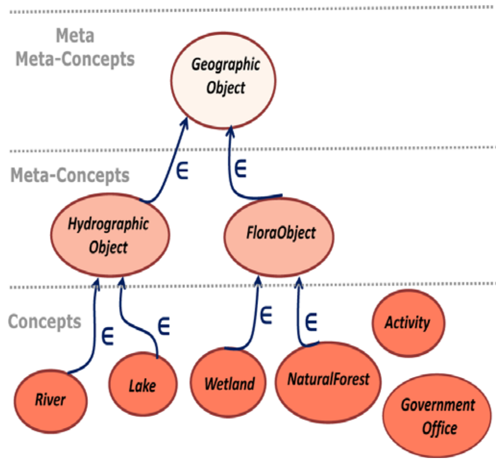

Fig. 3. Meta-modelling hierarchy for the ontology of Fig. 2.

Note that the horizontal lines in Fig. 2 do not separate meta-modelling levels but just ontologies. The ontology "Geographic Object Politics" has the meta–meta-concept *GeographicObject*, whose instances are concepts which have also instances being concepts, but we also have the concepts *GovernmentOffice*and *Activity* whose instances conceptualize atomic objects.

OWL has only one notion of hierarchy which classifies concepts with respect to the inclusion $\sqsubseteq$. Our approach has a new notion of hierarchy, called *meta-modelling hierarchy*, which classifies concepts with respect to the membership relation $\in$. The meta-modelling hierarchy for the concepts of Fig. 2 is depicted in Fig. 3. The concepts are *GovernmentOffice*, *Activity*, *River*, *Lake*, *Wetland* and *NaturalForest*, the meta-concepts are *HydrographicObject* and *FloraObject*, and the meta–meta-concept is *GeographicObject*.

The first advantage of our approach over some previous work concerns the reuse of ontologies when the same conceptual object is represented as an individual in one ontology and as a concept in the other. The identifiers for the individual and the concept will be syntactically different because they belong to different ontologies (with different URIs). Then, the ontology engineer can introduce an equation between these two different identifiers. This contrasts with previous approaches where one has to use the same identifier for an object used as a concept and as an individual. In Fig. 2, *river* and *River* represent the same real object. In order to detect inconsistency and do the proper inferences, one has to be able to equate them.

The second advantage is about the flexibility of the meta-modelling hierarchy. This hierarchy is easy to change by just adding equations. This is illustrated in the passage from Fig. 1 to Fig. 2. Fig. 1 has a very simple meta-modelling hierarchy where the concepts are *River* and *Lake* and the meta-concept is *HydrographicObject*. The rather more complex meta-modelling hierarchy for the ontology of Fig. 2 (see Fig. 3) has been obtained by combining the ontologies of Fig. 1 with other ontologies and by simply adding some few meta-modelling axioms. After adding the meta-modelling equations, the change of the meta-modelling hierarchy is *automatic* and *transparent* to the user. Concepts such as *GeographicObject* will automatically pass to be meta–meta-concepts and roles such as *associatedWith* will automatically pass to be meta-roles, i.e., roles between meta-concepts.

The third advantage is that the level of meta-modelling has no bound, i.e., we can have concepts, meta-concepts, meta–meta-concepts and so on. Fig. 1 has only one level of meta-modelling since there are concepts and meta-concepts. In Fig. 2, there are two levels of meta-modelling since it has concepts, meta-concepts and meta–meta-concepts. If we needed, we could extend it further by adding the equation *santaLucia* $=_m$ *SantaLucia* for some concept *SantaLucia* and this will add a new level in the meta-modelling hierarchy: concepts, meta-concepts, meta–meta-concepts and meta–meta–meta-concepts. Moreover, the user does not have to know the meta-modelling levels, they are transparent for him. Our algorithm detects inconsistencies without burdening the user with syntactic complications such as having to explicitly write the level the concept belongs to.

The fourth advantage is about the possibility of mixing levels of meta-modelling in the definition of concepts and roles. We can have concepts such as *GeographicObject* which has individuals with different levels of meta-modelling. The individual *physiographic* has no meta-modelling at all. Its interpretation belongs to a set of basic objects (level 0). The other two individuals in *GeographicObject* have meta-modelling. They actually represent meta-concepts and their interpretations have level 1. We can build concepts using union or intersection between two concepts of different levels (layers). We can also define roles whose domain and range live in different levels (or layers). For example, in Fig. 2, we have: (i) a role *over* whose domain is just a concept while the range is a meta-concept, (ii) a role *manages* whose domain is just a concept and whose range is a meta–meta-concept.

3. Preliminaries on $\mathcal{SHIQ}$

In this section we recall the Description Logic $\mathcal{SHIQ}$ [9,3]. Horrocks et al. define the notion of tableau as an abstract notion of model [9]. In this section, we introduce the new notion of tableau structure and isomorphism between them. These notions make our proof of correctness for $\mathcal{SHIQM}$ more elegant. The notion of tableau structure is an abstract notion of interpretation. In other words, tableau structure is to the notion of interpretation as tableau is to the notion of model. The notion of tableau structure and isomorphism are inspired in the notions of interpretation (structure) and isomorphism between interpretations for first order logic [11, Section 2.11]. Similar to the case of first order logic, we prove that "isomorphic structures satisfy the same properties".

3.1. Syntax and semantics of $\mathcal{SHIQ}$

We assume we have three pairwise disjoint sets: a set of individuals, a set of atomic concepts and a set of atomic roles. Individuals are denoted by $a, b, \ldots$, atomic concepts by $A, B, \ldots$ and atomic roles by $R, S, \ldots$. The set of atomic roles contains all role names and all inverse of role names (i.e., R^- for any role name R). To avoid considering roles such as R^{--}, the function $Inv(R)$ is defined as follows. $Inv(R) = R^-$ if R is a role name, and $Inv(R) = S$ if $R = S^-$. A role is *transitive* if it has a declaration of the form $Trans(R)$.

Let $\sqsubseteq^*$ be the transitive–reflexive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{Inv(R) \sqsubseteq Inv(S) \mid R \sqsubseteq S \in \mathcal{R}\}$. A role R is a *subrole* of S if $R \sqsubseteq^* S$. A role is *simple* if it is neither transitive nor has any transitive subroles.

Concepts are defined by the following grammar:

$$C, D ::= A \mid \top \mid \bot \mid (\neg C) \mid (C \sqcap D) \mid (C \sqcup D) \mid \\ (\forall R.C) \mid (\exists R.C) \mid (\geqslant n\, S.C) \mid (\leqslant n\, S.C)$$

here n is a non-negative integer and S is a simple role. Concepts are denoted by C, D and atomic concepts by A, B. We omit parenthesis according to the following precedence order of the description logics operators: (i) $\neg$, $\forall$, $\exists$, $\geqslant n$ and $\leqslant n$, (ii) $\sqcap$, (iii) $\sqcup$. Outermost parenthesis are always omitted

An *ontology or knowledge base* $\mathcal{O}$ *in* $\mathcal{SHIQ}$ is a triplet $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ where

1. $\mathcal{T}$, called a *Tbox*, is a finite set of axioms of the form $C \sqsubseteq D$, with C, D any two concepts.

$\mathsf{NNF}(A) = A$ if A is an atomic concept
$\mathsf{NNF}(\neg A) = \neg A$ if A is an atomic concept
$\mathsf{NNF}(\neg\neg C) = \mathsf{NNF}(C)$
$\mathsf{NNF}(C \sqcup D) = \mathsf{NNF}(C) \sqcup \mathsf{NNF}(D)$
$\mathsf{NNF}(C \sqcap D) = \mathsf{NNF}(C) \sqcap \mathsf{NNF}(D)$
$\mathsf{NNF}(\neg(C \sqcup D)) = \mathsf{NNF}(\neg C) \sqcap \mathsf{NNF}(\neg D)$
$\mathsf{NNF}(\neg(C \sqcap D)) = \mathsf{NNF}(\neg C) \sqcup \mathsf{NNF}(\neg D)$
$\mathsf{NNF}(\forall R.C) = \forall R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\exists R.C) = \exists R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\neg\forall R.C) = \exists R.\mathsf{NNF}(\neg C)$
$\mathsf{NNF}(\neg\exists R.C) = \forall R.\mathsf{NNF}(\neg C)$
$\mathsf{NNF}(\leqslant n\, R.C) = \leqslant n\, R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\geqslant n\, R.C) = \geqslant n\, R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\neg \leqslant n\, R.C) = \geqslant (n+1)\, R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\neg \geqslant (n+1)\, R.C) = \leqslant n\, R.\mathsf{NNF}(C)$
$\mathsf{NNF}(\neg \geqslant 0\, R.C) = \bot$

$\mathsf{NNF}(\mathcal{T}) = \bigcup_{C \sqsubseteq D \in \mathcal{T}} \mathsf{NNF}(\neg C \sqcup D)$
$\mathsf{NNF}(\mathcal{A}) = \bigcup_{C(a)\in\mathcal{A}} \mathsf{NNF}(C)(a) \cup \bigcup_{R(a,b)\in\mathcal{A}} R(a,b) \cup \bigcup_{a=b\in\mathcal{A}} a = b \cup \bigcup_{a\neq b\in\mathcal{A}} a \neq b$

Fig. 4. Negation normal form of a concept, a TBox and an ABox.

2. $\mathcal{R}$, called an *Rbox*, is a finite set of role inclusion axioms of the form $R \sqsubseteq S$ and transitive role declarations of the form $Trans(R)$, with R, S atomic roles.
3. $\mathcal{A}$, called an *Abox*, is a finite set of statements of the form $C(a)$, $R(a, b)$, $a = b$ or $a \neq b$.

Note that we changed the standard definition of $\mathcal{SHIQ}$ by adding equalities of the form $a = b$ for individuals a and b in the Abox. There are two reasons for adding equality between individuals. First of all, this is a very useful OWL feature. Second and most important, it makes it evident that equality and difference between individuals play an important role in the presence of meta-modelling since an equality between individuals is transferred into an equality between the corresponding concepts and conversely.

We say that a concept is in *negation normal form* if negation occurs in front of atomic concepts only. Fig. 4 defines a function that computes the negation normal form of a concept, a TBox and an ABox.

We say that C is a *(syntactic) sub-concept* of a concept D if $C \in \mathsf{subcon}(D)$ where subcon is defined as follows.

$$
\begin{aligned}
\mathsf{subcon}(C) &= \{C\} \quad \text{if } C \in \{A, \top, \bot\}\\
\mathsf{subcon}(\neg C) &= \mathsf{subcon}(C) \cup \{\neg C\}\\
\mathsf{subcon}(C \sqcap D) &= \mathsf{subcon}(C) \cup \mathsf{subcon}(D) \cup \{C \sqcap D\}\\
\mathsf{subcon}(C \sqcup D) &= \mathsf{subcon}(C) \cup \mathsf{subcon}(D) \cup \{C \sqcup D\}\\
\mathsf{subcon}(\forall R.C) &= \mathsf{subcon}(C) \cup \{\forall R.C\}\\
\mathsf{subcon}(\exists R.C) &= \mathsf{subcon}(C) \cup \{\exists R.C\}\\
\mathsf{subcon}(\geqslant n\, S.C) &= \mathsf{subcon}(C) \cup \{\geqslant n\, S.C\}\\
\mathsf{subcon}(\leqslant n\, S.C) &= \mathsf{subcon}(C) \cup \{\leqslant n\, S.C\}.
\end{aligned}
$$

Let $\mathsf{clos}(C)$ be the smallest set that contains the concept C (assumed to be in negation normal form) and is closed under (syntactic) sub-concepts and $\sim$ where $\sim C$ is the negation normal form of $\neg C$. We define $\mathbf{R}_{\mathcal{O}}$ as the set of roles occurring in $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{R}$ together with their inverses.

For $\mathcal{O}$ in $\mathcal{SHIQ}$, we define $\mathbf{I}_{\mathcal{O}}$ as the set of individuals occurring in $\mathcal{A}$.

Definition 1 (*Closure of a $\mathcal{SHIQ}$ Ontology*)**.** Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be in negation normal form.

We define the *closure of the ontology* $\mathcal{O}$ as

$$\mathsf{clos}(\mathcal{O}) = \bigcup_{C(a)\in\mathcal{A} \text{ or } C\in\mathcal{T}} \mathsf{clos}(C).$$

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (sometimes we drop the super-index when the name of the interpretation is clear from the context and write just Δ), called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$ which maps every concept to a subset of Δ and every role to a subset of $\Delta \times \Delta$ such that, for all concepts C, D, roles R, S, and non-negative integers n, the following equations are satisfied, where $\sharp X$ denotes the cardinality of a set X:

$$
\begin{aligned}
&(R^-)^{\mathcal{I}} = \{(x, y) \mid (y, x) \in R^{\mathcal{I}}\}\\
&(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}\\
&(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}\\
&(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}\\
&(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}\\
&(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.(x, y) \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}\\
&(\geqslant n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}\\
&(\leqslant n\, R.C)^{\mathcal{I}} = \{x \mid \sharp\{y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}.
\end{aligned}
$$

Note that the definition of interpretation for $\mathcal{SHIQ}$ does not require that the domain Δ is a set of only basic objects [9].

An interpretation $\mathcal{I}$ *satisfies a TBox* $\mathcal{T}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each $C \sqsubseteq D$ in $\mathcal{T}$.

An interpretation $\mathcal{I}$ *satisfies an RBox* $\mathcal{R}$ iff (i) $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ for each $R \sqsubseteq S$ in $\mathcal{R}$ and (ii) if $\{(x, y), (y, z)\} \subseteq R^{\mathcal{I}}$ then $(x, z) \in R^{\mathcal{I}}$ for each $Trans(R)$ in $\mathcal{R}$.

An interpretation $\mathcal{I}$ *satisfies an ABox* $\mathcal{A}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for each $C(a)$ in $\mathcal{A}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ for each $R(a, b)$ in $\mathcal{A}$, $a^{\mathcal{I}} = b^{\mathcal{I}}$ for each $a = b$ in $\mathcal{A}$ and $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for each $a \neq b$ in $\mathcal{A}$.

An interpretation $\mathcal{I}$ is a *model* of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ iff it satisfies $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$.

We say that an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ is *consistent* if there exists a model of $\mathcal{O}$.

Some Description Logics weaker than $\mathcal{SHIQ}$ have the *finite model property*, i.e., a consistent knowledge base always admits a model with finite domain. $\mathcal{SHIQ}$, however, does not have the finite model property. This is caused by the combination of cardinality restrictions, role hierarchies, transitive and inverse roles in $\mathcal{SHIQ}$. An example of a $\mathcal{SHIQ}$ knowledge base that does not satisfy the finite model property is shown in [12].

3.2. Checking consistency for $\mathcal{SHIQ}$

We now recall the tableau algorithm for checking consistency of an ontology in $\mathcal{SHIQ}$ [9,3]. Horrocks et al. assume the existence of a universal role (a transitive super-role of all roles occurring in $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$, and their respective inverses) to internalize the TBox [9]. We give a presentation of the tableau algorithm that does not internalize the TBox (and hence, we do not need to add the universal role in our syntax). Instead of internalizing the Tbox, we add a Tbox rule as in [3]. Our initialization is a bit different too because our Aboxes can contain equalities between individuals.

A *completion forest* $\mathcal{F}$ for a $\mathcal{SHIQ}$ knowledge base consists of

1. a set of nodes, labelled with individual names or variable names (fresh individuals which do not belong to the ABox),
2. directed edges between some pairs of nodes,
3. for each node labelled x, a set $\mathcal{F}(x)$ of concept expressions,
4. for each pair of nodes x and y, a set $\mathcal{F}(x, y)$ containing role names or inverses of role names, and
5. two relations between nodes, denoted by $\approx$ and $\not\approx$. These relations keep record of the equalities and inequalities of nodes in the algorithm. The relation $\approx$ is assumed to be reflexive, symmetric and transitive while $\not\approx$ is assumed to be symmetric. We also assume that the relation $\not\approx$ is *compatible with* $\approx$, i.e., if $x' \approx x$ and $x \not\approx y$ then $x' \not\approx y$ for all x, x', y. In the algorithm, every time we add a pair in $\approx$, we close $\approx$ under reflexivity, symmetry and transitivity. Moreover, every time we add a pair in either $\not\approx$ or $\approx$, we close $\not\approx$ under compatibility with $\approx$.

Nodes labelled with individual names, which are present in the input ABox, are named *root nodes*.

If nodes x and y are connected by an edge (x, y) with $R \in \mathcal{F}(x, y)$ and $R \sqsubseteq^* S$, then y is called an *S-successor* of x and x is called an *S-predecessor* of y. If y is an S-successor or an $Inv(S)$-predecessor of x, then y is called an *S-neighbour* of x. A node y is a *successor* (resp. *predecessor* or *neighbour*) of x if it is an S-successor (resp. S-predecessor or S-neighbour) of x for some role S. Finally, *ancestor* is the transitive closure of predecessor.

A node is *blocked* iff it is not a root node and it is either directly or indirectly blocked. A node x is *directly blocked* iff none of its ancestors are blocked, and it has ancestors x', y and y' such that

1. y is not a root node and
2. x is a successor of x' and y is a successor of y' and
3. $\mathcal{F}(x) = \mathcal{F}(y)$ and $\mathcal{F}(x') = \mathcal{F}(y')$ and
4. $\mathcal{F}(x', x) = \mathcal{F}(y', y)$.

In this case, we say that y *blocks* x.

A node y is *indirectly blocked* iff one of its ancestors is blocked, or it is a successor of a node x and $\mathcal{F}(x, y) = \emptyset$; the latter condition avoids wasted expansions after an application of the rule $\leqslant$-rule.

In Definition 2 and Fig. 5, we assume that $\mathcal{T}$ and $\mathcal{A}$ have already been converted into negation normal form (see Fig. 4).

Definition 2 (*Initialization*)**.** The *initial completion forest* for $\mathcal{O}$ is defined by the following procedure.

1. For each individual a in the ontology ($a \in \mathcal{A}$) set $a \approx a$. Those individuals in the ontology that do not appear in an equality axiom of the form $a = b$, will not be in the relation $\approx$.
2. For each $a = b \in \mathcal{A}$, set $a \approx b$. We also choose an individual as a representative of each equivalence class.
3. For each $a \neq b$ in $\mathcal{A}$, set $a \not\approx b$.
4. For each $a \in \mathcal{A}$, we do the following:
   (a) in case a is a representative of an equivalence class then set $\mathcal{F}(a) = \{C \mid C(a') \in \mathcal{A}, a \approx a'\}$;
   (b) in case a is not a representative of an equivalence class then set $\mathcal{F}(a) = \emptyset$.
5. For all $a, b \in \mathcal{A}$ that are representatives of some equivalence class, if $\{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\} \neq \emptyset$ then create an edge from a to b and set $\mathcal{F}(a, b) = \{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\}$.

Note that in case a is not a representative of an equivalence class and it has some axiom $C(a)$, we set $\mathcal{F}(a) = \emptyset$ because we do not want to apply any expansion rule to $\mathcal{F}(a)$. The expansion rules will only be applied to the representative of the equivalence class of a. Something similar happens in the $\leq$-root rule where we also choose a canonical representative z and set $z \approx y$ and $\mathcal{F}(y)$ to be empty. In case of the $\leq$-rule, we also choose a canonical representative z and set $z \approx y$ where z may not be a root node. However, in this case we avoid wasted expansions by setting $\mathcal{F}(x, y) = \emptyset$ (see the definition of indirectly blocked node).

The tableau algorithm for $\mathcal{SHIQ}$ without ABoxes has only one $\leqslant$-rule that deals with the case $\leqslant n\, R.C \in \mathcal{F}(x)$ [13]. However, for checking consistency for a knowledge base $(\mathcal{T}, \mathcal{A})$ that includes an ABox $\mathcal{A}$ in $\mathcal{SHIQ}$, the $\leqslant$-rule is split into two rules: (i) the $\leqslant$-root-rule is applied when y, z are both root nodes and (ii) the $\leqslant$-rule is applied otherwise. For an explanation of these rules, we refer the reader to [9].

Definition 3 (*Contradiction*)**.** $\mathcal{F}$ has a *contradiction* if either

- A and $\neg A$ belongs to $\mathcal{F}(x)$ for some atomic concept A and node x or
- there are nodes x and y such that $x \not\approx y$ and $x \approx y$.
- there is a node x such that $\leqslant n\, S.C \in \mathcal{F}(x)$, and x has $n + 1$ S-neighbours $y_1, \ldots, y_{n+1}$ with $C \in \mathcal{F}(y_i)$, $y_i \not\approx y_j$ for all $i, j \in \{1, \ldots, n + 1\}$ with $i \neq j$.

Definition 4 ($\mathcal{SHIQ}$*-Complete*)**.** A forest $\mathcal{F}$ is $\mathcal{SHIQ}$*-complete* (or just *complete*) if none of the rules of Fig. 5 is applicable.

$\sqcap$-rule:
If x is not indirectly blocked, $C \sqcap D \in \mathcal{F}(x)$ and $\{C, D\} \not\subseteq \mathcal{F}(x)$ then add $\{C, D\}$ to $\mathcal{F}(x)$.

$\sqcup$-rule:
If x is not indirectly blocked, $C \sqcup D \in \mathcal{F}(x)$ and $\{C, D\} \cap \mathcal{F}(x) = \emptyset$ then add either C or D to $\mathcal{F}(x)$.

$\exists$-rule:
If x is not blocked, $\exists R.C \in \mathcal{F}(x)$ and x has no R-neighbour y with $C \in \mathcal{F}(y)$ then
1. add a new node with label y (where y is a new node label),
2. set $\mathcal{F}(x, y) = \{R\}$,
3. set $\mathcal{F}(y) = \{C\}$.

$\forall$-rule:
If x is not indirectly blocked, $\forall R.C \in \mathcal{F}(x)$ and x has an R-neighbour y with $C \notin \mathcal{F}(y)$ then add C to $\mathcal{F}(y)$.

Tbox-rule:
If x is not indirectly blocked, C is a TBox statement and $C \notin \mathcal{F}(x)$, then add C to $\mathcal{F}(x)$.

trans-rule:
If x is not indirectly blocked, $\forall S.C \in \mathcal{F}(x)$, S has a transitive subrole R, and x has an R-neighbour y with $\forall R.C \notin \mathcal{F}(y)$, then add $\forall R.C$ to $\mathcal{F}(y)$.

choose-rule:
If x is not indirectly blocked, $\leqslant n\, S.C \in \mathcal{F}(x)$ or $\geqslant n\, S.C \in \mathcal{F}(x)$ and there is an S-neighbour y of x with $\{C, \sim C\} \cap \mathcal{F}(y) = \emptyset$, then add either C or $\sim C$ to $\mathcal{F}(y)$.

$\geqslant$-rule:
If x is not blocked, $\geqslant n\, S.C \in \mathcal{F}(x)$ and there are no n S-neighbours $y_1, \ldots, y_n$ of x with $C \in \mathcal{F}(y_i)$, $y_i \not\approx y_j$ for $i, j \in \{1, \ldots n\}$ and $i \neq j$, then
1. create n new nodes $y_1, \ldots y_n$.
2. set $\mathcal{F}(x, y_i) = \{S\}$, $\mathcal{F}(y_i) = \{C\}$ and $y_i \not\approx y_j$ for $i, j \in \{1, \ldots n\}$, $i \neq j$.

$\leqslant$-rule:
If x is not indirectly blocked, $\leqslant n\, S.C \in \mathcal{F}(x)$, there are more than n S-neighbours y_i of x with $C \in \mathcal{F}(y_i)$, and x has two S-neighbours y, z such that y is neither a root node nor an ancestor of z, $y \not\approx z$ does not hold, and $C \in \mathcal{F}(y) \cap \mathcal{F}(z)$, then besides setting $y \approx z$, we also do:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(z)$,
2. if z is an ancestor of x, then add $\{R^- \mid R \in \mathcal{F}(x, y)\}$ to $\mathcal{F}(z, x)$,
3. if z is not an ancestor of x, then add $\mathcal{F}(x, y)$ to $\mathcal{F}(x, z)$,
4. set $\mathcal{F}(x, y) = \emptyset$, and
5. set $u \not\approx z$ for all u with $u \not\approx y$.

$\leqslant$-root-rule:
If $\leqslant n\, S.C \in \mathcal{F}(x)$, there are more than n S-neighbours y_i of x with $C \in \mathcal{F}(y_i)$, and x has two S-neighbours y, z which are both root nodes, $y \not\approx z$ does not hold, and $C \in \mathcal{F}(y) \cap \mathcal{F}(z)$, then besides setting $y \approx z$, we also do:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(z)$,
2. for all directed edges from y to some w, create an edge from z to w if it does not exist with $\mathcal{F}(z, w) = \emptyset$.
3. add $\mathcal{F}(y, w)$ to $\mathcal{F}(z, w)$,
4. for all directed edges from some w to y, create an edge from w to z if it does not exist with $\mathcal{F}(w, z) = \emptyset$,
5. add $\mathcal{F}(w, y)$ to $\mathcal{F}(w, z)$,
6. set $\mathcal{F}(y) = \emptyset$ and remove all edges from/to y.
7. set $u \not\approx z$ for all u with $u \not\approx y$.

Fig. 5. Expansion rules for $\mathcal{SHIQ}$.

After initialization, the tableau algorithm proceeds by non-deterministically applying the *expansion rules for* $\mathcal{SHIQ}$ given in Fig. 5.

The algorithm says that the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ is consistent iff the expansion rules can be applied in such a way they yield a complete forest $\mathcal{F}$ without contradictions. Otherwise the algorithm says that it is inconsistent. Note that due to the non-determinism of the algorithm, implementations of it have to guess the choices and possibly have to backtrack to choice points if a choice already made has led to a contradiction. The algorithm stops when we reach *some* $\mathcal{F}$ that is complete and has no contradiction or when all the choices have yield a forest with contradictions.

In spite of the fact that the following lemma will not be needed later for the proof of termination and correctness of our $\mathcal{SHIQM}$ algorithm, we include it here to add clarity to the exposition.

Lemma 1 (*Termination of Tableau for* $\mathcal{SHIQ}$). *The tableau algorithm for* $\mathcal{SHIQ}$ *terminates when started with a* $\mathcal{SHIQ}$*-knowledge base* $(\mathcal{T}, \mathcal{R}, \mathcal{A})$.

The above lemma is proved in [9, Lemma 3].

3.3. Tableau structure and isomorphism

We now introduce the notion of tableau structure as an abstract notion of interpretation. We need to make the distinction between tableau (the abstract model) and tableau structure (the abstract interpretation) to express the sentence "isomorphic structures satisfy the same properties".

Definition 5 (*Tableau Structure*). Let $\mathbf{I}$ and $\mathbf{R}$ be some arbitrary sets of individuals and roles respectively. We say that $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a *tableau structure* for $\mathbf{I}$ and $\mathbf{R}$ if

- $\mathbf{S}$ is a non-empty set,
- $\mathcal{L}$ maps each element in $\mathbf{S}$ to a set of concepts,
- $\mathcal{E} : \mathbf{R} \to 2^{\mathbf{S}\times\mathbf{S}}$ maps each role to a set of pairs of elements in $\mathbf{S}$, and
- $\mathcal{J} : \mathbf{I} \to \mathbf{S}$ maps individuals to elements in $\mathbf{S}$.

We now give the definition of tableau for $\mathcal{SHIQ}$ by making some minor alterations to [9, Definition 3]. We added the properties (P12) and (P16). These properties are necessary because we have given a slightly different presentation of tableau algorithm for $\mathcal{SHIQ}$. (P12) takes into account the Tbox and (P16) accommodates the equalities in the Abox.

Definition 6 (*Tableau for* $\mathcal{SHIQ}$). Let $\mathcal{O}$ be a $\mathcal{SHIQ}$-knowledge base of the form $(\mathcal{T}, \mathcal{R}, \mathcal{A})$. We say that $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a *tableau for a* $\mathcal{SHIQ}$ *ontology* $\mathcal{O}$ if $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a tableau structure for $\mathbf{I}_{\mathcal{O}}$ and $\mathbf{R}_{\mathcal{O}}$; and it also satisfies the following properties for all $s, t, x, y \in \mathbf{S}$, $a, b \in \mathbf{I}_{\mathcal{O}}$, $R, S \in \mathbf{R}_{\mathcal{O}}$ and concepts C, C_1, C_2:

(P1) if $C \in \mathcal{L}(s)$, then $\neg C \notin \mathcal{L}(s)$.
(P2) if $C_1 \sqcap C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ and $C_2 \in \mathcal{L}(s)$.
(P3) if $C_1 \sqcup C_2 \in \mathcal{L}(s)$, then $C_1 \in \mathcal{L}(s)$ or $C_2 \in \mathcal{L}(s)$.
(P4) if $\forall S.C \in \mathcal{L}(s)$ and $(s,t) \in \mathcal{E}(S)$, then $C \in \mathcal{L}(t)$.
(P5) if $\exists S.C \in \mathcal{L}(s)$, then there is some $t \in \mathbf{S}$ such that $(s,t) \in \mathcal{E}(S)$ and $C \in \mathcal{L}(t)$.
(P6) if $\forall S.C \in \mathcal{L}(s)$ and $(s,t) \in \mathcal{E}(R)$ for some $R \sqsubseteq^* S$ with $Trans(R)$, then $\forall R.C \in \mathcal{L}(t)$.
(P7) $(x,y) \in \mathcal{E}(R)$ iff $(y,x) \in \mathcal{E}(Inv(R))$.
(P8) if $(s,t) \in \mathcal{E}(R)$ and $R \sqsubseteq^* S$, then $(s,t) \in \mathcal{E}(S)$.
(P9) if $\leqslant n\, S.C \in \mathcal{L}(s)$, then $\sharp\{t \mid (s,t) \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\} \leq n$.
(P10) if $\geqslant n\, S.C \in \mathcal{L}(s)$, then $\sharp\{t \mid (s,t) \in \mathcal{E}(S) \text{ and } C \in \mathcal{L}(t)\} \geq n$.
(P11) if $\leqslant n\, S.C \in \mathcal{L}(s)$ or $\geqslant n\, S.C \in \mathcal{L}(s)$, and $(s,t) \in \mathcal{E}(S)$, then $C \in \mathcal{L}(t)$ or $\sim C \in \mathcal{L}(t)$.
(P12) if $C \in \mathcal{T}$ then $C \in \mathcal{L}(s)$ for all $s \in \mathbf{S}$.
(P13) if $C(a) \in \mathcal{A}$, then $C \in \mathcal{L}(\mathcal{J}(a))$.
(P14) if $R(a,b) \in \mathcal{A}$, then $(\mathcal{J}(a), \mathcal{J}(b)) \in \mathcal{E}(R)$.
(P15) if $a \neq b \in \mathcal{A}$, then $\mathcal{J}(a) \neq \mathcal{J}(b)$.
(P16) if $a = b \in \mathcal{A}$, then $\mathcal{J}(a) = \mathcal{J}(b)$.

The proof of the following lemma is very similar to [9, Lemma 2]. For the proof of correctness of our tableau algorithm for $\mathcal{SHIQM}$, we only need the if-direction of this lemma. The converse is not going to be used later.

Lemma 2. *Let* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$. *A* $\mathcal{SHIQ}$*-ontology* $\mathcal{O}$ *is consistent iff there exists a* $\mathcal{SHIQ}$*-tableau for* $\mathcal{O}$.

Proof. Direction $\Leftarrow$. Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ be a tableau for a $\mathcal{SHIQ}$ ontology $\mathcal{O}$. Then, we consider the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}} := \mathbf{S}$ and

$$A^{\mathcal{I}} := \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\}$$
$$a^{\mathcal{I}} := \mathcal{J}(a)$$
$$R^{\mathcal{I}} := \begin{cases} \mathcal{E}(R)^+ & \text{if } Trans(R) \\ \mathcal{E}(R) \cup \bigcup\limits_{P \sqsubseteq^* R, P \neq R} P^{\mathcal{I}} & \text{otherwise} \end{cases}$$

where $\mathcal{E}(R)^+$ is the transitive closure of $\mathcal{E}(R)$.

We prove that $\mathcal{I}$ is a model of the $\mathcal{SHIQ}$-ontology. One has to prove first that

$$C \in \mathcal{L}(s) \text{ implies } s \in C^{\mathcal{I}}. \quad (1)$$

The proof of this fact is exactly as in [9, Lemma 2] since we have not changed the syntax for concepts (we have only added equality axioms in the Abox and not internalized the Tbox). It follows from (1) and (P12) that $\mathcal{I}$ satisfies the Tbox. Similarly, by (1) and (P13), $\mathcal{I}$ satisfies all individual assertions $C(a)$ of the Abox and by (1) and (P14), $\mathcal{I}$ satisfies all role assertions $R(a,b)$ of the Abox. The interpretation $\mathcal{I}$ satisfies all assertions $a = b$ of the Abox by (P16) without any need of (1). Similarly, it satisfies the assertions $a \neq b$ by (P15).

Direction $\Rightarrow$. Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model of $\mathcal{O}$, we define a tableau structure $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ for $\mathcal{O}$ as follows.

$$\mathbf{S} := \Delta^{\mathcal{I}}$$
$$\mathcal{L}(s) := \{C \in \mathsf{clos}(\mathcal{O}) \mid s \in C^{\mathcal{I}}\}$$
$$\mathcal{E}(R) := R^{\mathcal{I}}$$
$$\mathcal{J}(a) := a^{\mathcal{I}}$$

where clos is given in Definition 1. It is easy to prove that $\mathbb{T}$ is a tableau for $\mathcal{O}$.

We now introduce the notion of isomorphism between tableau structures (abstract interpretations).

Definition 7 (*Isomorphism*). Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ and $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ be two tableau structures for some $\mathbf{I}$ and $\mathbf{R}$. An isomorphism between $\mathbb{T}$ and $\mathbb{T}'$ is a bijective function $f : \mathbf{S} \to \mathbf{S}'$ such that

1. $C \in \mathcal{L}(s)$ if and only if $C \in \mathcal{L}'(f(s))$.
2. $(s,t) \in \mathcal{E}(R)$ if and only if $(f(s), f(t)) \in \mathcal{E}'(R)$.
3. $f(\mathcal{J}(a)) = \mathcal{J}'(a)$.

For all $s, t \in \mathbf{S}$, $a \in \mathbf{I}$, $R \in \mathbf{R}$ and concepts C. We say that $\mathbb{T}$ and $\mathbb{T}'$ are *isomorphic* if there exists an isomorphism between them.

We now prove that isomorphic tableau structures (isomorphic abstract interpretations) satisfy the same properties and hence, if one is a model so is the other one.

Lemma 3. *Let $\mathcal{O}$ be a $\mathcal{SHIQ}$ ontology, $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ and $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ be two isomorphic tableau structures for $\mathbf{I}_{\mathcal{O}}$ and $\mathbf{R}_{\mathcal{O}}$.*

1. *$\mathbb{T}$ satisfies a property (Pi) iff $\mathbb{T}'$ satisfies the same property (Pi) for $1 \le i \le 16$.*
2. *$\mathbb{T}$ is a tableau for $\mathcal{O}$ iff $\mathbb{T}'$ is a tableau for $\mathcal{O}$.*

Proof. It is enough to prove only one direction. Assume that $\mathbb{T}$ satisfies property (P5). We show that $\mathbb{T}'$ satisfies (P5) too. The rest are similar. Suppose $\exists S.C \in \mathcal{L}'(s)$. By the first clause in Definition 7, $\exists S.C \in \mathcal{L}(f^{-1}(s))$. Since $\mathbb{T}$ satisfies (P5), there is some $t \in \mathbf{S}$ such that $(f^{-1}(s), t) \in \mathcal{E}(S)$ and $C \in \mathcal{L}(t)$. By the second clause in Definition 7, $(s, f(t)) \in \mathcal{E}'(S)$ and by the first one, $C \in \mathcal{L}'(f(t))$.

3.4. Abstract canonical model for $\mathcal{SHIQ}$

We define the notion of "abstract canonical model" which is built from a complete forest without contradictions [9]. The domain $\mathbf{S}$ of this abstract canonical model is the set of paths instead of the set of nodes of the forest. For Description Logics that enjoy the finite model property (e.g., $\mathcal{ALCQ}$), the domain of the canonical model is the set of nodes of the forest. However, this does not work for $\mathcal{SHIQ}$ because it does not have the finite model property.

We start by recalling the notions of path and tail. Intuitively an element in the domain $\mathbf{S}$ of the canonical model corresponds to a path in $\mathcal{F}$ from some root node to some node that is not blocked, and which goes only via non-root nodes. More precisely, a *path* is a sequence of pairs of nodes of $\mathcal{F}$ of the form $p = \left[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}\right]$. For such a path we define $\mathsf{Tail}(p) = x_n$ and $\mathsf{Tail}'(p) = x'_n$. With $\left[p \mid \frac{x_{n+1}}{x'_{n+1}}\right]$, we denote the path $\left[\frac{x_0}{x'_0}, \ldots, \frac{x_n}{x'_n}, \frac{x_{n+1}}{x'_{n+1}}\right]$. The set $\mathsf{Paths}(\mathcal{F})$ is defined inductively as follows.

- For a root node a in $\mathcal{F}$ which is a representative:
  - $\left[\frac{a}{a}\right] \in \mathsf{Paths}(\mathcal{F})$
- For a path $p \in \mathsf{Paths}(\mathcal{F})$ and a node z in $\mathcal{F}$ which is a representative of some equivalence class:
  - if z is a successor of $\mathsf{Tail}(p)$ and z is neither blocked nor a root node, then $\left[p \mid \frac{z}{z}\right] \in \mathsf{Paths}(\mathcal{F})$, or
  - if, for some node y in $\mathcal{F}$, y is a successor of $\mathsf{Tail}(p)$ and z blocks y, then $\left[p \mid \frac{z}{y}\right] \in \mathsf{Paths}(\mathcal{F})$.

Note that we slightly change the definition of path given in [9, Lemma 4] making it explicit that we consider representatives.

Definition 8 ($\mathcal{SHIQ}$ *Canonical Structure*)**.** Let $\mathcal{F}$ be a completion forest. We define the *canonical tableau structure* $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ built from $\mathcal{F}$ as follows.

$$\mathbf{S} = \mathsf{Paths}(\mathcal{F})$$

$$\mathcal{L}(p) = \mathcal{F}(\mathsf{Tail}(p))$$

$$\begin{aligned}\mathcal{E}(R) = &\left\{\left(p, \left[p\middle|\frac{x}{x'}\right]\right) \in \mathbf{S}\times\mathbf{S} \mid x' \text{ is an } R\text{-successor of } \mathsf{Tail}(p)\right\}\\ &\cup\left\{\left(\left[q\middle|\frac{x}{x'}\right], q\right) \in \mathbf{S}\times\mathbf{S} \mid x' \text{ is an } Inv(R)\text{-successor of } \mathsf{Tail}(q)\right\}\\ &\cup\left\{\left(\left[\frac{a}{a}\right], \left[\frac{b}{b}\right]\right) \in \mathbf{S}\times\mathbf{S} \mid a, b \text{ are representative root nodes and } b \text{ is an } R\text{-neighbour of } a\right\}\end{aligned}$$

$$\mathcal{J}(a) = \begin{cases} \left[\dfrac{a}{a}\right] & \text{if } a \text{ is itself a representative} \\ \left[\dfrac{b}{b}\right] & \text{if } b \text{ is the representative of } a \approx b. \end{cases}$$

In the following lemma, the properties (P1)–(P12) do not depend on the initialization and rely only on the hypothesis that $\mathcal{F}$ is complete.

Lemma 4. *If $\mathcal{F}$ is a $\mathcal{SHIQ}$-complete forest without contradictions then the canonical tableau structure built from $\mathcal{F}$ satisfies the properties* (P1)–(P12).

Proof. The properties (P1)–(P11) are proved in [9, Lemma 4]. We only have to prove the property (P12). By Definition 8 and the definition of $\mathsf{Paths}(\mathcal{F})$, for all $p \in \mathbf{S} = \mathsf{Paths}(\mathcal{F})$ we have that $\mathcal{L}(p) = \mathcal{F}(\mathsf{Tail}(p)) = \mathcal{F}(x)$ for some node x in $\mathcal{F}$ which is not blocked and is a representative of an equivalence class. So, if $C \in \mathcal{T}$ (hypothesis of (P12)), as $\mathcal{F}$ is $\mathcal{SHIQ}$-complete, by the Tbox-rule $C \in \mathcal{F}(x) = \mathcal{L}(p)$.

Besides assuming that $\mathcal{F}$ is complete, the following lemma has the hypothesis that $\mathcal{F}$ is obtained by applying the expansion rules to the forest of the initialization.

Lemma 5 ($\mathcal{SHIQ}$ *Abstract Canonical Model*)**.** *Let $\mathcal{O}$ be a $\mathcal{SHIQ}$-knowledge base $(\mathcal{T}, \mathcal{R}, \mathcal{A})$. If the expansion rules for $\mathcal{SHIQ}$ are applied to $\mathcal{O}$ and yield a complete forest $\mathcal{F}$ without contradictions then the canonical tableau structure built from $\mathcal{F}$ is a tableau for $\mathcal{O}$.*

Proof. The properties (P13)–(P15) are proved in [9, Lemma 4]. We only have to prove (P16). If $a = b \in \mathcal{A}$ (hypothesis of (P16)), in the initial completion forest for $\mathcal{O}$, $a \approx b$. The expansion rules can change the representative of the equivalence class but we will always have that the representative will be some c such that $c \approx a \approx b$. So, by Definition 8 we have that $\mathcal{J}(a) = \left[\frac{c}{c}\right] = \mathcal{J}(b)$.

In this section, we included what will be needed later to prove correctness of the algorithm for $\mathcal{SHIQM}$. Strictly speaking, since we modified the $\mathcal{SHIQ}$-tableau algorithm from [9] by adding equality axioms between individuals, one should re-do the proofs of soundness and completeness. Soundness is easy to prove since it follows directly from Lemmas 2 and 5. The proof of completeness is not included here because we do not need it later for proving correctness of the tableau algorithm for $\mathcal{SHIQM}$ and actually, completeness of the tableau algorithm for $\mathcal{SHIQ}$ is a consequence of Theorem 4 (completeness of the tableau for $\mathcal{SHIQM}$) since a $\mathcal{SHIQ}$-ontology is a particular case of a $\mathcal{SHIQM}$-ontology when the Mbox is empty. When the Mbox is empty the tableau algorithm for $\mathcal{SHIQM}$ behaves just like the algorithm for $\mathcal{SHIQ}$ since the new rules for meta-modelling are never applied.

4. Preliminaries on well-founded sets and relations

In this section, we recall the notions of well-founded relation and set as well as the induction and recursion principles [14,15,4].

Definition 9 (*Well-Founded Relation*)**.** Let X be a set and $\prec$ a binary relation on X.

1. Let $Y \subseteq X$. We say that $m \in Y$ is a *minimal element* of Y if there is no $y \in Y$ such that $y \prec m$.
2. We say that $\prec$ is *well-founded* (on X) if for all $Y \neq \emptyset$ such that $Y \subseteq X$, we have that Y has a minimal element.

Note that in the general definition above the relation $\prec$ does not need to be transitive.

Definition 10 (*Well-Founded Set*)**.** A set X is *well-founded* if the set membership relation $\in$ is well-founded on the set X.

Lemma 6. *The order $\prec$ is well-founded on X iff there are no infinite $\prec$-decreasing sequences, i.e., there is no $\langle x_n\rangle_{n\in\mathbb{N}}$ such that $x_{n+1} \prec x_n$ and $x_n \in X$ for all $n \in \mathbb{N}$.*

The proof of the above lemma can be found in [15].

Now we have the above lemma, we can introduce the following notation.

Definition 11 (*Maximal Length*)**.** Let $\prec$ be a well-founded relation on X. Then, $\mathsf{maxl}^{\prec}(x)$ is the *maximal length* of all descending $\prec$-sequences starting from $x \in X$.

Definition 12 (*Cycle*)**.** We say that $\prec$ *has a cycle (on X)* if there exists $x_1, \ldots, x_n \in X$ such that $x_1 \prec x_2 \prec \ldots \prec x_{n-1} \prec x_n \prec x_1$.

In particular, we also have the following lemma:

Lemma 7. *Let X be a finite set. Then,* $\prec$ *is well-founded on X iff it does not have a cycle, i.e., there are no* $x_1, \ldots, x_n \in X$ *such that* $x_1 \prec x_2 \prec \ldots \prec x_{n-1} \prec x_n \prec x_1$.

As a consequence of Lemma 6, we also have that:

1. If X is a well-founded set then $X \notin X$.
2. If X is a well-founded set then it cannot contain an infinite $\in$-decreasing sequence, i.e., there is no $\langle x_n \rangle_{n \in \mathbb{N}}$ such that $x_{n+1} \in x_n$ and $x_n \in X$ for all $n \in \mathbb{N}$.

The following sets will be used to define the domain of the models of an ontology in $\mathcal{SHIQM}$.

Definition 13 (S_n *for* $n \in \mathbb{N}$)**.** Given a non empty set S_0 of atomic objects, we define S_n by induction on $\mathbb{N}$ as follows: $S_{n+1} = S_n \cup \mathcal{P}(S_n)$.

It is easy to prove that $S_n \subseteq S_{n+1}$ for all $n \in \mathbb{N}$.

A set $X \subseteq S_n$ can contain elements x such that $x \in S_i$ for any $i \leq n$. In the case study presented before, this means that elements with different levels of meta-modelling can coexist in a set $X \subseteq S_n$, e.g., the set of geographic objects in Fig. 2 has two elements with meta-modelling and one with no meta-modelling at all.

Lemma 8 (*Well-Founded Domain*)**.** *The sets* S_n *are well-founded.*

The above lemma is proved by induction on n using Lemma 6.

The following lemma will be used in Section 8.

Lemma 9. *If* $x_1 \in x_2 \in \cdots \in x_{n-1} \in x_n \in S_m$ *then* $n \leq m$.

Proof. The proof is by induction on m. Suppose $m = 0$. Then, S_0 has only basic objects and $n = 0$. Suppose $m > 0$. Then, $x_n \in S_m = S_{m-1} \cup \mathcal{P}(S_{m-1})$. Either $x_n \in S_{m-1}$ or $x_n \subseteq S_{m-1}$. In the first case, by induction hypothesis, we get that $n \leq m - 1$. In the second case, $x_{n-1} \in S_{m-1}$, and by induction hypothesis we get $n - 1 \leq m - 1$. In both cases, we have that $n \leq m$.

An important reason that well-founded relations are interesting is because we can apply the induction and recursion principles, e.g., [15]. In this paper both principles will be used to prove correctness of the Tableau algorithm for $\mathcal{SHIQM}$.

Definition 14 (*Induction Principle*)**.** If $\prec$ is a well-founded relation on X, φ is some property of elements of X, and we want to show that $\varphi(x)$ holds for all elements $x \in X$, it suffices to show that:
if $x \in X$ *and* $\varphi(y)$ *is true for all* $y \in X$ *such that* $y \prec x$*, then* $\varphi(x)$ *must also be true.*

Definition 15 (*Function Restriction*)**.** The restriction of a function $f : X \to Y$ to a subset X' of X is denoted as $f \restriction_{X'}$ and defined as follows.

$$f \restriction_{X'} = \{(x, f(x)) \mid x \in X'\}.$$

On par with induction, well-founded relations also support construction of objects by recursion.

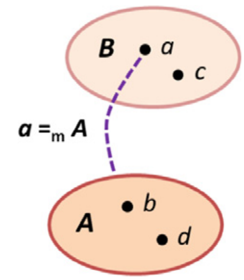

Fig. 6. Meta-modelling axiom example.

Definition 16 (*Recursion Principle*)**.** If $\prec$ is a well-founded relation on X and F a function that assigns an object $F(x, g)$ to each pair of an element $x \in X$ and a function g on the initial segment $\{y \in X \mid y \prec x\}$ of X. Then there is a unique function G such that for every $x \in X$,

$$G(x) = F(x, G \restriction_{\{y \in X \mid y \prec x\}}).$$

5. The description logic $\mathcal{SHIQM}$

In this section we introduce the new description logic $\mathcal{SHIQM}$, with the aim of expressing meta-modelling in a knowledge base. Our notion of meta-modelling allows us to equate individuals to atomic concepts. This notion of meta-modelling is very expressive as illustrated by the case study of Section 2.

Definition 17 (*Meta-Modelling Axiom*)**.** A *meta-modelling axiom* is a statement of the form $a =_{\mathsf{m}} A$ where a is an individual and A is an atomic concept. We pronounce $a =_{\mathsf{m}} A$ as *a corresponds to A through meta-modelling*. An *Mbox* $\mathcal{M}$ is a finite set of meta-modelling axioms.

In Fig. 6, the meta-modelling axiom $a =_{\mathsf{m}} A$ express that the individual a corresponds to the concept A through meta-modelling.

We define $\mathcal{SHIQM}$ by keeping the same syntax for concept expressions as for $\mathcal{SHIQ}$.

An *ontology or a knowledge base* $\mathcal{O}$ *in* $\mathcal{SHIQM}$ is a tuple $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ where $\mathcal{T}, \mathcal{R}, \mathcal{A}$ and $\mathcal{M}$ are a Tbox, Rbox, Abox and an Mbox respectively. The set of all individuals with meta-modelling of an ontology is denoted by $\mathsf{dom}(\mathcal{M})$.

Fig. 7 shows the Tbox, Abox and Mbox of the ontology that corresponds to Fig. 1. Fig. 8 shows the Tbox, Rbox, Abox and Mbox of the ontology discussed in Section 2.

The following definition clarifies what "corresponds through meta-modelling" means.

Definition 18 (*Satisfiability of Meta-Modelling*)**.** An interpretation $\mathcal{I}$ *satisfies* (or it is a *model of*) $a =_{\mathsf{m}} A$ if $a^{\mathcal{I}} = A^{\mathcal{I}}$. An interpretation $\mathcal{I}$ *satisfies* (or it is *model of*) $\mathcal{M}$ if it satisfies each statement in $\mathcal{M}$.

We can see that in Fig. 6 the interpretation of the individual a is the same as that of the concept A, that is: $a^{\mathcal{I}} = A^{\mathcal{I}} = \{b, d\}$.

Unlike $\mathcal{SHIQ}$, the semantics of $\mathcal{SHIQM}$ makes use of the structured domain elements. In order to give semantics to meta-modelling, the domain has to consists of basic objects, sets of objects, sets of sets of objects and so on.

Definition 19 (*Model of an Ontology in* $\mathcal{SHIQM}$)**.** An interpretation $\mathcal{I}$ is a *model of an ontology* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ *in* $\mathcal{SHIQM}$ (denoted as $\mathcal{I} \models \mathcal{O}$) if the following holds:

1. the domain Δ of the interpretation is a *subset* of some S_n for some $n \in \mathbb{N}$.

Tbox

$River \sqcap Lake \sqsubseteq \bot$

Mbox

$river =_{\mathrm{m}} River$
$lake =_{\mathrm{m}} Lake$

Abox

$HydrographicObject(river)$
$HydrographicObject(lake)$
$River(queguay)$
$River(santaLucia)$
$Lake(deRocha)$
$Lake(delSauce)$

Fig. 7. Tbox, Rbox, Abox and Mbox for Fig. 1.

2. $\mathcal{I}$ is a model of the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ in $\mathcal{SHIQ}$.
3. $\mathcal{I}$ is a model of $\mathcal{M}$.

In the first part of Definition 19 we restrict the domain of an interpretation in $\mathcal{SHIQM}$ to be a *subset* of S_n. The domain Δ can now contain sets since the set S_n is defined recursively using the power set operation. Note that S_0 does not have to be the same for all models of an ontology.

The second part of Definition 19 refers to the $\mathcal{SHIQ}$-ontology without the Mbox axioms. In the third part of the definition, we add another condition that the model must satisfy considering the meta-modelling axioms. This condition restricts the interpretation of an individual that has a corresponding concept through meta-modelling to be equal to the concept interpretation.

Example 3. We define a model for the ontology of Fig. 7 where

$S_0 = \{queguay, santaLucia, deRocha, delSauce\}$.

The interpretation is defined on the individuals with meta-modelling and the corresponding atomic concepts to which they are equated as follows:

$$river^{\mathcal{I}} = River^{\mathcal{I}} = \{queguay, santaLucia\}$$
$$lake^{\mathcal{I}} = Lake^{\mathcal{I}} = \{deRocha, delSauce\}$$

and on the remaining atomic concept which does not appear on the MBox the interpretation is defined as follows:

$$HydrographicObject^{\mathcal{I}} = \{river^{\mathcal{I}}, lake^{\mathcal{I}}\} = \{\{queguay, santaLucia\}, \{deRocha, delSauce\}\}.$$

Definition 20 (*Consistency in $\mathcal{SHIQM}$*)**.** We say that an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ is *consistent* if there exists a model of $\mathcal{O}$.

The ontologies of Figs. 7 and 8 without any further axioms are consistent.

Note that the domain Δ does not have to be closed under unions, i.e., if $X, Y \in \Delta$ then $X \cup Y \in \Delta$. We only need that if $X, Y \subseteq \Delta$ then $X \cup Y \subseteq \Delta$.
The following example illustrates our treatment of negation.

Example 4. The ontology obtained by adding the assertion $\neg Wetland(river)$ to the ontology of Fig. 8 is consistent. Since we have only one universe Δ, *river* belongs to the complement of *Wetland*. If we had a domain for each meta-modelling level, then by adding $\neg Wetland(river)$ the ontology would become inconsistent. This is because $(\neg Wetland)^{\mathcal{I}} = \Delta_0 \setminus Wetland^{\mathcal{I}} \subseteq \Delta_0$ where Δ_0 consists of only basic objects. But $river^{\mathcal{I}} \in \Delta_1 = \mathcal{P}(\Delta_0)$ and since $\Delta_1 \cap \Delta_0 = \emptyset$, we have that $river^{\mathcal{I}} \notin (\neg Wetland)^{\mathcal{I}}$.

Tbox

$GovernmentOffice \sqsubseteq \exists manages.GeographicObject$
$Activity \sqsubseteq \forall over.(HydrographicObject \sqcup FloraObject)$
$FloraObject \sqsubseteq \forall associatedWith.HydrographicObject$
$River \sqcap Lake \sqsubseteq \bot$

Rbox

$onTheBanks \sqsubseteq associatedWith$

Abox

$GeographicObject(hydrographic)$
$GeographicObject(physiographic)$
$GeographicObject(flora)$
$GovernmentOffice(environment)$
$GovernmentOffice(agriculture)$
$Activity(preservation)$
$Activity(irrigation)$
$Activity(pollutionControl)$
$manages(environment, physiographic)$
$promotes(environment, preservation)$
$promotes(agriculture, irrigation)$
$HydrographicObject(river)$
$HydrographicObject(lake)$
$FloraObject(wetland)$
$FloraObject(naturalForest)$
$over(preservation, wetland)$
$over(pollutionControl, naturalForest)$
$onTheBanks(naturalForest, river)$
$associatedWith(wetland, lake)$
$River(queguay)$
$River(santaLucia)$
$Lake(deRocha)$
$Lake(delSauce)$
$Wetland(staLuciaWetland)$
$Wetland(deRochaWetland)$
$NaturalForest(castillosPalmForest)$
$NaturalForest(queguayForest)$

Mbox

$river =_{\mathrm{m}} River$
$lake =_{\mathrm{m}} Lake$
$wetland =_{\mathrm{m}} Wetland$
$naturalForest =_{\mathrm{m}} NaturalForest$
$hydrographic =_{\mathrm{m}} HydrographicObject$
$flora =_{\mathrm{m}} FloraObject$

Fig. 8. Tbox, Rbox, Abox and Mbox for Fig. 2.

The approaches in the literature that have a domain for each meta-modelling level forbid the assertion $\neg Wetland(river)$ at the syntactic level [5,7,8]. One can write $E(X)$ only when X is at stratum i and E is at stratum $i+1$.

We now give several examples of inconsistent ontologies.

Example 5. We add now the axiom $HydrographicObject \sqsubseteq River$ to the ontology of Fig. 7. It is easy to see that the $\mathcal{SHIQ}$ ontology without the Mbox is consistent. However, when we include the Mbox it is not longer consistent. Suppose we have a model $\mathcal{I}$. Then,

$$River^{\mathcal{I}} = river^{\mathcal{I}} \in HydrographicObject^{\mathcal{I}} \subseteq River^{\mathcal{I}}.$$

That is, the set $River^{\mathcal{I}}$ is a non well-founded set, since belongs to itself. This contradicts the first clause of Definition 19.

The following example illustrates how the second and third conditions of Definition 19 interact.

Example 6. If in the ontology of Fig. 7 we add the axiom

$$river = lake$$

the $\mathcal{SHIQ}$ ontology without the Mbox is consistent. However, the $\mathcal{SHIQM}$ ontology with the Mbox is not consistent because *River* and *Lake* are non-empty and disjoint.

In the above example, we see that the equality $river = lake$ between individuals is "transferred into" an equality between the corresponding concepts, i.e., $River \equiv Lake$. The following three examples illustrate that the transference can be done in the reverse order as well, i.e., from concepts to individuals. In the second example, the equality $river = lake$ is not explicit in the ontology but it is inferred because we have a functional property.

Example 7. If in the ontology of Fig. 8 we now add the axioms

$$hydrographic \neq flora \tag{2}$$

$$HydrographicObject \equiv FloraObject. \tag{3}$$

Since there is no axiom that says that *HydrographicObject* and *FloraObject* are disjoint, the $\mathcal{SHIQ}$-ontology without the Mbox is consistent. However, the $\mathcal{SHIQM}$-ontology with the Mbox is not consistent. Suppose towards a contradiction that there exists a model $\mathcal{I}$ of this ontology. It follows from Definition 19 that $\mathcal{I}$ should satisfy the meta-modelling axioms, i.e.,

$$hydrographic^{\mathcal{I}} = HydrographicObject^{\mathcal{I}}$$

$$flora^{\mathcal{I}} = FloraObject^{\mathcal{I}}.$$

The interpretation $\mathcal{I}$ should also satisfy the axioms (2) and (3). This is clearly a contradiction. Hence, the ontology is indeed inconsistent.

Example 8. We consider the ontology of Fig. 8 extended with $Wetland \equiv NaturalForest$ and the statement that *associatedWith* is functional.

As before, the $\mathcal{SHIQ}$-ontology without the Mbox is consistent. However, the $\mathcal{SHIQM}$-ontology with the Mbox is not consistent. Suppose towards a contradiction that there exists a model $\mathcal{I}$ of this ontology. Using the fact that $\mathcal{I}$ should satisfy the MBox axioms, we have that

$$wetland^{\mathcal{I}} = naturalForest^{\mathcal{I}}.$$

It follows from the fact that *onTheBanks* is a subrole of *associatedWith* and the functionality of the role *associatedWith* that

$$river^{\mathcal{I}} = lake^{\mathcal{I}}.$$

Then, the interpretations of their corresponding concepts by meta-modelling must also be equal, i.e.,

$$River^{\mathcal{I}} = Lake^{\mathcal{I}}.$$

But this is not possible because the above two sets are disjoint as well as non-empty.

Example 9. We consider the ontology of Fig. 8 extended with

$$Wetland \equiv NaturalForest$$

$$preservation \neq pollutionControl$$

$$FloraObject \sqsubseteq \leqslant 1\, over^{-}.Activity.$$

Then, in presence of meta-modelling there is also an inconsistency. Suppose towards a contradiction that $\mathcal{I}$ is a model of this ontology. Then $\mathcal{I}$ should satisfy the TBox axiom $Wetland \equiv NaturalForest$ as well as $wetland = naturalForest$ by meta-modelling. Since $over^{-}$ is functional, we can deduce that $\mathcal{I}$ also satisfies $preservation = pollutionControl$ which contradicts the fact that $\mathcal{I}$ should also satisfy the second axiom above.

Definition 21 (*Logical Consequence in $\mathcal{SHIQM}$*)**.** We say that $\mathcal{S}$ is a *logical consequence of* $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ (denoted as $\mathcal{O} \models \mathcal{S}$) if all models of $\mathcal{O}$ are also models of $\mathcal{S}$ where $\mathcal{S}$ is any of the following $\mathcal{SHIQM}$ statements, i.e.,

$$C \sqsubseteq D \quad R \sqsubseteq S \quad Trans(R) \quad C(a) \quad R(a, b) \quad a =_{\mathrm{m}} A$$

$$a = b \quad a \neq b.$$

It is possible to infer new knowledge in the ontology with the meta-modelling that is not possible without it as illustrated by Examples 5–9. Example 6 show that an equality $a = b$ between individuals is transferred as an equality $A \equiv B$ between concepts when $a =_{\mathrm{m}} A$ and $b =_{\mathrm{m}} B$. In other words, if $a = b$ is deduced from the knowledge base, so is $A \equiv B$. Examples 7–9 all show the converse, i.e., if $A \equiv B$ is deduced from the knowledge base, so is $a = b$. This transference of equalities is expressed formally in the following lemma.

Lemma 10 (*Equality Transference*)**.** *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a knowledge base, $\mathcal{O} \models a =_{\mathrm{m}} A$ and $\mathcal{O} \models b =_{\mathrm{m}} B$.*

1. *If $\mathcal{O} \models a = b$ then $\mathcal{O} \models A \equiv B$.*
2. *If $\mathcal{O} \models A \equiv B$ then $\mathcal{O} \models a = b$.*

The proof of the above lemma is immediate since a, b, A and B are all interpreted as the same object.

Remark 1. The above two properties are called *intensional regularity* and *extensionality* respectively by Homola et al. [7]. The HiLog style semantics [2,16,7,8] does not satisfy extensionality (the second property of transference) [7]. As already observed by Motik [2], HiLog semantics satisfies intensional regularity (the first property of transference).

Lemma 11. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a consistent ontology. Then, $\mathcal{O} \not\models a =_{\mathrm{m}} \top$.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{O}$. Suppose that $\mathcal{O} \models a =_{\mathrm{m}} \top$. Then, $a^{\mathcal{I}} \in \Delta = a^{\mathcal{I}}$. This contradicts the well-foundedness of Δ.

Definition 22 (*Meta-Concept*)**.** We say that C is a meta-concept in $\mathcal{O}$ if there exists an individual a such that $\mathcal{O} \models C(a)$ and $\mathcal{O} \models a =_{\mathrm{m}} A$.

Then, C is a meta–meta-concept if there exists an individual a such that $\mathcal{O} \models C(a)$, $\mathcal{O} \models a =_{\mathrm{m}} A$ and A is a meta-concept. Note that a meta–meta-concept is also a meta-concept.
We have some new inference problems:

1. *Meta-modelling*. Find out whether $a =_{\mathrm{m}} A$ or not.
2. *Meta-concept*. Find out whether C is a meta-concept or not.

Most inference problems in Description Logic can be reduced to satisfiability by applying a standard result in logic which says that a formula ϕ is a semantic consequence of a set of formulas Γ if and only if $\Gamma \cup \neg\phi$ is not satisfiable. The above two problems can be reduced to satisfiability following this general idea. For the first problem, note that since $a \neq_m A$ is not directly available in the syntax, we have replaced it by $a \neq b$ and $b =_{\mathrm{m}} A$ which is an equivalent statement to the negation of $a =_{\mathrm{m}} A$ and can be expressed in $\mathcal{SHIQM}$.

Lemma 12. *$\mathcal{O} \models a =_{\mathrm{m}} A$ if and only if for some new individual b, $\mathcal{O} \cup \{a \neq b, b =_{\mathrm{m}} A\}$ is unsatisfiable.*

Proof. First we prove the $\Rightarrow$ direction. Suppose towards a contradiction that there exists a model $\mathcal{I}$ of $\mathcal{O}$ such that $\mathcal{I} \models$

$\mathcal{O} \cup \{a \neq b, b =_{\mathsf{m}} A\}$. Then, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ and $b^{\mathcal{I}} = A^{\mathcal{I}}$. Since $\mathcal{O} \models a =_{\mathsf{m}} A$, we have that $a^{\mathcal{I}} = A^{\mathcal{I}}$. Hence $a^{\mathcal{I}} = A^{\mathcal{I}} = b^{\mathcal{I}}$. This contradicts the fact that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

We now prove the $\Leftarrow$ direction by transposition. Suppose that $\mathcal{O} \not\models a =_{\mathsf{m}} A$. Then for some model $\mathcal{I}$ of $\mathcal{O}$, $a^{\mathcal{I}} \neq A^{\mathcal{I}}$. We introduce a new individual b such that $b^{\mathcal{I}} = A^{\mathcal{I}}$ and clearly, $b^{\mathcal{I}} \neq a^{\mathcal{I}}$. Hence, $\mathcal{O} \cup \{a \neq b, b =_{\mathsf{m}} A\}$ is satisfiable.

Lemma 13. *A concept C is a meta-concept if and only if for some individual a we have that $\mathcal{O} \cup \{\neg C(a)\}$ is unsatisfiable and for some new individual b, $\mathcal{O} \cup \{a \neq b, b =_{\mathsf{m}} A\}$ is unsatisfiable.*

Proof. By Definition 22, C is a meta-concept iff $\mathcal{O} \models C(a)$ and $\mathcal{O} \models a =_{\mathsf{m}} A$. It is easy to see that $\mathcal{O} \models C(a)$ is equivalent to the statement that $\mathcal{O} \cup \{\neg C(a)\}$ is unsatisfiable.

6. Checking consistency in $\mathcal{SHIQM}$

In this section we will define a tableau algorithm for checking consistency of an ontology in $\mathcal{SHIQM}$ by extending the tableau algorithm for $\mathcal{SHIQ}$. From the practical point of view, extending tableau for $\mathcal{SHIQ}$ has the advantage that one can easily change and reuse the code of existing OWL reasoners.

The tableau algorithm for $\mathcal{SHIQM}$ is defined by adding three expansion rules and a condition to the tableau algorithm for $\mathcal{SHIQ}$. The new expansion rules deal with the equalities and inequalities between individuals with meta-modelling which need to be transferred to the level of concepts as equalities and inequalities between the corresponding concepts. The new condition deals with circularities avoiding sets that belong to themselves and more generally, avoiding non well-founded sets.

Definition 23 (*Cycles*). We say that the completion forest $\mathcal{F}$ has a cycle with respect to $\mathcal{M}$ if there exist a sequence of meta-modelling axioms $A_0 =_{\mathsf{m}} a_0, A_1 =_{\mathsf{m}} a_1, \ldots, A_n =_{\mathsf{m}} a_n$ all in $\mathcal{M}$ such that

$$\begin{array}{ll} A_1 \in \mathcal{F}(x_0) & x_0 \approx a_0 \\ A_2 \in \mathcal{F}(x_1) & x_1 \approx a_1 \\ \vdots & \vdots \\ A_n \in \mathcal{F}(x_{n-1}) & x_{n-1} \approx a_{n-1} \\ A_0 \in \mathcal{F}(x_n) & x_n \approx a_n. \end{array}$$

Example 10. Suppose we have an ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ with two individuals a and b, the individual assignments: $B(a)$ and $A(b)$; and the meta-modelling axioms:

$$a =_{\mathsf{m}} A \qquad b =_{\mathsf{m}} B.$$

In the completion forest, $\mathcal{F}(a) = \{B\}$ and $\mathcal{F}(b) = \{A\}$ has a cycle since $A \in \mathcal{F}(b)$ and $B \in \mathcal{F}(a)$.

Initialization for the completion forest from an ontology in $\mathcal{SHIQM}$ is nearly the same as for $\mathcal{SHIQ}$. The nodes of the initial completion forest will be created from individuals that occur in the Abox as well as in the Mbox.

In Definition 24 and Fig. 9, we assume that $\mathcal{T}$ and $\mathcal{A}$ have already been converted into negation normal form (see Fig. 4).

Definition 24 (*Initialization*). The *initial completion forest* for $\mathcal{O}$ is defined by the following procedure.

1. For each individual a in the ontology ($a \in \mathcal{A} \cup \mathcal{M}$) set $a \approx a$.
2. For each $a = b \in \mathcal{A}$, set $a \approx b$. We also choose an individual as a representative of each equivalence class.
3. For each $a \neq b$ in $\mathcal{A}$, set $a \not\approx b$.
4. For each $a \in \mathcal{A} \cup \mathcal{M}$, we do the following:
   (a) in case a is a representative of an equivalence class then set $\mathcal{F}(a) = \{C \mid C(a') \in \mathcal{A}, a \approx a'\}$;
   (b) in case a is not a representative of an equivalence class then set $\mathcal{F}(a) = \emptyset$.
5. For all $a, b \in \mathcal{A} \cup \mathcal{M}$ that are representatives of some equivalence class, if $\{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\} \neq \emptyset$ then create an edge from a to b and set $\mathcal{F}(a, b) = \{R \mid R(a', b') \in \mathcal{A}, a \approx a', b \approx b'\}$.

$\approx$-rule:
Let $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ in $\mathcal{M}$. If $a \approx b$ and $A \sqcup \neg B, B \sqcup \neg A$ does not belong to $\mathcal{T}$ then add $A \sqcup \neg B, B \sqcup \neg A$ to $\mathcal{T}$.

$\not\approx$-rule:
Let $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ in $\mathcal{M}$. If $a \not\approx b$ and there is no root node z such that $A \sqcap \neg B \sqcup B \sqcap \neg A \in \mathcal{F}(z)$ then create a new root node z with $\mathcal{F}(z) = \{A \sqcap \neg B \sqcup B \sqcap \neg A\}$

close-rule:
Let $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ in $\mathcal{M}$ where $a \approx x$, $b \approx y$, x and y are their respective representatives of the equivalence classes. If neither $x \approx y$ nor $x \not\approx y$ then we add either $x \approx y$ or $x \not\approx y$. In the case $x \approx y$, we also do the following:
1. add $\mathcal{F}(y)$ to $\mathcal{F}(x)$,
2. for all directed edges from y to some w, create an edge from x to w if it does not exist with $\mathcal{F}(x, w) = \emptyset$,
3. add $\mathcal{F}(y, w)$ to $\mathcal{F}(x, w)$,
4. for all directed edges from some w to y, create an edge from w to x if it does not exist with $\mathcal{F}(w, x) = \emptyset$,
5. add $\mathcal{F}(w, y)$ to $\mathcal{F}(w, x)$,
6. set $\mathcal{F}(y) = \emptyset$ and remove all edges from/to y.

Fig. 9. Additional expansion rules for $\mathcal{SHIQM}$.

After initialization, the tableau algorithm proceeds by non-deterministically applying the expansion rules for $\mathcal{SHIQM}$. The *expansion rules for $\mathcal{SHIQM}$* are obtained by adding the rules of Fig. 9 to the expansion rules for $\mathcal{SHIQ}$.

Blocking is defined as for $\mathcal{SHIQ}$ but the definition of root nodes is different. In $\mathcal{SHIQM}$, the root nodes are the nodes in the Abox, the Mbox and the ones created by the $\not\approx$-rule. As a consequence of this, nodes created by the $\not\approx$-rule cannot be blocked.

We explain the intuition behind the new expansion rules. If $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ then the individuals a and b represent concepts. Any equality at the level of individuals should be transferred as an equality between concepts and similarly with the difference.

The $\approx$-rule transfers the equality $a \approx b$ to the level of concepts by adding two statements to the Tbox which are equivalent to $A \equiv B$. This rule is necessary to detect the inconsistency of Example 6 where the equality *river* $=$ *lake* is transferred as an equality $River \equiv Lake$ between concepts. A particular case of the application of the $\approx$-rule is when $a =_{\mathsf{m}} A$ and $a =_{\mathsf{m}} B$. In this case, the algorithm also adds $A \equiv B$. Actually, it adds an equivalent concept which is in negation normal form (see Fig. 4).

The $\not\approx$-rule is similar to the $\approx$-rule. However, in the case that $a \not\approx b$, we cannot add $A \not\equiv B$ because the negation of $\equiv$ is not directly available in the language. So, what we do is to replace it by an equivalent statement, i.e., add an element z that witnesses this difference. Again, note that the concepts we added to the ABox are in negation normal form (see Fig. 4).

The rules $\approx$ and $\not\approx$ are not sufficient to detect all inconsistencies. With only these rules, we could not detect the inconsistency of Example 8. The idea is that we also need to transfer the equality $A \equiv B$ between concepts as an equality $a \approx b$ between individuals. However, here we face a delicate problem. It is not enough to transfer the equalities that are in the Tbox. We also need to transfer the semantic consequences, e.g., $\mathcal{O} \models A \equiv B$. Unfortunately, a recursive call of the form $\mathcal{O} \models A \equiv B$ is not possible. Otherwise we will be captured in a vicious circle[1] since the problem of finding out the semantic consequences is reduced to the one of satisfiability. The solution to this problem is to explicitly try either $a \approx b$ or $a \not\approx b$. This is exactly what the close-rule does. The close-rule adds either $a \approx b$ or $a \not\approx b$. It is similar to the choose-rule which adds either C or $\neg C$. This works because we are working in Classical Logic and we have the law of excluded middle. For a model $\mathcal{I}$ of the ontology, we have that either $a^{\mathcal{I}} = b^{\mathcal{I}}$ or $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. Since the tableau algorithm works with representatives, we also have to be careful how we equate two individuals or make them different.

Note that the application of the tableau algorithm to a $\mathcal{SHIQM}$ knowledge base $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ changes the Tbox as well as the completion forest $\mathcal{F}$.

Definition 25 (*$\mathcal{SHIQM}$-Complete*)**.** We say that $(\mathcal{T}, \mathcal{F})$ is $\mathcal{SHIQM}$-complete if none of the expansion rules for $\mathcal{SHIQM}$ is applicable.

The algorithm says that the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ is consistent iff the expansion rules can be applied in such a way they yield a $\mathcal{SHIQM}$-complete $(\mathcal{T}, \mathcal{F})$ without contradictions nor cycles. Otherwise the algorithm says that it is inconsistent. As in $\mathcal{SHIQ}$, due to the non-determinism of the algorithm, implementations of it have to guess the choices and possibly have to backtrack to choice points if a choice already made has led to a contradiction. The algorithm stops when we reach *some* $\mathcal{SHIQM}$-complete $(\mathcal{T}, \mathcal{F})$ that has neither contradictions nor cycles or when all the choices have yield $(\mathcal{T}, \mathcal{F})$ that has either contradictions or cycles.

7. Correctness of the tableau algorithm

In this section we prove termination and correctness of the tableau algorithm for $\mathcal{SHIQM}$ which was described in the previous section. Our proof of soundness (Theorem 3) is *modular*. We actually prove *preservation of soundness*: if the algorithm for $\mathcal{SHIQ}$ is sound, so does the algorithm for $\mathcal{SHIQM}$. This allows us to reuse the results on soundness for $\mathcal{SHIQ}$ [9] and has the advantage of making our proofs shorter and more conspicuous.

7.1. Termination of the tableau algorithm

We prove termination of the tableau algorithm for $\mathcal{SHIQM}$. Before doing this, we add a couple of definitions which will be used in that proof.

Definition 26 (*Concepts from Meta-Modelling*)**.** Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be in negation normal form.

We define the *set of concepts generated by the application of meta-modelling rules for* $\mathcal{O}$ as

$$\begin{aligned}\mathsf{concepts}(\mathcal{M}) = \{&A \sqcap \neg B \sqcup B \sqcap \neg A, A \sqcup \neg B,\\ &B \sqcup \neg A \mid a =_{\mathsf{m}} A, b =_{\mathsf{m}} B \in \mathcal{M}\}.\end{aligned}$$

Definition 27 (*Closure of a $\mathcal{SHIQM}$ Ontology*)**.** Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be in negation normal form.

We define the *closure of the ontology* $\mathcal{O}$ as

$$\mathsf{clos}(\mathcal{O}) = \bigcup_{C(a)\in\mathcal{A} \text{ or } C\in\mathcal{T}\cup\mathsf{concepts}(\mathcal{M})} \mathsf{clos}(C).$$

Theorem 1 (*Termination*)**.** *The tableau algorithm for $\mathcal{SHIQM}$ described in the previous section always terminates.*

Proof. Let $m = \#\mathcal{M}$, $k = \#\mathsf{clos}(\mathcal{O})$, $r = \#\mathbf{R}_{\mathcal{O}}$ and $g := max\{n \mid \geqslant nR.C \in \mathsf{clos}(\mathcal{O})\}$.

The algorithm constructs a graph that consists of an arbitrary set of interconnected root nodes and "trees" of blockable nodes rooted in some root node. Termination is a consequence of the following properties of the expansion rules:

1. New root nodes can be generated by the $\not\approx$-rule but they are bounded by $\#\{(a, b) \mid a, b \in \mathsf{dom}(\mathcal{M})\} = m^2$.
2. The expansion rules never remove nodes from the forest. The only rules that remove elements associated to nodes or edges in the forest are either the $\leqslant$, $\leqslant$-root or *close*-rule which sets them to $\emptyset$. If an edge label is set to $\emptyset$ by one of these rules, the node below this edge is blocked and will remain blocked forever. The $\leqslant$-root rule or close-rule only set the label of a root node x to $\emptyset$, and after this, the label x is never changed again since all edges to/from x are removed. Hence, this removal may only happen a finite number of times.
3. Nodes are labelled with subsets of $\mathsf{clos}(\mathcal{O})$ and edges with subsets of $\mathbf{R}_{\mathcal{O}}$, so there are at most $2^k.2^r.2^k = 2^{2k+r}$ different possible labellings for a pair of nodes and an edge. Therefore, if a path p is of length at least 2^{2k+r}, the pair-wise blocking condition implies the existence of two nodes x, y on p such that x directly blocks y. Since a path on which nodes are blocked cannot become longer, paths are of length at most 2^{2k+r}.
4. The concepts of the form $\exists R.C$ or $\geqslant nR.C$ in $\mathsf{clos}(\mathcal{O})$ trigger the generation of at most g successors y_i. The rule application which led to the generation of y_i will not be repeated. Since $\mathsf{clos}(\mathcal{O})$ contains a total of at most k concepts of the form $\exists R.C$ or $\geqslant nR.C$, the out-degree of the forest is bounded by gkr.

7.2. Tableau for $\mathcal{SHIQM}$

We now extend the definition of tableau given in Definition 6 with new properties that take into account the meta-modelling.

Definition 28 (*Tableau for $\mathcal{SHIQM}$*)**.** Let $\mathcal{O}$ be a $\mathcal{SHIQM}$-knowledge base of the form $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. We say that $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is a tableau for a $\mathcal{SHIQM}$ ontology $\mathcal{O}$ if

1. $\mathbb{T}$ is a tableau structure for $\mathbf{I}_{\mathcal{O}}$ and $\mathbf{R}_{\mathcal{O}}$ where $\mathbf{I}_{\mathcal{O}}$ is now the set of individuals occurring in $\mathcal{A}$ and $\mathcal{M}$ (not only in $\mathcal{A}$), i.e., $\mathcal{J} : \mathbf{I}_{\mathcal{O}} \to \mathbf{S}$ maps individuals occurring in $\mathcal{A}$ and $\mathcal{M}$ to elements in $\mathbf{S}$.
2. $\mathbf{S} \subseteq S_n$ for some S_n (see Definition 13),
3. Let $\mathbf{I}_{\mathcal{A}}$ be the set of individuals occurring in $\mathcal{A}$. Then, $\mathbb{T}' = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_{\mathcal{A}}})$ is a tableau for the $\mathcal{SHIQ}$-ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$, i.e., it satisfies the properties (P1)–(P16) from Definition 6.
4. Besides the properties (P1)–(P16), $\mathbb{T}$ also satisfies these extra properties:
   (P17) if $a =_{\mathsf{m}} A \in \mathcal{M}$, then $\mathcal{J}(a) = \{x \in \mathbf{S} \mid A \in \mathcal{L}(x)\}$.
   (P18) if $\mathcal{J}(a) = \mathcal{J}(b)$, $a =_{\mathsf{m}} A \in \mathcal{M}$ and $b =_{\mathsf{m}} B \in \mathcal{M}$, then $A \sqcup \neg B \in \mathcal{L}(s)$ and $B \sqcup \neg A \in \mathcal{L}(s)$ for all $s \in \mathbf{S}$.
   (P19) if $\mathcal{J}(a) \neq \mathcal{J}(b)$, $a =_{\mathsf{m}} A \in \mathcal{M}$ and $b =_{\mathsf{m}} B \in \mathcal{M}$, then there is some $t \in \mathbf{S}$ such that $A \sqcap \neg B \sqcup B \sqcap \neg A \in \mathcal{L}(t)$.

The following lemma says that "consistency" is equivalent to "having an abstract model".

Lemma 14. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. A $\mathcal{SHIQM}$-ontology $\mathcal{O}$ is consistent iff there exists a $\mathcal{SHIQM}$-tableau for $\mathcal{O}$.*

Proof. Direction $\Leftarrow$. Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ be a tableau for a $\mathcal{SHIQM}$ ontology $\mathcal{O}$. Then, we consider the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}} := \mathbf{S}$ and

$$A^{\mathcal{I}} := \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\}$$

$$a^{\mathcal{I}} := \mathcal{J}(a)$$

$$R^{\mathcal{I}} := \begin{cases} \mathcal{E}(R)^+ & \text{if } Trans(R) \\ \mathcal{E}(R) \cup \bigcup_{P \sqsubseteq^* R, P \neq R} P^{\mathcal{I}} & \text{otherwise} \end{cases}$$

where $\mathcal{E}(R)^+$ is the transitive closure of $\mathcal{E}(R)$.

[1] Consistency is the egg and semantic consequence is the chicken.

We prove that $\mathcal{I}$ is a model of the $\mathcal{SHIQM}$-ontology, i.e., it satisfies the three conditions of Definition 19. The first condition is trivial because it is exactly the same condition that appears in the definition of tableau (see Definition 28). The second condition follows from Lemma 2 where it is shown that $\mathcal{I}$ is a model of the $\mathcal{SHIQ}$-ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ without $\mathcal{M}$. The third condition follows from (P17) and the definition of $\mathcal{I}$, i.e., for all $a =_{\mathsf{m}} A$ in $\mathcal{T}$:

$$a^{\mathcal{I}} = \mathcal{J}(a) = \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\} = A^{\mathcal{I}}.$$

Direction $\Rightarrow$. Given $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a model of $\mathcal{O}$, we define a tableau $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ for $\mathcal{O}$ as follows.

$$\begin{aligned}
\mathbf{S} &:= \Delta^{\mathcal{I}} \\
\mathcal{L}(s) &:= \{C \in \mathsf{clos}(\mathcal{O}) \mid s \in C^{\mathcal{I}}\} \\
\mathcal{E}(R) &:= R^{\mathcal{I}} \\
\mathcal{J}(a) &:= a^{\mathcal{I}}
\end{aligned}$$

where clos is given in Definition 27. We now show that this is a tableau for $\mathcal{O}$.

To prove (P17), assume $a =_{\mathsf{m}} A$. It follows from the third condition of Definition 19 that $A^{\mathcal{I}} = a^{\mathcal{I}}$. We also have that

$$a^{\mathcal{I}} = \mathcal{J}(a) = \{s \in \Delta^{\mathcal{I}} \mid s \in A^{\mathcal{I}}\} = \{s \in \mathbf{S} \mid A \in \mathcal{L}(s)\}.$$

To prove (P18), assume that $\mathcal{J}(a) = \mathcal{J}(b)$, $a =_{\mathsf{m}} A \in \mathcal{M}$ and $b =_{\mathsf{m}} B \in \mathcal{M}$. Since $\mathcal{I}$ is a model of the ontology, $A^{\mathcal{I}} = a^{\mathcal{I}} = b^{\mathcal{I}} = B^{\mathcal{I}}$. This means that $(A \sqcup \neg B)^{\mathcal{I}} = (\neg A \sqcup B)^{\mathcal{I}} = \Delta^{\mathcal{I}} = \mathbf{S}$. Hence, $A \sqcup \neg B \in \mathcal{L}(s)$ and $\neg A \sqcup B \in \mathcal{L}(s)$ for all $s \in \mathbf{S}$.

To prove (P19), assume that $\mathcal{J}(a) \neq \mathcal{J}(b)$, $a =_{\mathsf{m}} A \in \mathcal{M}$ and $b =_{\mathsf{m}} B \in \mathcal{M}$. Since $\mathcal{I}$ is a model of the ontology, $a^{\mathcal{I}} = A^{\mathcal{I}} \neq b^{\mathcal{I}} = B^{\mathcal{I}}$. Then, there exists some $s \in (A \sqcap \neg B \sqcup B \sqcap \neg A)^{\mathcal{I}}$. Hence, $A \sqcap \neg B \sqcup B \sqcap \neg A \in \mathcal{L}(s)$.

The rest of the properties, (P1) to (P16) are easy to show using the fact that $\mathcal{I}$ is a model of $(\mathcal{T}, \mathcal{R}, \mathcal{A})$.

The proof of the above lemma invokes the corresponding result for $\mathcal{SHIQ}$ (Lemma 2) but only in one direction. The direction $\Rightarrow$ cannot invoke that lemma because clos for $\mathcal{SHIQ}$ does not include the concepts from the Mbox (see Definition 27).

7.3. Abstract canonical model for $\mathcal{SHIQM}$

The "abstract canonical interpretation" of a $\mathcal{SHIQM}$-knowledge base is built as the composition of two interpretations: the abstract $\mathcal{SHIQ}$-canonical interpretation (Definition 8) and the function set that computes the set associated to an individual with meta-modelling recursively.

$$\begin{tikzcd}
\mathcal{SHIQM} \arrow[d, "\mathcal{J}\ \textit{Abstract Canonical Model for } \mathcal{SHIQ}"] \arrow[dd, bend right, dotted, "\mathcal{J}'"'] & \\
\mathbf{S} = \mathsf{Paths}(\mathcal{F}) \arrow[d, "\cong"', "\mathsf{set}\ \ \textit{Recursive Computation of Sets}"] & \textit{Domain for } \mathcal{SHIQ} \\
\mathbf{S}' = \mathsf{set}(\mathsf{Paths}(\mathcal{F})) & \textit{Domain for } \mathcal{SHIQM}
\end{tikzcd}$$

The domain $\mathbf{S}$ of the tableau built from the completion forest $\mathcal{F}$ of a $\mathcal{SHIQ}$-ontology is the set of paths in $\mathcal{F}$ while the domain $\mathbf{S}'$ of the tableau of a $\mathcal{SHIQM}$-ontology consists of paths, sets of paths, sets of sets of paths, etc. The idea of the function set is to associate the set of objects that the individual c with meta-modelling represents. We define now set formally as follows.

Definition 29 (*From Basic Paths to Sets*). Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ and let $\mathcal{F}$ be a complete completion forest without contradictions nor cycles w.r.t. $\mathcal{M}$. For $p \in \mathsf{Paths}(\mathcal{F})$ we define $\mathsf{set}(p)$ as follows.

$$\begin{aligned}
\mathsf{set}(p) \;&=\; \{\mathsf{set}(q) \mid A \in \mathcal{F}(\mathsf{Tail}(q))\} \\
&\qquad \text{if } p = \left[\frac{c}{c}\right] \text{ for some } c \approx a =_{\mathsf{m}} A \in \mathcal{M} \\
\mathsf{set}(p) &= p \quad \text{otherwise.}
\end{aligned}$$

The function set only changes the paths of the form $p = [\frac{c}{c}]$ since only root nodes can have meta-modelling. Paths that are not of the form $p = [\frac{c}{c}]$ are left unchanged. This means that the function set acts as the identity if it is restricted to the paths that are not equal to $p = [\frac{c}{c}]$. More formally,

$$\mathsf{set} \upharpoonright_{\{p \in \mathsf{Paths}(\mathcal{F}) \mid p \neq [\frac{c}{c}]\}} = id$$

where id is the identity function.

The following example illustrates the idea behind the function set. We also write c instead of $p = [\frac{c}{c}]$ since a path of the form $p = [\frac{c}{c}]$ can be identified with c (see also Definition 8 and the proof of Lemma 2 where c is interpreted as $p = [\frac{c}{c}]$).

Example 11. We consider the ontology network of Fig. 2. Here we have for example that *river* is an individual with meta-modelling. As such, its interpretation should be a set and not a basic object. The set associated to *river* is given by the function set and it is as follows.

$$\mathsf{set}(river) = \{queguay, santaLucia\}.$$

The individual *hydrographic* has also meta-modelling. But its inhabitants also have meta-modelling. The set associated to *hydrographic* is a set of sets given as follows.

$$\begin{aligned}
\mathsf{set}(hydrographic) = \{&\{queguay, santaLucia\}, \\
&\{deRocha, delSauce\}\}.
\end{aligned}$$

On the other hand, *queguay* does not have meta-modelling and we define set as follows.

$$\mathsf{set}(queguay) = queguay.$$

The function set is actually defined recursively. We will prove later the correctness of this recursive definition in Corollary 2. We will also prove that set is an injective function (surjectivity is obvious). The fact that this function is a bijection is pictured in the diagram by means of the symbol $\cong$. As we mentioned before, we use the function set to build a canonical interpretation of the $\mathcal{SHIQM}$-ontology from the canonical interpretation of $\mathcal{SHIQ}$. The fact that set is an isomorphism plays an important role in the proof that the canonical interpretation of $\mathcal{SHIQM}$ is a model since it allows us to use Lemma 3 in the proof of Soundness (Theorem 3).

In order to understand the idea of how we use the function set to build a canonical model of the $\mathcal{SHIQM}$-ontology from the canonical model of the corresponding ontology without meta-modelling, we give an example of an ontology in $\mathcal{ALCQM}$ where all the paths are of the form $p = [\frac{c}{c}]$. It is not necessary to consider a more complicated example, since paths that are not of the form $p = [\frac{c}{c}]$ do not really help understanding the idea of set since they are left unchanged. As before we write c instead of "$p = [\frac{c}{c}]$" since "$p = [\frac{c}{c}]$" can be identified with c. The following example also illustrates the idea behind the justification of the recursive definition for set.

Example 12. Suppose we have an ontology $(\mathcal{T}, \mathcal{A}, \mathcal{M})$ with four individuals a, b, c and d with axioms $B(a)$, $A(c)$, $A(d)$ and the meta-modelling axioms given by $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$. The canonical interpretation $\mathcal{I}$ of the $\mathcal{ALCQ}$ ontology is then,

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{a, b, c, d\} \\
A^{\mathcal{I}} &= \{c, d\} \\
B^{\mathcal{I}} &= \{a\}
\end{aligned}$$

coming from a complete forest without contradictions where $\mathcal{F}(a) = \{B\}$, $\mathcal{F}(c) = \{A\}$ and $\mathcal{F}(d) = \{A\}$. Intuitively, we see that we need to force the following equations to make the metamodelling axioms $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$ satisfiable:

$$a = \{c, d\}$$
$$b = \{a\}.$$

These equations do not have cycles because $\mathcal{F}$ does not have cycles w.r.t. $\mathcal{M}$. We can then define a function set as follows:

$$\mathsf{set}(a) = \{c, d\}$$
$$\mathsf{set}(b) = \{\mathsf{set}(a)\}.$$

The canonical interpretation $\mathcal{I}_{\mathsf{m}}$ for the ontology in $\mathcal{ALCQM}$ is now defined as follows.

$$\begin{aligned}\Delta^{\mathcal{I}_{\mathsf{m}}} &= \mathsf{set}(\Delta^{\mathcal{I}}) \\ &= \{\{c, d\}, \{\{c, d\}\}, c, d\} \\ (A)^{\mathcal{I}_{\mathsf{m}}} &= \{c, d\} \\ (B)^{\mathcal{I}_{\mathsf{m}}} &= \{\{c, d\}\}.\end{aligned}$$

In this case, $\mathcal{I}_{\mathsf{m}}$ is a model of $(\mathcal{T}, \mathcal{A}, \mathcal{M})$. By defining $S_0 = \{c, d\}$, we see that $\Delta^{\mathcal{I}_{\mathsf{m}}} \subset S_2$.

We now define two relations: (1) $\prec$ on the nodes of the forest and (2) $\ll$ on the set of paths. We will prove that both relations are well-founded if $\mathcal{F}$ has no cycles w.r.t. $\mathcal{M}$. The fact that $\ll$ is well-founded allows us to use the recursion and the induction principles given in Definitions 14 and 16. The recursion principle will be used to justify the recursive definition of set (Definition 29). The induction principle will be used to prove injectivity of set (Lemma 18). Injectivity is needed to prove Theorem 3 since we are using set to build a canonical model of the $\mathcal{SHIQM}$-ontology from the canonical model of the corresponding $\mathcal{SHIQ}$-ontology without meta-modelling.

Definition 30. The relation $\prec$ on the set of nodes of $\mathcal{F}$ is defined as $b \prec c$ iff $A \in \mathcal{F}(b)$, $c \approx a$ and $a =_{\mathsf{m}} A \in \mathcal{M}$.

Example 13. In Example 12, we define $\prec$ on the set $\{a, b, c, d\}$ as follows:

$$c, d \prec a$$
$$a \prec b.$$

If a $\prec$-decreasing sequence uses the same axiom twice then that sequence has a cycle and $\prec$ cannot be well-founded. From this, it is easy to prove the following lemma.

Lemma 15. *If $\prec$ is well-founded then we have that $\mathsf{maxl}^{\prec}(c) \leq \sharp(\mathcal{M})$ for all nodes c in $\mathcal{F}$.*

Definition 31. The relation $\ll$ on $\mathsf{Paths}(\mathcal{F})$ is defined as $q \ll p$ iff $\mathsf{Tail}(q) \prec c$ and $p = [\frac{c}{c}]$.

Note that neither $\prec$ nor $\ll$ are transitive. We now prove that $\prec$ and $\ll$ are well-founded if $\mathcal{F}$ has no cycles.

Theorem 2. *Suppose $\mathcal{F}$ has a finite set of nodes. If the completion forest $\mathcal{F}$ has no cycles w.r.t. $\mathcal{M}$ then $\prec$ is well-founded.*

Proof. Suppose $\prec$ is not well-founded. Since the set of nodes of $\mathcal{F}$ is finite, by Lemma 7, $\prec$ has a cycle,

$$y_n \prec y_1 \prec \cdots \prec y_n.$$

It is easy to see that this contradicts the fact that $\mathcal{F}$ has no cycles.

Since the relation $\ll$ is defined on the set of paths which can be infinite, we cannot apply Lemma 7 in the following corollary. Instead, we apply Lemma 6.

Corollary 1. *Suppose $\mathcal{F}$ has a finite set of nodes. If the completion forest $\mathcal{F}$ has no cycles w.r.t. $\mathcal{M}$ then $\ll$ is well-founded.*

Proof. Suppose towards a contradiction that $\ll$ is not well-founded. It follows from Lemma 6 that there exists an infinite $\ll$-decreasing sequence starting from some path p_0.

$$\ldots \ll \underset{\left[\frac{c_2}{c_2}\right]}{\underset{\parallel}{p_2}} \ll \underset{\left[\frac{c_1}{c_1}\right]}{\underset{\parallel}{p_1}} \ll \underset{\left[\frac{c_0}{c_0}\right]}{\underset{\parallel}{p_0}}.$$

It follows from the definition of $\ll$ that for all $i \in \mathbb{N}$, we have that $p_i = [\frac{c_i}{c_i}]$ for some node c_i in the forest. By definition of $\ll$, we have that

$$\ldots \prec c_2 \prec c_1 \prec c_0.$$

This contradicts the fact that $\prec$ is well-founded.

Since $\ll$ is well-founded, we can now apply the recursion principle and define the function set recursively. Note that in the recursive step of that definition, we have that $q \ll p$. Hence, we have the following result:

Corollary 2 (*Correctness of the Recursion for* set). *The function* set *is a correct recursive definition.*

Lemma 16. *Let $\mathcal{F}$ be a completion forest without cycles with a finite number of nodes. If $S_0 = \mathsf{Paths}(\mathcal{F})$, for all $p \in \mathsf{Paths}(\mathcal{F})$, we have that*

$$\mathsf{set}(p) \in S_{\sharp(\mathcal{M})}.$$

Proof. From the proof of Corollary 1, we have that

$$\mathsf{maxl}^{\ll}(p) = \begin{cases} \mathsf{maxl}^{\prec}(c) & \text{if } p = \left[\frac{c}{c}\right] \text{ for some} \\ & c \approx a =_{\mathsf{m}} A \in \mathcal{M} \\ 0 & \text{otherwise.}\end{cases} \tag{4}$$

It follows from Lemma 15 and (4) that

$$\mathsf{maxl}^{\ll}(p) \leq \sharp(\mathcal{M}).$$

It is easy to prove that $\mathsf{set}(p) \in S_{\mathsf{maxl}^{\ll}(p)}$ by induction on $\mathsf{maxl}^{\ll}(p)$. Then, $\mathsf{set}(p) \in S_{\mathsf{maxl}^{\ll}(p)} \subseteq S_{\sharp(\mathcal{M})}$.

Lemma 17. *Let $\mathcal{F}$ be a $\mathcal{SHIQM}$-complete completion forest without contradictions and let $A^{\mathcal{I}} = \{p \in \mathsf{Paths}(\mathcal{F}) \mid A \in \mathcal{F}(\mathsf{Tail}(p))\}$ where $\mathcal{I}$ is the interpretation of Lemma 14 associated to the specific tableau $\mathbb{T}$ of Definition 8. If $a =_{\mathsf{m}} A$ and $a' =_{\mathsf{m}} A'$ then either $a \approx a'$ or $a \not\approx a'$. In the first case, $A^{\mathcal{I}} = A'^{\mathcal{I}}$ and in the second case, $A^{\mathcal{I}} \neq A'^{\mathcal{I}}$.*

Proof. Since $\mathcal{F}$ is complete, the close-rule cannot be applied any more. Hence we have that either $a \approx a'$ or $a \not\approx a'$.

Suppose that $a \approx a'$. By the $\approx$-rule, $A \sqcup \neg A'$ and $A' \sqcup \neg A$ should both belong to $\mathcal{T}$. By the Tbox-rule, $\{A \sqcup \neg A', A' \sqcup \neg A\} \subseteq \mathcal{F}(x)$ for all individuals x that are not indirectly blocked. All paths p end in a node x that is not indirectly blocked. Hence, we have that $\{A \sqcup \neg A', A' \sqcup \neg A\} \subseteq \mathcal{F}(\mathsf{Tail}(p))$ for all $p \in \mathsf{Paths}(\mathcal{F})$. It is easy to see that $A \in \mathcal{F}(\mathsf{Tail}(p))$ iff $A' \in \mathcal{F}(\mathsf{Tail}(p))$ because $\mathcal{F}$ is complete and has no contradictions. Hence, $A^{\mathcal{I}} = A'^{\mathcal{I}}$.

Suppose now that $a \not\approx a'$. By the $\not\approx$-rule, there exists a node z such that $A \sqcap \neg A' \sqcup A' \sqcap \neg A$ belongs to $\mathcal{F}(z)$. Note that z can never be blocked since z is a root node. Hence, there exists a path p where $\mathsf{Tail}(p) = z$. Since $\mathcal{F}$ is complete and has no contradictions, $\mathcal{F}(z)$ will contain either A and $\neg A'$ or A' and $\neg A$. In the first case, it cannot contain A' and in the second case, it cannot contain A. Hence, $A^{\mathcal{I}} \neq A'^{\mathcal{I}}$.

The following lemma shows that set is an *injective function*. The proof of injectivity is interesting because we apply the induction principle using the fact that $\ll$ is well-founded. This lemma is necessary to build the canonical model of the $\mathcal{SHIQM}$ ontology from the canonical model of the corresponding ontology without meta-modelling in Theorem 3.

The fact that set is a *function* is not so evident and it is essentially a consequence of the $\approx$-rule as illustrated by the following example (see the proof of Lemma 17).

Example 14. Consider the case of an ontology where $A(c)$, $A'(d)$, $a =_{\mathsf{m}} A$ and $a =_{\mathsf{m}} A'$. The $\mathcal{SHIQ}$-canonical interpretation of the ontology without meta-modelling given by $A^{\mathcal{I}} = \{c\}$ and $(A')^{\mathcal{I}} = \{d\}$ is not a model of the $\mathcal{SHIQM}$-ontology since

$$a^{\mathcal{I}} = A^{\mathcal{I}} = \{c\}$$
$$a^{\mathcal{I}} = (A')^{\mathcal{I}} = \{d\}.$$

The $\approx$-rule has to be applied to ensure that the set associated to a is uniquely determined and equal to $\{c, d\}$.

The fact that set is *injective* is a consequence of the $\not\approx$-rule as illustrated by the following example (this can also be seen in the proof of Lemma 17).

Example 15. Consider the case of an ontology where $a \neq b$, $A(c)$, $B(c)$, $a =_{\mathsf{m}} A$ and $b =_{\mathsf{m}} B$. The $\mathcal{SHIQ}$-canonical interpretation of the ontology without meta-modelling given by $A^{\mathcal{I}} = \{c\}$ and $B^{\mathcal{I}} = \{c\}$ is not a model of the $\mathcal{SHIQM}$-ontology since

$$a^{\mathcal{I}} = A^{\mathcal{I}} = \{c\} = B^{\mathcal{I}} = b^{\mathcal{I}}.$$

The $\not\approx$-rule has to be applied to ensure that the sets associated to a and b are different.

Lemma 18 (*Injective Function*)**.** *Let $\mathcal{F}$ be a $\mathcal{SHIQM}$-complete completion forest that has neither contradictions nor cycles. Then,* set *is an injective function, i.e., $p = p'$ if and only if* $\mathsf{set}(p) = \mathsf{set}(p')$.

Proof. We prove first that set is a function. It is enough to consider the case when $p = [\frac{c}{c}]$, $c \approx a =_{\mathsf{m}} A$ and $c \approx a' =_{\mathsf{m}} A'$. By Lemma 17, $a \approx a'$ and $\{x \mid A \in \mathcal{F}(x)\} = \{x \mid A' \in \mathcal{F}(x)\}$. Hence, $\mathsf{set}(p)$ is uniquely determined.

To prove that set is injective, we do induction on $\ll$ which we know that is well-founded by Corollary 1. By definition of set, we have two cases. The first case is when $\mathsf{set}(p) = p$. We have that $\mathsf{set}(p') = p$ and p' is exactly p. This was the base case. In the second case, we have that for $p = [\frac{c}{c}]$, $c \approx a$ and $a =_{\mathsf{m}} A$,

$$\mathsf{set}(p) = \{\mathsf{set}(q) \mid A \in \mathcal{F}(\mathsf{Tail}(q))\}.$$

Since $\mathsf{set}(p) = \mathsf{set}(p')$, we also have that $p' = [\frac{c'}{c'}]$, $c' \approx a'$ and $a' =_{\mathsf{m}} A'$ such that

$$\mathsf{set}(p') = \{\mathsf{set}(q') \mid A' \in \mathcal{F}(\mathsf{Tail}(q'))\}.$$

Again since $\mathsf{set}(p) = \mathsf{set}(p')$, for all $A \in \mathcal{F}(\mathsf{Tail}(q))$ there exists q' such that $A' \in \mathcal{F}(\mathsf{Tail}(q'))$ and $\mathsf{set}(q) = \mathsf{set}(q')$. By Induction Hypothesis, $q = q'$. Hence, $A^{\mathcal{I}} = \{q \mid A \in \mathcal{F}(\mathsf{Tail}(q))\} \subseteq \{q' \mid A' \in \mathcal{F}(\mathsf{Tail}(q'))\} = A'^{\mathcal{I}}$. Similarly, we get $A'^{\mathcal{I}} \subseteq A^{\mathcal{I}}$. Then, $A^{\mathcal{I}} = A'^{\mathcal{I}}$. It follows from Lemma 17 that $a \approx a'$. Then $c \approx c'$. Since the paths consists only of representatives of equivalence classes, we have that $c = c'$ and hence, $p = p'$.

We now define the notion of "abstract canonical model" which is built from a complete forest that has neither contradictions nor cycles.

Definition 32 ($\mathcal{SHIQM}$ *Canonical Structure*)**.** Let $\mathcal{F}$ be a completion forest. We define *the canonical tableau structure* $\mathbb{T}' = (\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}')$ for $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ as follows:

$$\begin{aligned}
\mathbf{S}' &= \{\mathsf{set}(p) \mid p \in \mathbf{S}\}\\
\mathcal{L}'(s) &= \mathcal{L}(p) \text{ with } s = \mathsf{set}(p)\\
\mathcal{E}'(R) &= \{(\mathsf{set}(p), \mathsf{set}(q)) \in \mathbf{S} \times \mathbf{S} \mid (p, q) \in \mathcal{E}(R)\}\\
\mathcal{J}'(a) &= \mathsf{set}(\mathcal{J}(a))
\end{aligned}$$

where $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ is the canonical tableau structure (i.e., the "$\mathcal{SHIQ}$-abstract canonical interpretation") built from the completion forest $\mathcal{F}$ given in Definition 8.

Since the tableau structure $\mathbb{T}$ from Definition 8 is built from the completion forest $\mathcal{F}$, the domains of $\mathcal{J}$ and $\mathcal{J}'$ is the set of individuals in the ontology $(\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ which includes the individuals occurring in the MBox.

Theorem 3 ($\mathcal{SHIQM}$ *Abstract Canonical Model*)**.** *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$. If the expansion rules for $\mathcal{SHIQM}$ can be applied to $\mathcal{O}$ in such a way that they yield a complete completion forest $\mathcal{F}$ that has no contradictions and has no cycles w.r.t. $\mathcal{M}$ then the tableau structure given in Definition 32 is a tableau for the $\mathcal{SHIQM}$-ontology $\mathcal{O}$.*

Proof. We have to prove the four conditions in the definition of tableau for $\mathcal{SHIQM}$ (Definition 28).

The first condition follows from Definition 32.

The second condition follows from Lemma 16, i.e.,

$$\mathbf{S}' = \mathsf{set}(\mathsf{Paths}(\mathcal{F})) \subseteq S_{\sharp(\mathcal{M})}.$$

We now prove the third condition. By Lemma 18, set is a bijection from $\mathbf{S}$ to $\mathbf{S}'$. Hence, $(\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_{\mathcal{A}}})$ and $(\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}'\restriction_{\mathbf{I}_{\mathcal{A}}})$ are isomorphic tableau structures. Since $\mathcal{F}$ is $\mathcal{SHIQM}$-complete, it is also $\mathcal{SHIQ}$-complete. It follows from Lemma 4 that $(\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\restriction_{\mathbf{I}_{\mathcal{A}}})$ satisfies (P1)–(P12). By Lemma 3, $(\mathbf{S}', \mathcal{L}', \mathcal{E}', \mathcal{J}'\restriction_{\mathbf{I}_{\mathcal{A}}})$ satisfies (P1)–(P12) as well. We now prove that it also satisfies (P13)–(P16). This is because the canonical tableau structure built from the initial completion forest $\mathcal{F}_0$ satisfies (P13)–(P16) and all expansion rules preserve these properties.

In order to prove the fourth condition, we need to prove the rules (P17)–(P19).

Proof of (P17). Let $a =_{\mathsf{m}} A \in \mathcal{M}$. Then,

$$\begin{aligned}
\mathcal{J}'(a) &= \mathsf{set}(\mathcal{J}(a))\\
&= \mathsf{set}\left(\left[\frac{b}{b}\right]\right) \text{ for } b \approx a\\
&= \{\mathsf{set}(q) \mid A \in \mathcal{F}(\mathsf{Tail}(q))\}\\
&= \{\mathsf{set}(q) \mid A \in \mathcal{L}(q)\}\\
&= \{\mathsf{set}(q) \mid A \in \mathcal{L}'(\mathsf{set}(q))\}.
\end{aligned}$$

Proof of (P18). Suppose $\mathcal{J}'(a) = \mathcal{J}'(b)$, $a =_{\mathsf{m}} A \in \mathcal{M}$ and $b =_{\mathsf{m}} B \in \mathcal{M}$. Then, $\mathsf{set}(\mathcal{J}(a)) = \mathsf{set}(\mathcal{J}(b))$ by definition of $\mathcal{J}'$. By Lemma 18, set is injective and hence, $\mathcal{J}(a) = \mathcal{J}(b)$. By Definition 8, we have that $\mathcal{J}(a) = \mathcal{J}(b) = [\frac{c}{c}]$ where $c \approx a \approx b$. Since $\mathcal{F}$ is complete, the $\approx$-rule cannot be applied and we have that $A \sqcup \neg B$ and $B \sqcup \neg A$ should have been added to the Tbox. Since the Tbox-rule cannot be applied either, $A \sqcup \neg B \in \mathcal{F}(x)$ and $B \sqcup \neg A \in \mathcal{F}(x)$ for all nodes x that are not blocked. By construction, $\mathsf{Tail}(p)$ is not blocked for all paths p. Hence, for all nodes $x = \mathsf{Tail}(p)$, $A \sqcup \neg B \in \mathcal{F}(\mathsf{Tail}(p))$ and $B \sqcup \neg A \in \mathcal{F}(\mathsf{Tail}(p))$. This completes the proof of (P18) since $\mathcal{F}(\mathsf{Tail}(p)) = \mathcal{L}(p) = \mathcal{L}'(\mathsf{set}(p))$ for all $p \in \mathsf{Paths}(\mathcal{F})$.

Proof of (P19). Suppose $\mathcal{I}'(a) \neq \mathcal{I}'(b)$, $a =_{\mathrm{m}} A \in \mathcal{M}$ and $b =_{\mathrm{m}} B \in \mathcal{M}$. Then, $\mathsf{set}(\mathcal{I}(a)) \neq \mathsf{set}(\mathcal{I}(b))$ by definition of $\mathcal{I}'$. By Lemma 18, set is a function and hence, $\mathcal{I}(a) \neq \mathcal{I}(b)$. Then, $a \not\approx b$. Since $\mathcal{F}$ is complete, the $\not\approx$-rule cannot be applied and there will be a root node z such that $A \sqcap \neg B \sqcup B \sqcap \neg A \in \mathcal{F}(z)$. Since z is considered to be a root node, take $p = \left[\frac{z}{z}\right]$. Then, $A \sqcap \neg B \sqcup B \sqcap \neg A \in \mathcal{F}(\mathsf{Tail}(p)) = \mathcal{L}(p) = \mathcal{L}'(\mathsf{set}(p))$.

By using the notion of isomorphism, we have related the canonical interpretations of $\mathcal{SHIQ}$ and $\mathcal{SHIQM}$. We also avoided repeating the proof of (P1)–(P12). The properties (P1)–(P11) were already proved by Horrocks et al. [9] and we proved (P12) in Lemma 4.

7.4. Completeness of the tableau algorithm

We now prove the converse of Theorem 3 to conclude our final result on correctness.

Definition 33. Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{I})$ be a $\mathcal{SHIQM}$-tableau for a $\mathcal{SHIQM}$-ontology $\mathcal{O}$ and $\mathcal{F}$ a completion forest. We define a structure preserving map $\pi : \mathcal{F} \to \mathbb{T}$ as a function π from the set of nodes of $\mathcal{F}$ to $\mathbf{S}$ that satisfies the following conditions:

1. $\mathcal{F}(x) \subseteq \mathcal{L}(\pi(x))$.
2. If y is an S-neighbour of x, then $(\pi(x), \pi(y)) \in \mathcal{E}(S)$.
3. $x \not\approx y$ implies $\pi(x) \neq \pi(y)$.
4. $x \approx y$ implies $\pi(x) = \pi(y)$.

for all nodes x, y in $\mathcal{F}$.

Since the tableau algorithm changes the forest as well as the Tbox, we consider pairs $(\mathcal{T}, \mathcal{F})$ composed of a Tbox $\mathcal{T}$ and a forest $\mathcal{F}$.

Notation 1. *$(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow (\mathcal{T}_2, \mathcal{F}_2)$ means that $(\mathcal{T}_2, \mathcal{F}_2)$ is obtained from $(\mathcal{T}_1, \mathcal{F}_1)$ by the application of any of the $\mathcal{SHIQM}$-expansion rules. In order to express which rule has been applied, the arrow is equipped with the appropriate subscript, e.g., $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{close\approx} (\mathcal{T}_2, \mathcal{F}_2)$ is the application of the close-rule where the choice is an equality.*

Lemma 19. *Let $(\mathcal{T}_1, \mathcal{F}_1)$ be a Tbox and a completion forest generated by the tableau algorithm for $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ and let $\pi_1 : \mathcal{F}_1 \to \mathbb{T}$ be a structure preserving map such that $\pi_1(a) = \mathcal{I}(a)$ for all a in $\mathcal{O}$. If an expansion rule is applicable to $(\mathcal{T}_1, \mathcal{F}_1)$, then this rule can be applied such that it yields a completion forest $\mathcal{F}_2$ and a structure preserving map $\pi_2 : \mathcal{F}_2 \to \mathbb{T}$ extending π_1.*

$$\begin{tikzcd}
(\mathcal{T}_1, \mathcal{F}_1) \arrow[r, dotted] \arrow[d, "\pi_1"'] & (\mathcal{T}_2, \mathcal{F}_2) \arrow[dl, dotted, "\pi_2"] \\
\mathbb{T} &
\end{tikzcd}$$

Proof. We do the proof of the most interesting cases.

- Suppose the Tbox-rule is applicable to $(\mathcal{T}_1, \mathcal{F}_1)$. Then, $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{Tbox} (\mathcal{T}_2, \mathcal{F}_2)$ where $\mathcal{T}_1 = \mathcal{T}_2$ and $\mathcal{F}_2$ is exactly the same as $\mathcal{F}_1$ except in one node x_0 where $\mathcal{F}_2(x_0) = \mathcal{F}_1(x_0) \cup \{C\}$ and $C \in \mathcal{T}_1$.

  In this case the map π_2 is exactly the same as π_1. We have to prove that $\pi_2 : \mathcal{F}_2 \to \mathbb{T}$ is a structure preserving map. The second, third and fourth conditions of Definition 33 are trivial. The first condition is also trivial for all the nodes except for the node x_0 that has changed. We have to prove that $\mathcal{F}_2(x_0) = \mathcal{F}_1(x_0) \cup \{C\} \subseteq \mathcal{L}(\pi_2(x_0))$. Since $\mathcal{F}_1(x_0) \subseteq \mathcal{L}(\pi_1(x_0)) = \mathcal{L}(\pi_2(x_0))$ because π_1 is a structure preserving map, it is enough to prove that $C \in \mathcal{L}(\pi_1(x_0)) = \mathcal{L}(\pi_2(x_0))$. We have two cases:

  1. If $C \in \mathcal{T} \subseteq \mathcal{T}_1$ then $C \in \mathcal{L}(\pi_1(x_0)) = \mathcal{L}(\pi_2(x_0))$ by (P12).
  2. If $C \in \mathcal{T}_1 \setminus \mathcal{T}$ then C is either $A \sqcup \neg B$ or $\neg A \sqcup B$ for $a =_{\mathrm{m}} A$, $b =_{\mathrm{m}} B$ and $a \approx b$.

     $$\begin{aligned} \mathcal{I}(a) &= \pi_1(a) && \text{by hypothesis} \\ &= \pi_1(b) && \text{by Definition 33(4)} \\ &= \mathcal{I}(b) && \text{by hypothesis.} \end{aligned}$$

     Using (P18), we conclude that both $A \sqcup \neg B$ and $\neg A \sqcup B$ belong to $\mathcal{L}(\pi_1(x_0)) = \mathcal{L}(\pi_2(x_0))$.

- Suppose the $\not\approx$-rule is applicable to $(\mathcal{T}_1, \mathcal{F}_1)$ for $a \not\approx b$, $a =_{\mathrm{m}} A$ and $b =_{\mathrm{m}} B$. Then, $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{\not\approx} (\mathcal{T}_2, \mathcal{F}_2)$ where $\mathcal{T}_1 = \mathcal{T}_2$ and $\mathcal{F}_2$ is obtained by adding a node z to $\mathcal{F}_1$ with $\mathcal{F}_2(z) = \{A \sqcap \neg B \sqcup \neg A \sqcap \neg B\}$.

  $$\begin{aligned} \mathcal{I}(a) &= \pi_1(a) && \text{by hypothesis} \\ &\neq \pi_1(b) && \text{by Definition 33(3)} \\ &= \mathcal{I}(b) && \text{by hypothesis.} \end{aligned}$$

  By (P19), there exists $t \in \mathbf{S}$ with $A \sqcap \neg B \sqcup \neg A \sqcap \neg B \in \mathcal{L}(t)$. We define π_2 by extending the domain of π_1 with the new element z and setting $\pi_2(z) = t$.

  We have to prove that $\pi_2 : \mathcal{F}_2 \to \mathbb{T}$ is a structure preserving map. All the conditions are trivial except for the first one on the new node z which is proved as follows.

  $$\begin{aligned} \mathcal{F}_2(z) &= \{A \sqcap \neg B \sqcup \neg A \sqcap \neg B\} \\ &\subseteq \mathcal{L}(t) \\ &= \mathcal{L}(\pi_2(z)) && \text{by def. of } \pi_2. \end{aligned}$$

- Suppose the close rule is applicable to $(\mathcal{T}_1, \mathcal{F}_1)$ for $a =_{\mathrm{m}} A$ and $b =_{\mathrm{m}} B$. If $\mathcal{I}(a) = \mathcal{I}(b)$ then we add $a \approx b$ and $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{close\approx} (\mathcal{T}_2, \mathcal{F}_2)$. Otherwise, we add $a \not\approx b$ and $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{close\not\approx} (\mathcal{T}_2, \mathcal{F}_2)$. In both cases, the map π_2 is exactly the same as π_1. The map π_2 trivially satisfies the first and second conditions of Definition 33 since the forest has not changed except for the fact that an equality or inequality has been added (if $a \approx b$ then $\mathcal{F}$ is set to the empty set for a or b). It is enough to prove that π_2 satisfies the third and fourth conditions just for a and b since $\approx$ or $\not\approx$ has changed only on these elements. Suppose we added $a \approx b$. By our choice of expansion rule, we also have that $\mathcal{I}(a) = \mathcal{I}(b)$.

  $$\begin{aligned} \pi_2(a) &= \pi_1(a) && \text{since } \pi_2 = \pi_1 \\ &= \mathcal{I}(a) && \text{by hypothesis} \\ &= \mathcal{I}(b) && \text{by our choice of expansion rule} \\ &= \pi_1(b) && \text{by hypothesis} \\ &= \pi_2(b) && \text{since } \pi_2 = \pi_1. \end{aligned}$$

- Suppose the $\exists$-rule is applicable to $\mathcal{F}_1$. Then, $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{exists} (\mathcal{T}_2, \mathcal{F}_2)$, there exists x_0 such that $\exists R.C \in \mathcal{F}_1(x_0) = \mathcal{F}_2(x_0)$ and a successor y_0 of x_0 is generated for x_0 such that $\mathcal{F}_2(x_0, y_0) = \{R\}$ and $\mathcal{F}_2(y_0) = \{C\}$. Since π_1 is a structure preserving map, $\exists R.C \in \mathcal{F}_1(x_0) \subseteq \mathcal{L}(\pi_1(x_0))$. By (P5), we have that there exists $t \in \mathbf{S}$ such that $C \in \mathcal{L}(t)$ and $(s, t) \in \mathcal{E}(R)$ for $s = \pi_1(x_0)$. We define π_2 from π_1 by extending the domain of π_1 with the element y_0 and setting $\pi_2(y_0) = t$. To prove that π_2 is a structure preserving map, it is enough to consider these two cases (the rest are trivial):
  1. We prove the first condition for the new node y_0. By the above,
     $\mathcal{F}_2(y_0) = \{C\} \subseteq \mathcal{L}(t) = \mathcal{L}(\pi_2(y_0))$.
  2. We prove the second condition for the new pair (x_0, y_0). By the above,
     $(\pi_2(x_0), \pi_2(y_0)) = (s, t) \in \mathcal{E}(R)$.
- Suppose the $\geqslant$-rule is applicable to $\mathcal{F}_1$. Then, $(\mathcal{T}_1, \mathcal{F}_1) \Rightarrow_{\geqslant} (\mathcal{T}_2, \mathcal{F}_2)$, there exists x such that $\geqslant nR.C \in \mathcal{F}_1(x) = \mathcal{F}_2(x)$ and n distinct successors y_i of x are generated for x such that

$\mathcal{F}_2(x, y_i) = \{R\}$ and $\mathcal{F}_2(y_i) = \{C\}$ for $1 \le i \le n$. Since π_1 is a structure preserving map, $\geqslant nR.C \in \mathcal{F}_1(x) \subseteq \mathcal{L}(\pi_1(x))$. By (P10), we have that there exist n distinct $t_i \in \mathbf{S}$ such that $C \in \mathcal{L}(t_i)$ and $(s, t_i) \in \mathcal{E}(R)$ for $s = \pi_1(x)$. We define π_2 from π_1 by extending the domain of π_1 with n elements y_i and setting $\pi_2(y_i) = t_i$. To prove that π_2 is a structure preserving map, it is enough to consider these three cases (the rest are trivial):

1. We prove the first condition for the n new nodes y_i. By the above,
$\mathcal{F}_2(y_i) = \{C\} \subseteq \mathcal{L}(t_i) = \mathcal{L}(\pi_2(y_i))$.
2. We prove the second condition for the n new pairs (x, y_i). By the above,
$(\pi_2(x), \pi_2(y_i)) = (s, t_i) \in \mathcal{E}(R)$.
3. We prove the third condition for the n new nodes. Let $y_i \not\approx y_j$ for $i, j \in \{1, \dots, n\}$. By the definition of π_2 given above,
$\pi_2(y_i) = t_i \neq t_j = \pi_2(y_j)$.

Theorem 4. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{M})$ be a $\mathcal{SHIQM}$-ontology. If $\mathcal{O}$ has a tableau, then the expansion rules for $\mathcal{SHIQM}$ can be applied to $\mathcal{O}$ such that they yield a complete completion forest that has no contradictions and has no cycles w.r.t. $\mathcal{M}$.*

Proof. Let $\mathbb{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J})$ be a $\mathcal{SHIQM}$-tableau for $\mathcal{O}$. Let $\mathcal{F}_0$ be the completion forest after running the initialization in the Tableau algorithm with input $\mathcal{O}$. We define $\pi_0 : \mathcal{F}_0 \to \mathbb{T}$ as $\pi_0(a) = \mathcal{J}(a)$ for all individuals in $\mathcal{O}$. We prove that π_0 satisfies the conditions of Definition 33. These conditions are proved for the individuals a, b that are in the Abox since these are the only nodes of the initial forest $\mathcal{F}_0$.

1. We have to prove that $\mathcal{F}_0(a) \subseteq \mathcal{L}(\pi_0(a))$. Assume $C \in \mathcal{F}_0(a)$. By the initialization (Definition 24), there exists $a' \approx a$ and $C(a') \in \mathcal{A}$.
$$\begin{aligned} C &\in \mathcal{L}(\mathcal{J}(a')) && \text{by (P13)}\\ &= \mathcal{L}(\mathcal{J}(a)) && \text{by (P16)}\\ &= \mathcal{L}(\pi_0(a)) && \text{by def. of } \pi_0. \end{aligned}$$
2. Suppose b is an S-neighbour of a. Then, there exists an edge from a to b in $\mathcal{F}_0$ such that $R \in \mathcal{F}_0(a, b)$ and $R \sqsubseteq^* S$ or an edge from b to a in $\mathcal{F}_0$ such that $R \in \mathcal{F}_0(b, a)$ and $R \sqsubseteq^* Inv(S)$. By the initialization (Definition 24), in the first case there are $a' \approx a$ and $b' \approx b$ such that $R(a', b') \in \mathcal{A}$.
$$\begin{aligned} (\pi_0(a), \pi_0(b)) &= (\mathcal{J}(a), \mathcal{J}(b)) && \text{by def. of } \pi_0\\ &= (\mathcal{J}(a'), \mathcal{J}(b')) && \text{by (P16)}\\ &\in \mathcal{E}(R) && \text{by (P14)}\\ &\subseteq \mathcal{E}(S) && \text{by (P8)}. \end{aligned}$$
3. Suppose $a \not\approx b$. By the initialization (Definition 24), there are a', b' such that $a' \approx a$, $b \approx b'$ and $a' \neq b' \in \mathcal{A}$.
$$\begin{aligned} \pi_0(a) &= \mathcal{J}(a) && \text{by def. of } \pi_0\\ &= \mathcal{J}(a') && \text{by (P16)}\\ &\neq \mathcal{J}(b') && \text{by (P15)}\\ &= \mathcal{J}(b) && \text{by (P16)}\\ &= \pi_0(b) && \text{by def. of } \pi_0. \end{aligned}$$
4. Similarly to the previous one.

It follows from Theorem 1 and Lemma 19 that there exists a complete completion forest $\mathcal{F}$ and a structure preserving map $\pi : \mathcal{F} \to \mathbb{T}$ such that $\pi(a) = \mathcal{J}(a)$ for all a in $\mathcal{O}$. From (P1), (P9) and the fact that π is a structure preserving map, we can deduce that $\mathcal{F}$ does not have any contradictions. Note that $x \approx y$ and $x \not\approx y$ cannot be in $\mathcal{F}$ by the third and fourth condition of structure preserving map.

We will prove that $\mathcal{F}$ has no cycles using (P17). Suppose towards a contradiction that $\mathcal{F}$ has a cycle. Then, there would be a set of meta-modelling axioms $A_0 =_m a_0, A_1 =_m a_1, \dots, A_n =_m a_n$ all in $\mathcal{M}$ such that
$$\begin{array}{ll} A_1 \in \mathcal{F}(x_0) & x_0 \approx a_0\\ A_2 \in \mathcal{F}(x_1) & x_1 \approx a_1\\ \vdots & \vdots\\ A_n \in \mathcal{F}(x_{n-1}) & x_{n-1} \approx a_{n-1}\\ A_0 \in \mathcal{F}(x_n) & x_n \approx a_n. \end{array}$$
Using (P17) and the fact that π is a structure preserving map such that $\pi(a) = \mathcal{J}(a)$ for all a in $\mathcal{O}$, we have that:
$$\begin{array}{ll} \pi(a_0) = \mathcal{J}(a_0) = \{x \in \mathbf{S} \mid A_0 \in \mathcal{L}(x)\} & A_1 \in \mathcal{L}(\pi(a_0))\\ \pi(a_1) = \mathcal{J}(a_1) = \{x \in \mathbf{S} \mid A_1 \in \mathcal{L}(x)\} & A_2 \in \mathcal{L}(\pi(a_1))\\ \vdots & \vdots\\ \pi(a_n) = \mathcal{J}(a_n) = \{x \in \mathbf{S} \mid A_n \in \mathcal{L}(x)\} & A_0 \in \mathcal{L}(\pi(a_n)). \end{array}$$
Then, we have that:
$$\pi(a_0) \in \pi(a_1) \in \cdots \in \pi(a_n) \in \pi(a_0).$$
This contradicts the fact that $\mathbf{S}$ is well founded (see Definition 28).

The following corollary follows from Lemma 14, Theorems 1, 3 and 4.

Corollary 3 (*Correctness of Tableau for $\mathcal{SHIQM}$*). *The Tableau algorithm is a decision procedure for the consistency of knowledge bases in $\mathcal{SHIQM}$.*

8. Meta-modelling level

In this section, we introduce the notions of meta-modelling level of an ontology and show how to compute it. From now on, we assume that all the individuals with meta-modelling are inhabited, i.e., for all $\mathcal{O} \models a =_m A$, there exists an individual b such that $\mathcal{O} \models A(b)$.

Definition 34 (*Meta-Modelling Level*). The *meta-modelling level of an interpretation* $\mathcal{I}$ – denoted as $\mathsf{level}(\mathcal{I})$ – is the smallest n such that $\Delta^{\mathcal{I}} \subseteq S_n$.

A concept C is *at level n in the interpretation* $\mathcal{I}$ – denoted as $\mathsf{level}(\mathcal{I}, C)$ – if n is the smallest natural number such that $C^{\mathcal{I}} \subseteq S_n$.

The *meta-modelling level of an ontology* $\mathcal{O}$ – denoted as $\mathsf{level}(\mathcal{O})$ – is the smallest n where n is the level of some model of $\mathcal{O}$.

A concept C of a consistent ontology $\mathcal{O}$ is *at level n* – denoted as $\mathsf{level}(\mathcal{O}, C)$ – if n is the smallest natural number such that $C^{\mathcal{I}} \subseteq S_n$ and $\mathcal{I}$ is a model of $\mathcal{O}$.

Example 16. Suppose we have an ontology $\mathcal{O}$ where the ABox has the assertions $A(a_0), B(b_0), C(c_0), P(k, b), P(k, a_0), Q(f, c), Q(f, b_0)$ and the MBox has the axioms $a =_m A$, $b =_m B$, $c =_m C$. In this case, we have that $\mathsf{level}(\mathcal{O}) = 1$.

Example 17. We now consider the ontology of Example 16, with the following Tbox:
$$\top \sqsubseteq (\leqslant 1\, P.\top) \sqcup (\leqslant 1\, Q.\top).$$
In this case, we have that $\mathsf{level}(\mathcal{O}) = 2$. This is because in any model $\mathcal{I}$, we have that either $b^{\mathcal{I}} = (a_0)^{\mathcal{I}}$ or $c^{\mathcal{I}} = (b_0)^{\mathcal{I}}$. It is easy to see that there exists a model of this ontology which has level 2 and that also satisfies $b = a_0$ and $c \neq b_0$. Similarly, there exists a model of this ontology which has level 2 that satisfies $b \neq a_0$ and $c = b_0$. Any other model has greater level.

We define a model with level 2 that satisfies $b = a_0$ and $c \neq b_0$ as follows.

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{b_0, c_0, \{b_0\}, \{c_0\}, \{\{b_0\}\}\} \\
(b_0)^{\mathcal{I}} &= b_0 \\
(c_0)^{\mathcal{I}} &= c_0 \\
c^{\mathcal{I}} &= C^{\mathcal{I}} = \{c_0\} \\
b^{\mathcal{I}} &= B^{\mathcal{I}} = (a_0)^{\mathcal{I}} = \{b_0\} \\
a^{\mathcal{I}} &= A^{\mathcal{I}} = \{(a_0)^{\mathcal{I}}\} = \{\{b_0\}\}.
\end{aligned}$$

In the interpretation $\mathcal{I}$ given above, we have that $\mathsf{level}(\mathcal{I}, B) = \mathsf{level}(\mathcal{I}, C) = 0$, $\mathsf{level}(\mathcal{I}, A) = 1$ and $\mathsf{level}(\mathcal{I}, \top) = 2$.

We now define a model with level 2 that satisfies $b \neq a_0$ and $c = b_0$.

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \{a_0, c_0, \{a_0\}, \{c_0\}, \{\{c_0\}\}\} \\
(a_0)^{\mathcal{I}} &= a_0 \\
(c_0)^{\mathcal{I}} &= c_0 \\
a^{\mathcal{I}} &= A^{\mathcal{I}} = \{a_0\} \\
b^{\mathcal{I}} &= B^{\mathcal{I}} = \{(b_0)^{\mathcal{I}}\} = \{\{c_0\}\} \\
c^{\mathcal{I}} &= C^{\mathcal{I}} = \{(c_0)^{\mathcal{I}}\} = \{c_0\}.
\end{aligned}$$

In the interpretation $\mathcal{I}$ given above, we have that $\mathsf{level}(\mathcal{I}, A) = \mathsf{level}(\mathcal{I}, C) = 0$, $\mathsf{level}(\mathcal{I}, B) = 1$ and $\mathsf{level}(\mathcal{I}, \top) = 2$.

In the ontology defined in Example 17, $\mathsf{level}(\mathcal{O}, A) = \mathsf{level}(\mathcal{O}, B) = \mathsf{level}(\mathcal{O}, C) = 0$. Note that the level of A and B in this ontology is 0 but these two concepts do not share a common interpretation where they both have level 0.

The level of the model found by the tableau algorithm is in itself an upper bound for the meta-modelling level of an ontology but it may not be the minimal one. In order to compute the meta-modelling level of an ontology, we could compute all the complete and consistent tableau graphs and choose the model that has minimum level. Since this method is very inefficient, we will propose a different algorithm that may not always give the exact level but just a range of values where it belongs.

It follows from Lemma 16 that the level of the model $\mathcal{I}$ given by the tableau algorithm is bounded by the cardinality m of the Mbox. So, we have that $\mathsf{level}(\mathcal{O}) \leq \mathsf{level}(\mathcal{I}) \leq m = \sharp(\mathcal{M})$.

We will define a function lb that computes a lower bound for $\mathsf{level}(\mathcal{O})$ and $\mathsf{level}(\mathcal{O}, C)$. The following lemma allows us to define this function recursively. The set of individuals of an ontology is denoted by $\mathsf{Individuals}(\mathcal{O}) = \{a \mid a \in \mathcal{O}\}$.

Definition 35. We define a relation $\blacktriangleleft$ on the set $\mathsf{Individuals}(\mathcal{O})$ as $b \blacktriangleleft a$ if $\mathcal{O} \models a =_{\mathsf{m}} A$ and $\mathcal{O} \models A(b)$.

Lemma 20. *The relation $\blacktriangleleft$ is well founded.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{O}$. Suppose towards a contradiction that $\blacktriangleleft$ is not well-founded. By Lemma 7, $\blacktriangleleft$ has a cycle,

$$a_0 \blacktriangleleft a_1 \blacktriangleleft a_2 \blacktriangleleft \cdots \blacktriangleleft a_{n-1} \blacktriangleleft a_0.$$

By the definitions of $\blacktriangleleft$ and $\models$, we have that

$$\begin{array}{ll}
(a_1)^{\mathcal{I}} = (A_1)^{\mathcal{I}} & (a_0)^{\mathcal{I}} \in (A_1)^{\mathcal{I}} \\
(a_2)^{\mathcal{I}} = (A_2)^{\mathcal{I}} & (a_1)^{\mathcal{I}} \in (A_2)^{\mathcal{I}} \\
\vdots & \\
(a_0)^{\mathcal{I}} = (A_0)^{\mathcal{I}} & (a_{n-1})^{\mathcal{I}} \in (A_0)^{\mathcal{I}}.
\end{array}$$

So, there is a cycle in the domain of the interpretation $\mathcal{I}$ contradicting the first clause in Definition 19.

We now define the function lb that computes a lower bound for $\mathsf{level}(\mathcal{O})$ and $\mathsf{level}(\mathcal{O}, C)$.

Definition 36 (*Lower Bound*)**.** For an individual a in $\mathcal{O}$ we define $\mathsf{lb}(a)$ as follows:

$$\mathsf{lb}(a) = 0 \quad \text{if there is no } A \text{ such that } \mathcal{O} \models a =_{\mathsf{m}} A$$
$$\mathsf{lb}(a) = \max\{\mathsf{lb}(b) \mid \mathcal{O} \models a =_{\mathsf{m}} A,\ \mathcal{O} \models A(b)\} + 1.$$

For an ontology $\mathcal{O}$ we define $\mathsf{lb}(\mathcal{O})$ as follows.

$$\mathsf{lb}(\mathcal{O}) = \max\{\mathsf{lb}(a) \mid a \in \mathcal{O}\}.$$

For a concept C of an ontology $\mathcal{O}$ we define $\mathsf{lb}(\mathcal{O}, C)$ as follows.

$$\mathsf{lb}(\mathcal{O}, C) = \max\{\mathsf{lb}(a) \mid \mathcal{O} \models C(a)\}.$$

Note that C is a meta-concept iff $\mathsf{lb}(\mathcal{O}, C) \geq 1$. We now prove that lb is a lower bound for the meta-modelling level of an ontology. This bound may be strictly less than the meta-modelling level. In Example 16, we have that $\mathsf{level}(\mathcal{O}) = 1 = \mathsf{lb}(\mathcal{O})$, while in Example 17 we have that $\mathsf{lb}(\mathcal{O}) = 1 < \mathsf{level}(\mathcal{O}) = 2$.

Lemma 21. *Let $\mathcal{O}$ be a consistent ontology. Then,* $\mathsf{lb}(\mathcal{O}, C) \leq \mathsf{level}(\mathcal{O}, C)$.

Proof. We prove that $\mathsf{level}(\mathcal{I}, C) \geq \mathsf{lb}(\mathcal{O}, C)$ for an arbitrary model $\mathcal{I}$ of $\mathcal{O}$. Let $\mathsf{lb}(\mathcal{O}, C) = n$. Then, it follows from Definitions 36 and 35 and the fact that the set $\mathsf{Individuals}(\mathcal{O})$ is finite that there exists a finite sequence

$$a_0 \blacktriangleleft a_1 \blacktriangleleft a_1 \blacktriangleleft \cdots \blacktriangleleft a_{n-1} \blacktriangleleft a_n.$$

By the definitions of $\blacktriangleleft$ and $\models$, we have that

$$\begin{array}{ll}
(a_n)^{\mathcal{I}} \in C^{\mathcal{I}} & \\
(a_n)^{\mathcal{I}} = (A_n)^{\mathcal{I}} & (a_{n-1})^{\mathcal{I}} \in (A_n)^{\mathcal{I}} \\
(a_{n-1})^{\mathcal{I}} = (A_{n-1})^{\mathcal{I}} & (a_{n-2})^{\mathcal{I}} \in (A_{n-1})^{\mathcal{I}} \\
\vdots & \\
(a_1)^{\mathcal{I}} = (A_1)^{\mathcal{I}} & (a_0)^{\mathcal{I}} \in (A_1)^{\mathcal{I}}.
\end{array} \tag{5}$$

Let $\mathsf{level}(\mathcal{I}, C) = m$ and $C^{\mathcal{I}} \subseteq S_m$. Since $(a_n)^{\mathcal{I}} \in C^{\mathcal{I}}$, we have that

$$(a_n)^{\mathcal{I}} \in S_m. \tag{6}$$

It follows from (5), (6) and Lemma 9 that $m \geq n$.

A direct corollary from the above result is that $\mathsf{lb}(\mathcal{O}) \leq \mathsf{level}(\mathcal{O})$, since $\mathsf{lb}(\mathcal{O}, \top) = \mathsf{lb}(\mathcal{O})$ and $\mathsf{level}(\mathcal{O}, \top) = \mathsf{level}(\mathcal{O})$.

As we mentioned before, in order to compute the meta-modelling level of an ontology, we could compute all the complete and consistent tableau graphs and choose a model that has minimum level. Since this method is very inefficient, we instead propose the following algorithm:

1. Run tableau for checking consistency of the ontology and getting a model $\mathcal{I}$.
2. Let $n = \mathsf{lb}(\mathcal{O})$ and $m = \mathsf{level}(\mathcal{I})$.
   If $n = m$ then the level of the ontology is n
   Otherwise the level of the ontology is between n and m.

Something similar can be done for the meta-modelling level of a concept. It is enough to substitute $\mathsf{lb}(\mathcal{O})$ by $\mathsf{lb}(\mathcal{O}, C)$ and $\mathsf{level}(\mathcal{I})$ by $\mathsf{level}(\mathcal{I}, C)$.

9. Related work

In this section, we discuss other approaches to meta-modelling that appear in the literature. Table 1 gives a summary of our comparisons.

Punning. OWL 2 DL has a very restricted form of meta-modelling called *punning* [3]. In spite of the fact that the same identifier can be used simultaneously as an individual and as a concept, they are semantically different. In order to use the punning of OWL 2

Table 1
Comparison of meta-modelling approaches.

Approach	Base DL	Unlim. levels	Flexible structure	Inter-layer roles	Intensional regularity	Extensionality	Well-foundedness	Meta-modelling for roles
Punning	$\mathcal{SROIQ}$	Y	Y	Y	N	N	N	Y
Pan et al.	$\mathcal{SROIQ}$	Y	N	N	Y	Y	Y	Y
Motik (υ-sem.)	$\mathcal{ALCHIQ}$	Y	Y	Y	Y	N	N	Y
De Giacomo et al.	$\mathcal{SHIQ}$	Y	Y	Y	Y	N	N	Y
Glimm et al.	$\mathcal{SROIQ}$	N	N	Y(1–2)	Y	N	Y	N
Homola et al.	$\mathcal{SROIQ}$	Y	N	Y	Y	N	Y/N	N
$\mathcal{SHIQM}$	$\mathcal{SHIQ}$	Y	Y	Y	Y	Y	Y	N

DL in the example of Fig. 1, we could change the name *river* to *River* and *lake* to *Lake*. In spite of the fact that the identifiers look syntactically equal, OWL would not detect certain inconsistencies as the ones illustrated in Examples 1–9. In the first example, OWL will not detect that there is a circularity and in the other examples, OWL will not detect that there is a contradiction. Apart from having the disadvantage of not detecting certain inconsistencies, this approach is not natural for reusing ontologies. For these scenarios, it is more useful to assume the identifiers be syntactically different and allow the user to equate them by using axioms of the form $a =_{\mathrm{m}} A$.

Integrated Meta-modelling in OWL 2. Glimm et al. do not define a set-theoretical semantics for meta-modelling. Instead, they codify meta-modelling within OWL DL [17]. This encoding is used to formalize the rules from the OntoClean methodology in OWL [18]. This approach has the limitation of having only two levels of meta-modelling (concepts and meta-concepts) and it is not enough for "fully" detecting inconsistencies coming from meta-modelling, e.g., Example 8. On the other hand, by encoding inclusion $C \sqsubseteq D$ between concepts using the role *subclass*, the authors are able to express the rules of the OntoClean methodology such as

$$\frac{a =_{\mathrm{m}} A \quad b =_{\mathrm{m}} B \quad A \sqsubseteq B \quad RigidClass(a)}{\neg AntiRigidClass(b)}$$

inside OWL 2. It does not seem possible to express such a rule by using only the meta-modelling expressive power of $\mathcal{SHIQM}$.

Henkin vs. Hilog Semantics. Our semantics follows the style of the so-called Henkin's semantics in the sense that all syntactic higher order objects have a direct set-theoretical interpretation via a hierarchy of power sets.[2] This is also the style of semantics followed by Pan et al. [5,6]. The semantics for meta-modelling given by Motik, De Giacomo et al. and Homola et al. follow a Hilog style semantics [2,16,7,8].[3] In this style of semantics, the same syntactic object can have different interpretations depending on the position or role it plays in a sentence, e.g., A is an individual in $B(A)$ and the same A is a concept in $A(B)$. The first A playing the role of an individual does not always have the same interpretation as the second A which plays the role of a concept. The main drawback of Hilog semantics is that it cannot really express that the interpretation of a given symbol taken as individual is the same as the interpretation of another (or the same) symbol taken as concept. As a consequence, the Hilog style semantics for meta-modelling is weaker than the Henkin semantics since it does not satisfy extensionality (see Remark 1 and Lemma 10). We think that Henkin's style semantics is more appropriate for our applications since besides being more direct, it allows us to check for inconsistencies which are not detected with the Hilog semantics, e.g., Examples 7–9.

υ-semantics. Motik proposes a solution for meta-modelling that is not so expressive as RDF but which is decidable [2]. He defines two alternative semantics: the context approach (π-semantics) and the HiLog approach (υ-semantics). The context approach is similar to the so-called punning supported by OWL 2 DL. The HiLog semantics looks more useful than the context semantics since it can detect the inconsistency of Example 2. Apart from the fact that this semantic does not satisfy extensionality, it also ignores the issue on well-founded sets. One can have an ontology with $A(A)$ which is satisfiable in the Hilog Semantics. The model is $\Delta^{\mathcal{I}} = \{X\}$ where $X = \{X\}$ is a set that belongs to itself. The conditions in the definition of Hilog semantics are satisfied since $A^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$. The algorithm given by Motik [2, Theorem 2] does not check for circularities (see Example 5) which is one of the main contributions of this paper. Moreover, as mentioned before, his semantics being a Hilog semantics cannot detect the inconsistencies of some ontologies such as Examples 7–9. Since his syntax does not restrict the sets of individuals, concepts and roles to be pairwise disjoint, an identifier can be used as a concept (or role) and an individual at the same time. In this way, Motik can have meta-modelling for concepts as well as for roles. The description logic $\mathcal{SHIQM}$ presented in this paper has only meta-modelling for concepts but not for roles.

Higher order description logic. De Giacomo et al. specify a formalism, called *higher order description logic*, that allows to treat the same symbol of the signature as an instance, a concept and a role [16,21]. They use a semantics very similar to Motik's Hilog semantics, with a single domain (without layers) in which non-well founded sets are allowed and only intensional regularity holds. Their approach makes some ontologies *wrongly consistent* since

1. Hilog semantics allows some undesired models which give different interpretations to an individual and a concept representing the same object.
2. The domain Δ can contain sets that are not well-founded.

Ontologies that are wrongly inconsistent in this approach due to (1) are given in Examples 7–9. While an example of an ontology which is wrongly consistent in this approach due to (2) is Example 1.

OWL FA. Pan et al. address meta-modelling by defining different "layers" or "strata" within a knowledge base [5,6]. Their semantics can be easily seen to belong to well-founded set theory provided the universe at level 0 is a set of basic objects. Their semantics satisfies extensionality and interpret the individual and the concept connected by meta-modelling as the same object. Though they have meta-modelling for roles (which we do not), all the individuals of a certain concept need to be at the same level. The fixed layer approach forces the user to explicitly write the information of the layer in the concept. This has several disadvantages: the user should know beforehand in which layer the concept lies and it does not give the flexibility of changing the layer in which it lies. Neither it allows us to mix different layers when building concepts, inclusions or roles, e.g., we cannot define a role whose domain and range live in different layers.

Typed higher order description logic. Homola et al. define a *typed higher order description logic* [7,8] where atomic concept and role names are "typed" with the layer or level of meta-modelling. The so-called typed higher order DL is more expressive than OWL FA

[2] The name "Henkin" comes from the semantics given by Henkin to the theory of types [19].

[3] The name Hilog comes from the semantics given to higher order logic programming [20]. This is also the style of semantics followed by RDF [3].

$$\begin{array}{ccc} S'_n & \subset & S_n \\ \cap & & \cap \\ S''_n & \subset & S'''_n \end{array}$$

Fig. 10. Comparison of interpretation domains.

since roles have a source and a target level. However, this logic still lacks expressibility which we think important for integrating ontologies. As in OWL FA, all the individuals of a certain concept must be at the same level. This means that elements with different levels of meta-modelling cannot coexist in the same set, e.g. the set *GeographicObject* in Fig. 2. One cannot perform the union of concepts at different levels or have a role whose domain or range have individuals at different levels, e.g., the role *manages* in Fig. 8. Like OWL FA, this approach also forces the user to know the levels of the meta-modelling for each atomic concept and role. Homola et al. also analyse two semantic approaches: a Henkin semantics and a Hilog-style semantics [7]. The Henkin semantics (theirs as well as ours) satisfy both properties of intensional regularity and extensionality (see Remark 1). The Hilog semantics is weaker than the Henkin semantics and it only satisfies intensional regularity.

Comparing well-founded domains. Different domains defined for meta-modelling in the literature are compared with our set S_n in Fig. 10. The Henkin semantics by Homola et al. has one domain for each level n defined as $S'_{n+1} = \mathcal{P}(S'_n)$ [7]. Pan et al. define one domain for each level n but they also include relations $S''_{n+1} = \mathcal{P}(S''_n) \cup \mathcal{P}(S''_n \times S''_n)$ [5]. A similar domain to S''_n is defined by Kaushik et al. to give semantics of algebra operators that combine ontologies in RDF [22]. As future work we will consider $S'''_{n+1} = S'''_n \cup \mathcal{P}(S'''_n) \cup \mathcal{P}(S'''_n \times S'''_n)$, to include meta-modelling for roles.

In our case a set $C^{\mathcal{I}} = X \subseteq \Delta \subseteq S_n$ can contain elements x such that $x \in S_i$ for any $i \leq n$. This means that elements with different levels of meta-modelling can coexist in a set $X \subseteq S_n$, e.g., the set *GeographicObject* in Fig. 2. However, a set $X \subseteq S'_{n+1}$ (or S''_{n+1}) can have only elements that are in S'_n (or S''_n). In other words, elements with different levels of meta-modelling cannot coexist. The sets S''_n also include relations for capturing meta-modelling for roles which we do not consider in this paper.

10. Conclusions

In this paper we have shown a novel approach to meta-modelling which consists in adding equations between individuals and concepts. From the point of view of real applications of meta-modelling, our work combines two main advantages which are not both present in existing approaches: (i) the freedom to model through a flexible structure of meta-modelling levels, and (ii) a consistency mechanism which prevents from design errors such as non-well foundedness or contradictions that come because of meta-modelling.

Reusability. We think that our approach result more natural in a scenario where we want to reuse a set of independent ontologies to build a knowledge base for a given application. It is usually the case that the same real object is represented with different granularity in different ontologies, e.g., as an individual in one ontology and as a concept in another one. Then, without altering the original ontologies, we can express through a meta-modelling axiom that the interpretation of an individual in one ontology is (in a direct way) the same as the interpretation of a concept in the other ontology. That is, given a model of the knowledge base, each symbol will have a single interpretation, regardless the axiom in which it is placed.

Flexible meta-modelling hierarchy. If we analyse the approaches which define fixed layers or levels of meta-modelling [5–8] we observe that they impose a very strong limitation to the ontology engineer. Not always the instances of a concept need to be represented with the same granularity. For example, the concept *GeographicObject* of Fig. 2 has two instances, *hydrographic* and *flora* which are meta-concepts and an instance, *physiographic* that is an individual without meta-modelling. Perhaps the ontology engineer does not have access to an ontology about physiography as in the case of hydrography and flora, or perhaps obtaining more detail about physiographic objects is not what matters for the particular application. But if in the future, the need to integrate the ontology Geographic Objects Politics with an ontology about physiography arises, it is enough to merely equate the individual *physiographic* to a concept whose meta-modelling level will depend on the granularity of the particular ontology describing physiographic objects. Moreover, following the fixed layers approach, when several ontologies are integrated through meta-modelling and other links, such as roles or mapping of concepts, the ontology engineer has to synchronize the meta-modelling levels along with the other kind of relations among the ontologies. In our approach, we infer the meta-modelling level of each concept and check for inconsistencies through the proposed tableau algorithm. So, the ontology engineer is not attached to a rigid structure of layers.

Inference of meta-modelling level. The fixed layer approach is analogous to *typing à la Church* in typed lambda calculus where the user needs to annotate or declare the type of all his variables, e.g., this is the case in the proof assistant Coq [23]. Our approach is analogous to *typing à la Curry* where the user does not need to declare the types of the variables because the system will infer them instead e.g., this is the case in the functional programming language Haskell [24]. Inferring the meta-modelling level of a concept (or an ontology) is, however, a much more difficult problem than inferring the type of a program (a λ-term). One can infer the type of a λ-term from the syntax or shape of the term [25,26]. But one cannot infer the meta-modelling level just by looking at the syntax or shape of the concept and the knowledge base. In order to infer the meta-modelling level, one has to take into account the semantics. The syntax gives us an upper bound of the meta-modelling level which is the cardinality of the Mbox. However, in order to know whether this meta-modelling level exists or not we need to analyse the semantics and check consistency, e.g., run the tableau algorithm and find out a completion forest that has neither contradictions nor cycles.

Well-founded semantics. An interesting and original contribution of this paper from the theoretical point of view is the incorporation of the notion of well-founded set in the semantics of the logic as well as in the tableau algorithm. We think it is important to restrict the domain to be a well-founded set. In principle, non well-founded sets are not a source of contradictions as it is shown by studies on non-well founded set theory, e.g., [4]. The reason why we exclude non well-founded sets is because we think that non well-founded sets do not occur in the applications we are interested in. Note that we cannot replace the restriction $\Delta \subseteq S_n$ by a weaker one and just require that the domain Δ of the interpretation is a well-founded set. This looks more neat from the theoretical point of view. However, we are certainly not interested in having an infinite (or transfinite) number of levels of meta-modelling for our applications.

Unique domain of mixed-levels. An important difference with the fixed level approach to meta-modelling is that we do not have a domain separated in layers where Δ is *exactly* the union of a family of disjoint domains Δ_n for each $n \in \mathbb{N}$. Instead, we have *only one* domain which is a *subset* of S_n where elements with different levels of meta-modelling can coexist. Having *only one domain* makes a difference in the treatment of negation, e.g., Example 4. If we had the stricter requirement saying that Δ is *exactly the same as* S_n, we would be putting too many elements in Δ that are not necessary. It is important for the domain of our canonical interpretation $\mathbf{S}' = \mathsf{set}(\mathsf{Paths}(\mathcal{F}))$ that could be a proper subset of S_n.

Detecting contradictions for meta-modelling. The key feature in our semantics is to interpret a and A as the same object when a and A are connected through meta-modelling, i.e., if $a =_{\mathrm{m}} A$ then $a^{\mathcal{I}} = A^{\mathcal{I}}$. This allows us to detect the inconsistencies in the ontologies shown in the examples of Section 5 which is not possible under the Hilog semantics.

11. Future research

We plan to extend our approach to support all of OWL 2. We think it will be enough to extend the corresponding tableau algorithm to include our expansion rules for meta-modelling. Adding property chains and data types seems to be orthogonal to meta-modelling since we are not interested in having meta-modelling for data types [27]. Keys only deal with individuals (not variables) and the meta-modelling rules will add all the combinations of equalities and inequalities that are necessary [28]. Our expansion rules seem to work with nominals as well [29]. We also think that we will be able to apply the same proof techniques developed in this paper for proving correctness of these extensions but in a more complex setting.

At the moment we are working on extending Pellet to include our new expansion rules for handling meta-modelling [30,31]. For our specific tableau algorithm, a good optimization will consists in restricting the application of the expansion rules for meta-modelling: *the* $\approx$, $\not\approx$ *and close-rule should be applied only if the other rules are not applicable*. Since the worse case can happen when all the individuals with meta-modelling are equated (since the TBox is augmented with new axioms), one should try first the choice when all the individuals are different and *postponing the* $\approx$*-rule as much as possible*. At the moment, we are also studying conditions on the Mbox to have a PSpace tableau algorithm for $\mathcal{ALCM}$ and $\mathcal{ALCQM}$ when the TBox is definitorial or simple [32,33] and we will also study complexity for increasingly expressive logics.

In this paper, we have only considered meta-modelling for concepts. We plan to study extensions that include meta-modelling for roles.

Acknowledgements

The third author would like to acknowledge a Daphne Jackson fellowship sponsored by EPSRC and the University of Leicester. We are also grateful to Diana Comesaña for sharing with us the data from her ontology network on geographic objects [1].

References

[1] Servicio Geográfico Militar del Uruguay (SGM). Catálogo de objetos y símbolos geográficos. At http://www.sgm.gub.uy/index.php/component/content/article/49-noticias/novedades/124-catalogoversion1/ (Last date accessed November 2014).
[2] B. Motik, On the properties of metamodeling in OWL, in: International Semantic Web Conference, 2005, pp. 548–562.
[3] P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web Technologies, Chapman & Hall/CRC, 2009.
[4] K.J. Barwise, L. Moss, Vicious Circles. On the Mathematics of Non-Wellfounded Phenomena, Cambridge University Press, 1996.
[5] J.Z. Pan, I. Horrocks, G. Schreiber, OWL FA: A metamodeling extension of OWL DL, in: OWLED, 2005.
[6] N. Jekjantuk, G. Gröner, J.Z. Pan, Modelling and reasoning in metamodelling enabled ontologies, Int. J. Softw. Inform. 4 (3) (2010) 277–290.
[7] M. Homola, J. Kluka, V. Svátek, M. Vacura, Towards typed higher-order description logics, in: DL, 2013.
[8] M. Homola, J. Kluka, V. Svátek, M. Vacura, Typed higher-order variant of $\mathcal{SROIQ}$—Why not?, in: DL, 2014.
[9] I. Horrocks, U. Sattler, S. Tobies, Reasoning with individuals for the description logic $\mathcal{SHIQ}$, in: CADE, 2000, pp. 482–496.
[10] R. Motz, E. Rohrer, P. Severi, Reasoning for ALCQ extended with a flexible meta-modelling hierarchy, in: Semantic Technology-4th Joint International Conference, JIST 2014. Revised Selected Papers, pp. 47–62. Long version Available at: http://arxiv.org/abs/1410.7953.
[11] E. Mendelson, Introduction to Mathematics, fourth ed., Chapman and Hall/CRC, 1997.
[12] I. Horrocks, U. Sattler, A description logic with transitive and inverse roles and role hierarchies, J. Logic Comput. 9 (3) (1999) 385–410.
[13] I. Horrocks, U. Sattler, S. Tobies, Practical reasoning for expressive description logics, in: LPAR 1999, pp. 161–180.
[14] J. Winfried, M. Weese, Discovering Modern Set theory. I, American Mathematical Society, 1998.
[15] G. Winskel, Notes and Exercises for Set Theory for Computer Science (pdf) an MPhil course in Advanced Computer Science at the University of Cambridge. (Last data accessed January 2015).
[16] G. De Giacomo, M. Lenzerini, R. Rosati, Higher-order description logics for domain metamodeling, in: AAAI, 2011.
[17] B. Glimm, S. Rudolph, J. Völker, Integrated metamodeling and diagnosis in OWL 2, in: International Semantic Web Conference (1), 2010, pp. 257–272.
[18] N. Guarino, C. Welty, An overview of OntoClean, in: Handbook on Ontologies, Springer, 2004, pp. 151–171.
[19] L. Henkin, Completeness in the theory of types, J. Symbolic Comput. 15 (2) (1950) 81–91.
[20] W. Chen, M. Kifer, D.S. Warren, A foundation for higher order programming, J. Log. Program. 15 (3) (1993) 187–230.
[21] M. Lenzerini, L. Lepore, A. Poggi, Making metaquerying practical for Hi(DL-LiteR) knowledge bases, in: On the Move to Meaningful Internet Systems: OTM 2014 Conferences, 8841, 2014, pp. 580–596.
[22] S. Kaushik, C. Farkas, D. Wijesekera, P. Ammann, An algebra for composing ontologies, in: B. Bennett, C. Fellbaum (Eds.), FOIS, in: Frontiers in Artificial Intelligence and Applications, vol. 150, IOS Press, 2006, pp. 265–276.
[23] Proof Asssistant Coq. https://coq.inria.fr/ (Last data accessed January 2015).
[24] Functional Programming Language Haskell. https://wiki.haskell.org/Haskell (Last data accessed January 2015).
[25] R. Hindley, The principal type-scheme of an object in combinatory logic, Trans. Amer. Math. Soc. 146 (1969) 29–60.
[26] D. Milner, A theory of type polymorphism in programming, J. Comput. Syst. Sci. (JCSS) 17 (1978) 348–374.
[27] J.Z. Pan, I. Horrocks, Extending datatype support in web ontology reasoning, in: On the Move to Meaningful Internet Systems, 2002—DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002 Irvine, Proceedings, 2002, pp. 1067–1081.
[28] B. Parsia, U. Sattler, T. Schneider, Easy keys for OWL, in: OWLED, 2008.
[29] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible $\mathcal{SROIQ}$, in: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), AAAI Press, 2006, pp. 57–67.
[30] Clark, Parsia, Pellet. http://clarkparsia.com/pellet/ (Last date accessed October 2014).
[31] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL reasoner, J. Web Semant. 5 (2) (2007) 51–53.
[32] C. Lutz, Complexity of Terminological Reasoning Revisited, Springer-Verlag, 1999, pp. 181–200.
[33] S. Tobies, Complexity results and practical algorithms for logics in knowledge representation (Ph.D. thesis), LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.