Towards ABox Modularization of semi-expressive Description Logics

Sebastian Wandelt [a,*] and Ralf Möller [b]

[a] *Humboldt-Universität zu Berlin, WBI, Rudower Chaussee 25, 12489 Berlin, Germany,*
E-mail: wandelt@informatik.hu-berlin.de
[b] *Hamburg University of Technology, Institute for Software Systems, Schwarzenbergstraße 95, 21073 Hamburg, Germany,*
E-mail: moeller@tuhh.de

In the last years, the vision of the Semantic Web fostered the interest in reasoning over large and very large sets of assertional statements in knowledge bases. Traditional tableau-based reasoning systems perform bad answering queries over large data sets, because these reasoning systems are based on efficient use of main memory data structures. Increasing expressivity and worst-case complexity further tighten the memory burden. The purpose of our work is to investigate how to release the main memory burden from tableau-based reasoning systems and perform efficient instance checking over $\mathcal{SHI}$-knowledge bases.

The key idea is to reduce instance checking for an individual in a knowledge base to smaller subsets of relevant axioms. Modularization techniques are introduced and further refined in order to increase the granularity of modules.

For evaluation purposes, experiments on benchmark and real world knowledge bases are carried out. The principal conclusion is that the main memory burden for instance checking can be released from tableau-based reasoning systems for semi-expressive Description Logics, by using modularization techniques.

Keywords: Description logics, modularization, islands, reasoning

1. Introduction

The Semantic Web is intended to bring structure to the meaningful content of web pages and to create an accessible environment for software agents. There is an increased interest in the development of Semantic Web applications, e.g. digital libraries [KKS09, GFW08], community management [BM07, MP06], and health-care systems [DS05, CdK08]. As the Semantic Web evolves, the amount of data available in these knowledge bases and related formats is growing with an incredible speed. Since the size of the Semantic Web is expected to further grow in the coming years, scalability and performance of Semantic Web systems become increasingly important. Usually, such systems deal with information described in Description Logic-based ontology languages such as OWL [HKP$^+$09], and provide services for storing, querying, and updating large numbers of facts.

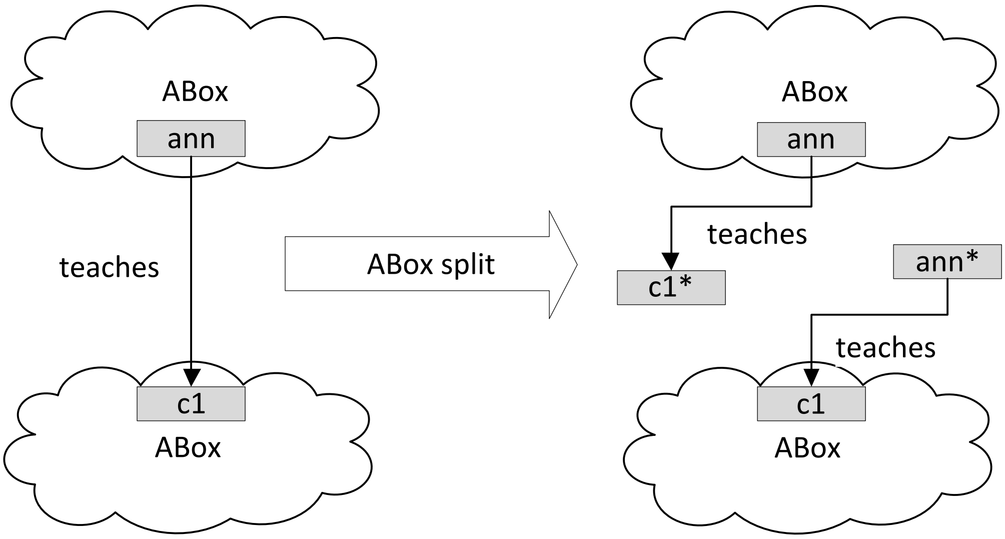
Decidability results for many expressive Description Logics and for query answering over these Description Logics have been shown, e.g., for $\mathcal{SHIQ}$ in [GHLS07], $\mathcal{SHOQ}$ in [GHS08], and $\mathcal{ALCHIOQ}b$ in [GR09]. However, existing tableau-based Description Logic reasoning systems, e.g. Racer [HMW04], Pellet [SPC$^+$07], and FaCT++ [FS06], do not perform well with large knowledge bases, since the implementation of tableau algorithms is usually based on efficient *main memory* data structures. As long as a tableau representation for an ontology fits into main memory, these systems are quite successfully used in practice.

However, if the tableau representation does not fit into main memory, these systems are doomed to fail because of out of memory errors or extensive paging activities of the operating system. Until now, to the best of our knowledge, there is no successful implementation of tableau algorithms for expressive Description Logics directly over external memory as, e.g. relational database systems. To sum up, many

[*] Corresponding author: Sebastian Wandelt: Phone: +49 (+30) 2093 3912, Fax: +49 (+30) 2093 3045

1570-5838/12/$17.00 © 2012 – IOS Press and the authors. All rights reserved

Fig. 1. Intuition of an ABox split



traditional reasoning algorithms raise serious scalability concerns, because these systems do support on secondary storage and appropriate indexing techniques.

The main goal of our research is to investigate optimizations and heuristics for instance checking with tableau-based reasoning systems. In detail, we focus on a class of Description Logics which we call semi-expressive. These semi-expressive Description Logics lie between tractable Description Logics, such as $\mathcal{EL}^{++}$ or *DL-LITE*, and inherently intractable logics, such as $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. Our focus is on the Description Logic $\mathcal{SHI}$. For more expressive Description Logics, especially including cardinality restrictions or nominals, efficient modularization becomes immediately harder.

We would like to release the main memory burden from Description Logic reasoning systems for semi-expressive ontologies. It should be possible to perform instance checks on large knowledge bases efficiently in the average case.

Inspired by graph partitioning approaches, in Section 3, we introduce techniques to break down an ABox into smaller chunks (modules), such that instance checking/retrieval can be solved by considering these smaller parts only. We formally define these *ABox modularizations* and present an initial ABox modularization algorithm.

While the initial modularization technique is quite naive, since it is basically inspired by graph components, it forms the basis of further ABox modularization techniques. We extend the naive modularizations by introducing so-called *ABox splits*. Informally speaking, an ABox split breaks up a role assertion in an ABox, while preserving the semantics (this is formalized below). The idea is depicted in Fig. 1. The clouds in Fig. 1 indicate a set of ABox assertions. We split up the role assertion $teaches(ann, c1)$, create two new individuals (ann^* and $c1^*$), and keep the concept assertions for each fresh individual copy. After applying all possible ABox splits to an ABox of a knowledge base, a graph-based ABox modularization becomes more fine-grained, i.e. one obtains more (and smaller) modules.

In order to decide whether role assertions can be broken up (split), we take into account the terminological part of the knowledge base. We extend this modularization technique step-wise from $\mathcal{ALC}$ to the semi-expressive Description Logic $\mathcal{SHI}$.

In Section 4, we show how to use ABox modularizations to solve the basic decision problem of instance checking over ontologies. We evaluate our modularization techniques with respect to benchmark and real world ontologies in Section 5. We conclude our work with directions for future work in Section 6.

Please note that in the following we will use the term *ontology* often as a synonym for the word *knowledge base*. Although in many research communities the term *ontology* only refers to the terminological part of a kowledge base, we usually mean the whole set of axioms (including the assertional part).

There exist several proposals in the research community for optimized reasoning over Description Logics. These results can be summarized as follows: There exists a lot of research to identify tractable Description Logics. For example the descriptions logic $\mathcal{EL}$ and extensions up to $\mathcal{EL}^{++}$, introduced in [BBL08], admit sound and complete reasoning in polynomial time for classification and instance checking. Another lightweight Description Logic (family) is *DL-LITE*. For an extensive overview see [ACKZ09]. *DL-LITE* enables the use of relational database management systems for query answering. Another tractable fragment is the rule-based language OWL-R, introduced in [HKP$^+$09]. All tractable fragments have in common that the set of constructors in the ontology language is restricted in order to obtain efficient reasoning algorithms. However, in practical applications, users often need more expressive languages.

Another approach to overcome the problem of reasoning over large ontologies is to approximate the ontology by a more compact representation or in a weaker Description Logic. In [PTZ09], the authors propose to reuse the idea of knowledge compilation to approximate ontologies in a weaker ontology language. For the ontology language of their choice, i.e. *DL-LITE*, efficient algorithms with polynomial complexity are known. Reasoning on the approximated ontology allows to include/reject potential answers with respect to the original ontology. A similar direction was taken in [RPZ10], where the terminology part of an ontology is approximated to the Description Logic $\mathcal{EL}^{++}$. The results from the approximated ontology are used for more efficient classification over the original ontology. The classification results can then be used for more efficient retrieval as well.

Another approach is presented in [TRKH08]. The algorithms in [TRKH08] are based on KAON2 [Mot08] algorithms, which transform the terminological part of an ontology into Datalog [MW88]. Depending on the transformation strategy, the obtained Datalog program can be used for sound or complete reasoning over instances in the source ontology. The preceding approximation approaches rely on expressivity reduction of the ontology language.

A different approach is proposed in [FKM$^+$06], [DFK$^+$07], and [DFK$^+$09], based on summarization and refinement. First, a summarization of the assertional part is created by aggregating individuals. This is part of a setup step that can be performed offline, i.e. before query answering takes place. During the summarization process, one has to take care of inconsistencies. Queries are then executed over the summarization. If the summarization leads to inconsistencies, because the individuals are not equivalent with respect to the input query, then a refinement step is executed. During the refinement step, previously merged individuals are broken up stepwise, until the result is consistent.

While approximation techniques usually rely, informally speaking, on reduction of the input or expressivity, there exist modularization techniques which try to extract independent modules with respect to a given reasoning problem. Most of the modularization techniques focus on TBox modularization. In [CPSK06], the notion of a module for the terminological part of an ontology is introduced and an algorithm for computing modules is presented. Initial research results for ABox partitioning have been shown in [GH06]. However, their presentation leaves many question open, since the authors have implemented several non-published optimizations, which contribute to their evaluation (but are not formally presented anywhere).

Usually, modularization of terminologies has not only the intention to extract modules, but to also combine modules from different source ontologies into one importing ontology. This is in detail discussed in [BS03], where so-called distributed Description Logics are proposed. The idea is to create rules between parts of terminologies, so-called bridge rules, to propagate information between source ontologies.

2. Preliminaries

2.1. Description Logics

2.1.1. Description Logics

Description logics are a family of languages for knowledge representation. Historically, Description Logics are descendants of semantic nets [Qui68] and frame systems [Min74]. In Artificial Intelligence, Description Logics are used for formal reasoning about application domains. The most prominent application

of Description Logics might be the use as a formalism for the Semantic Web and ontologies [BHS05]. For further information on the historical background of Description Logics, we refer to [BCM$^+$07]. A general review on logic-based knowledge representation with Description Logics and other logics as well, such as modal logics, is given in [Baa99].
In the following, we recapitulate syntax and semantics of the Description Logic $\mathcal{SHI}$. We assume a number of disjoint non-empty*base sets* as follows: $\mathbf{CN}$ is a set of *concept names*, $\mathbf{RN}$ is a set of *role names*, $\mathbf{NIN}$ is a set of *named individuals*, and $\mathbf{AIN}$ is a set of *anonymous individuals*. Anonymous individuals can only be used later by a tableau algorithm (existential rule), but not directly in an ABox. The expression R is a *role description* if and only if

- $R = S$ and $S \in \mathbf{RN}$ (R is called a *role name*) or
- $R = R_2^-$ and R_2 is a role description (R is called an *inverse role* of R_2). If R_2 is a role name, then R is called an *inverse role name*. The set of all role descriptions is denoted with $\mathbf{Rol}$. A role description R is called an *atomic role* if R is a role name or R is a inverse role name.

The set of all role descriptions is denoted with $\mathbf{Rol}$. A role description R is called an *atomic role* if R is a role name or R is a inverse role name.
The set of *individuals* is $\mathbf{IN} = \mathbf{NIN} \cup \mathbf{AIN}$. The set of $\mathcal{SHI}$ *-concept descriptions* is given by the following grammar:

$$C_1, C_2 ::= \top | \bot | A | \neg C_1 | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \forall R.C_1 | \exists R.C_1$$

where $A \in \mathbf{CN}$ and $R \in \mathbf{Rol}$. With $\mathbf{AtCon}$ we denote all *atomic concepts*, i.e. concept descriptions which are concept names or negated concept names. For the semantics of concept descriptions please refer to [BCM$^+$07].
A *TBox* $\mathcal{T}$ is a set of so-called *generalized concept inclusion axioms* $C_1 \sqsubseteq C_2$. A *RBox* $\mathcal{R}$ is a set of so-called *role inclusion axioms* $R_1 \sqsubseteq R_2$ and *role transitivity axioms* $Trans(R)$. An *ABox* $\mathcal{A}$ is a set of so-called *concept and role assertion axioms* $C(a)$ and $R(a_1, a_2)$. An *ontology* $\mathcal{O}$ consists of a 3-tuple $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$. We restrict the concept assertion axioms in $\mathcal{A}$ in such a way that each concept description is an atomic concept or a negated atomic concept. This is without loss of generality, since each non-atomic concept description can be given a name in the TBox. The set of TBoxes (RBoxes, ABoxes, ontologies) is denoted with $\mathbf{ST}$ ($\mathbf{SR}$, $\mathbf{SA}$, $\mathbf{SO}$).
Since we refer to less expressive Description Logics below, we summarize them as follows: The Description Logic $\mathcal{ALCHI}$ is $\mathcal{SHI}$ without transitive roles, the Description Logic $\mathcal{ALCH}$ is $\mathcal{ALCHI}$ without inverse roles, and the Description Logic $\mathcal{ALC}$ is $\mathcal{ALCH}$ without role subsumptions.
We denote with $clos(C)$ the closure of a concept description C, i.e. the set of all subconcepts. We assume that a concept description C is usually in negation normal form, i.e. for all $\neg C_1 \in clos(C)$, C_1 is a concept name. Using De Morgan laws, every concept description can be transformed into a concept description in negation normal form. The *negation normal form of a* concept description C is denoted $nnf(C)$. Given a TBox $\mathcal{T}$, the *concept closure of* $\mathcal{T}$, denoted $clos(\mathcal{T})$, is defined as $clos(\mathcal{T}) = \bigcup_{C_1 \sqsubseteq C_2 \in \mathcal{T}} (clos(\neg C_1) \cup clos(C_2))$.
Given an *ABox* $\mathcal{A}$, the set of *ABox individuals* in $\mathcal{A}$ is denoted with$Ind(\mathcal{A})$. We denote the set of *named ABox individuals* in $\mathcal{A}$ with $NInd(\mathcal{A})$. The set of *anonymous ABox individuals* in $\mathcal{A}$ is denoted with $AInd(\mathcal{A})$.

2.1.2. Decision Problems for Ontologies

Our notation for general decision problems is as follows (with the usual semantics via interpretations):

- Subsumption of concept descriptions: $\mathcal{O} \vDash C_1 \sqsubseteq C_2$
- Subsumption of role description: $\mathcal{O} \vDash R_1 \sqsubseteq R_2$
- Transitivity of role descriptions: $\mathcal{O} \vDash Trans(R_1)$

An ontology $\mathcal{O}$ is *consistent* if and only if there exists an interpretation $\mathcal{I}$, such that we have $\mathcal{I} \vDash \mathcal{O}$. An ontology which is not consistent is called *inconsistent*. In general, we assume that an ontology is consistent. A named individual $a \in NInd(\mathcal{A})$ is an *instance* of concept description C with respect to an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, denoted $\mathcal{O} \vDash C(a)$, if and only if for all interpretations $\mathcal{I}$, we have $\mathcal{I} \vDash \mathcal{O} \implies a^{\mathcal{I}} \in C^{\mathcal{I}}$. The problem of instance checking can be easily reduced to consistency checking.

2.2. Running Example

In the following, we introduce an example ontology which is used throughout the remaining part of our work. The example ontology is situated in the university domain and inspired by the Lehigh University Benchmark, introduced in [GPH05]. Sometimes we only use subsets of the example ontology.

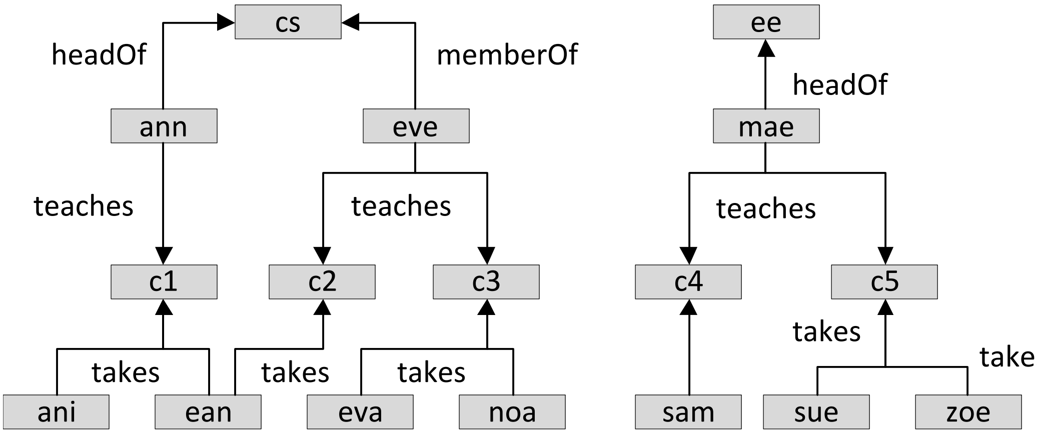
Example 1 (Running Example)**:**
The example ontology $\mathcal{O}_{Ex1} = \langle \mathcal{T}_{Ex1}, \mathcal{R}_{Ex1}, \mathcal{A}_{Ex1} \rangle$ is defined as follows

$$\begin{aligned}
\mathcal{T}_{Ex1} = \{ & \\
& Chair \equiv \exists headOf.Department, Student \equiv \exists takes.Course, \\
& GraduateStudent \equiv \exists takes.GraduateCourse, Student \sqsubseteq Person, \\
& Professor \sqsubseteq Person, UndergraduateCourse \sqsubseteq Course, GraduateCourse \sqsubseteq Course, \\
& GraduateCourse \sqcap UndergraduateCourse \sqsubseteq \bot, \top \sqsubseteq \forall teaches.Course, \\
& \top \sqsubseteq \forall takes.Course, \top \sqsubseteq \forall memberOf^{-}.Person, \top \sqsubseteq \forall isTaughtBy.Professor, \\
& \exists memberOf.\top \sqsubseteq Person, Student \sqsubseteq \exists takes.Course \\
& \}
\end{aligned}$$

$$\mathcal{R}_{Ex1} = \{headOf \sqsubseteq memberOf, teaches \equiv isTaughtBy^{-}\}$$

$$\begin{aligned}
\mathcal{A}_{Ex1} = \{ & \\
& Department(cs), Department(ee), \\
& Professor(ann), Professor(eve), Professor(mae), \\
& UndergraduateCourse(c1), UndergraduateCourse(c4), \\
& UndergraduateCourse(c5), \\
& GraduateCourse(c2), GraduateCourse(c3), \\
& Student(ani), Student(ean), Student(eva), Student(noa), \\
& Student(sam), Student(sue), Student(zoe), \\
& headOf(ann, cs), memberOf(eve, cs), headOf(mae, ee), \\
& teaches(ann, c1), teaches(eve, c2), teaches(eve, c3), \\
& teaches(mae, c4), teaches(mae, c5), \\
& takes(ani, c1), takes(ean, c1), takes(ean, c2), takes(eva, c3), \\
& takes(noa, c3), takes(sam, c4), takes(sue, c5), takes(zoe, c5) \\
& \}.
\end{aligned}$$

Fig. 2. Individual relationships for Example 1



In $\mathcal{T}_{Ex1}$, we define, for instance, the concept description $Chair$ as someone who is the head of a $Department$. Since the definition is sufficient, we use an concept equivalence axiom. In a similar style, we define the concept descriptions $Student$ and $GraduateStudent$. We introduce a $GraduateCourse$ and an $UndergraduateCourse$, and enforce that both concept descriptions are disjoint. In addition, we define domain and range restrictions on roles descriptions used in $\mathcal{O}_{Ex1}$.

We define two role subsumptions in $\mathcal{R}_{Ex1}$. The role description $headOf$ is subsumed by role description $memberOf$. Furthermore, we state that the role description $teaches$ is the inverse role of the role description $isTaughtBy$.

The relationships between individuals in $\mathcal{A}_{Ex1}$ are depicted in Fig. 2. Please note that only role assertions are shown in the graph, since we only intend to emphasize the relationship between the individuals.

It is easy to see that individual ann is an instance of concept description $Chair$ with respect to the ontology $\mathcal{O}_{Ex1}$. In order to prove the entailment of the concept assertion $Chair(ann)$, not the whole ABox is necessary. The two ABox assertions $headOf(ann, cs)$ and $Department\ (cs)$ already suffice to derive the fact that ann is a $Chair$. This small example already suggests that modularization techniques can be valuable in reasoning over ontologies. We define different kinds of modularization techniques below.

3. Modularization

Reasoning over Description Logic ontologies, such as the task of instance retrieval, is difficult. The worst-case time complexity even for solving the basic decision problem of instance checking is known to be double-exponential for $\mathcal{SHI}$. Furthermore, the sheer amount of data, supported by today's advanced storage and dissemination technologies, and the users willingness to use these technologies, makes achieving efficiency in information retrieval increasingly difficult.

We think that modularization techniques can be used in order to release the main memory burden from reasoning systems and to speed up instance checking/retrieval. Recent advances in distributed and parallel computing, such as multicore-systems and Cloud computing, give further support, since it might be possible to distribute modules over multiple cores or computers. We focus on the modularization of ABoxes here, since the size of the assertional part often exceeds the size of the terminological part by orders of magnitude, especially in database-motivated scenarios.

During the remaining part of our work we make two assumptions with respect to the input ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$:

- We assume that $\mathcal{O}$ is initially consistent. Despite recent research trends on reasoning over inconsistent (web-)ontologies, e.g. see [HVHT05] and [HH08], we focus on standard decision problems.

- We assume that all concept assertions in $\mathcal{A}$ (and in instance checking/retrieval queries) only contain atomic concept descriptions.

In this chapter, we introduce techniques to break down an ABox into smaller chunks (modules), such that decision problems can be solved by considering the smaller parts only. In Subsection 3.1, we formally define ABox modularizations and technical preliminaries. In Subsection 3.2, we present an initial ABox modularization algorithm. While the initial partitioning technique is quite naive, since it is basically inspired by graph components, the technique builds the basis of further modularization techniques. We extend the naive partitioning for the Description Logic $\mathcal{ALC}$ in Subsection 3.3, by taking into account terminological information. This extension usually offers a more fine grained partitioning/modularization. We further extend the technique to the semi-expressive Description Logic $\mathcal{SHI}$.

3.1. Modularization Preliminaries

3.1.1. ABox Modularizations

We define the (very general) notion of an ABox modularization in Definition 1. While our criterion for ABox modularizations seems quite lax, we would like to keep the definition of ABox modularizations as open as possible. For instance, we will define modularization techniques, such that the modules are not necessarily subsets of the original ABox. The intuition for these kinds of modules will become clear below. Please note that whenever we use the term *modularization* we usually refer to the result of the modularization process.

Definition 1 (ABox Modularization)**:**
An *ABox modularization* M is defined as a set of ABoxes $\{\mathcal{A}_1, ..., \mathcal{A}_n\}$. Each $\mathcal{A}_i$ is called an *ABox module*. Given a TBox $\mathcal{T}$, a RBox $\mathcal{R}$, and an *ABox modularization* M, we say that M *entails a concept assertion* $C(a)$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$, denoted $\langle \mathcal{T}, \mathcal{R}, M\rangle \vDash C(a)$, if $\exists \mathcal{A}_i \in M. \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i\rangle \vDash C(a)$. We say that M *entails a role assertion* $R(a_1, a_2)$ w.r.t. $\mathcal{T}$ and $\mathcal{R}$, denoted $\langle \mathcal{T}, \mathcal{R}, M\rangle \vDash R(a_1, a_2)$, if $\exists \mathcal{A}_i \in M. \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i\rangle \vDash R(a_1, a_2)$.
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$ and an ABox modularization $M = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$, we say that M is *sound for instance retrieval in ontology* $\mathcal{O}$ if for all atomic concept descriptions $C \in \mathbf{AtCon}$ and all individuals $a \in NInd(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}, M\rangle \vDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle \vDash C(a)$. The ABox modularization M is *complete for instance retrieval in ontology* $\mathcal{O}$ if for all atomic concept descriptions $C \in \mathbf{AtCon}$ and all individuals $a \in NInd(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle \vDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, M\rangle \vDash C(a)$.

Informally speaking, we have chosen to base soundness and completeness of modularizations on entailment of atomic concept descriptions for all named individuals. This assumption makes the definition and implementation of our techniques easier. However, please note that the restriction to atomic query concepts is without losing generality, since we can assign fresh concept names to non-atomic query concepts in the TBox and execute a query for the (atomic) concept names. Again, remember that we would like to obtain modularizations which preserve entailment of atomic concept assertions for all named individuals in the input ontology.
We present examples for further explanation in Example 3 and Example 4. First, we introduce one example ontology in Example 2.

Example 2 (Example Ontology for ABox Modularization)**:**
The ontology $\mathcal{O}_{Ex2} = \langle \mathcal{T}_{Ex2}, \mathcal{R}_{Ex2}, \mathcal{A}_{Ex2}\rangle$ is defined as follows

$$\mathcal{T}_{Ex2} = \{Chair \equiv \exists headOf.Department\}$$

$$\mathcal{R}_{Ex2} = \{headOf \sqsubseteq memberOf\}$$

$$\begin{aligned}
\mathcal{A}_{Ex2} = \{ & \\
& Department(ee), Professor(mae), UndergraduateCourse(c4), \\
& UndergraduateCourse(c5), Student(sam), Student(sue), Student(zoe), \\
& headOf(mae, ee), teaches(mae, c4), teaches(mae, c5), \\
& takes(sam, c4), takes(sue, c5), takes(zoe, c5) \\
& \}.
\end{aligned}$$

Example 3 (First Example for an ABox Modularization)**:**
One possible ABox modularization for ontology $\mathcal{O}_{Ex2}$ is $M_{Ex3} = \{\mathcal{A}_{Ex3,1}, \mathcal{A}_{Ex3,2}\}$, such that

$$\begin{aligned}
\mathcal{A}_{Ex3,1} &= \{Department(ee), headOf(mae, ee), Professor(mae)\} \\
\mathcal{A}_{Ex3,2} &= \{ \\
& \quad UndergraduateCourse(c4), UndergraduateCourse(c5), \\
& \quad Student(sam), Student(sue), Student(zoe), \\
& \quad teaches(mae, c4), teaches(mae, c5), \\
& \quad takes(sam, c4), takes(sue, c5), takes(zoe, c5) \\
& \quad \}.
\end{aligned}$$

It is easy to see that with respect to the original ontology $\mathcal{O}_{Ex2}$, we have that mae is an instance of the concept description $Chair$, since she has a $headOf$-relationship to a $Department$ called ee. The ABox modularization in Example 3 entails that individual mae is an instance of the concept description $Chair$, since all necessary axioms are being kept in one ABox module. Moreover, it can be shown that the modularization in Example 3 is sound and complete for reasoning over ontology $\mathcal{O}_{Ex2}$.
Another example modularization is given in Example 4.

Example 4 (Second Example for an ABox Modularization)**:**
Another possible ABox modularization for Ontology $\mathcal{O}_{Ex2}$ is $M_{Ex4} = \{\mathcal{A}_{Ex4,1}, \mathcal{A}_{Ex4,2}\}$, such that

$$\begin{aligned}
\mathcal{A}_{Ex4,1} &= \{Department(ee), UndergraduateCourse(c4), UndergraduateCourse(c5)\} \\
\mathcal{A}_{Ex4,2} &= \{ \\
& \quad Professor(mae), Student(sam), Student(sue), Student(zoe), \\
& \quad headOf(mae, ee), teaches(mae, c4), teaches(mae, c5), \\
& \quad takes(sam, c4), takes(sue, c5), takes(zoe, c5) \\
& \quad \}.
\end{aligned}$$

The ABox modularization in Example 4 is chosen quite arbitrarily and it can be seen that neither of the two modules entails that mae is an instance of the concept description $Chair$. This happens because the necessary information for entailment was split up into different ABoxes. In [BS03], this problem is solved by so called *bridge rules*, which communicate useful temporary reasoning results from one module to another module. However, we would like to keep relevant information together, in order to avoid the communication overhead.
The ABox modularizations from Example 3 and 4 show that the choice of ABox modularization is critical for use and quality during reasoning. In the remaining part of the paper, we discuss in detail how to obtain sound and complete ABox modularizations (one might even have to add new - yet redundant - assertions).

3.1.2. Reasoning Procedures

The purpose of a tableau algorithm is to check consistency of a given ontology $\mathcal{O}$. As pointed out before, (in-)consistency is one of the basic decision problems. Many other decision problems can be reduced to inconsistency checking. Given an input ontology $\mathcal{O}$, a tableau algorithm tries to generate a finite representation for a model of $\mathcal{O}$. If the algorithm succeeds, the algorithm returns a compact model representation and shows that the ontology is consistent. If the algorithm fails, then the output is false, i.e. there cannot exists a model for $\mathcal{O}$. In each step, a tableau algorithm applies one tableau rule to an intermediate ontology. For details and formal definition of a tableau algorithm for the Description Logic $\mathcal{SHI}$ please refer to [HS99].

There are different representations used as a basis for tableau algorithms, e.g. graph-based and ABox-based. Here, we focus on ABox-based views of tableau algorithms as for instance chosen in [BS01]. Each path in a tableau (as a tree) is called a *tableau run*. It is easy to see that a tableau proof is successful, if there exists at least one tableau run. Furthermore, please note that the composition of successful tableau runs of two individual disjoint ontologies $\mathcal{O}_1 = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_1 \rangle$ and $\mathcal{O}_2 = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_2 \rangle$ can be composed into a successful tableau run for ontology $\mathcal{O}_3 = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_1 \cup \mathcal{A}_2 \rangle$.

We use well known notions on tableau proofs and trees, which are derived from either structure, e.g. inner nodes, leaves, and root nodes. Furthermore, we call a tableau run *satisfying* if it does not contain a clash (directly contradicting assertions in the leaf ABox of the tableau run). An ABox in a tableau run is *complete* if no tableau rule is applicable.

In the remaining part of this section, we focus on how to find 'interesting' ABox modularizations, i.e. ABox modularizations which guarantee soundness and completeness for different classes of Description Logics.

3.2. Component-based Modularization

With *component-based modularization* we refer to modularization techniques which only consider the assertional part of an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ in order to decide how to break up an ABox into an ABox modularization. For this purpose, we look at ABoxes as graphs. The intuition is as follows: each individual in the ABox is mapped to a node in the graph. Node labels are concept assertions from the ABox and the edges of the graph are derived from the role assertions. We introduce a formal notion in order to define algorithms and proofs. Given an ABox $\mathcal{A}$, we define the corresponding ABox-graph in Definition 2.

Definition 2:

Given an ABox $\mathcal{A}$, the *ABox-graph* $\mathbb{G}^{\mathcal{A}} = \langle \mathbf{N}, \mathbf{E}, \phi, \sigma \rangle$ for $\mathcal{A}$ is a directed labeled graph such that

- $\mathbf{N} = Ind(\mathcal{A})$,
- $edges = Ind(\mathcal{A}) \times Ind(\mathcal{A})$,
- the domain of ϕ and σ are $\mathbf{N}$ and $\mathbf{E}$, respectively,
- the codomain of ϕ is $\wp(\mathbf{Con})$,
- the codomain of σ is $\wp(\mathbf{Rol})$,
- for all $n \in \mathbf{N}$, we have $C \in \phi(n)$ if and only if $C(n) \in \mathcal{A}$, and
- for all pairs of nodes $(n_1, n_2) \in \mathbf{N} \times \mathbf{N}$, we have $R \in \sigma(n_1, n_2)$ if and only if $R(n_1, n_2) \in \mathcal{A}$.

Please note that the construction of the ABox-graph for a given ABox $\mathcal{A}$ is deterministic and there is an obvious one-to-one correspondence between ABoxes and their graphs. This means that given a ABox-graph $\mathbb{G}^{\mathcal{A}}$, we can reconstruct the corresponding ABox $\mathcal{A}$. Given this relationship, we often change between the usual ABox-view and the ABox-graph-view whenever it is convenient.

Since the ABox $\mathcal{A}$ of an ontology $\mathcal{O}$ can be seen as a graph, it seems natural to apply standard connectedness-based graph partitioning techniques to determine ABox modules: if two individuals a_1 and a_2 are connected in the ABox-graph, then these two individuals end up in the same ABox module.

Definition 3 (Graph Component-based ABox Modularization for an ABox)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, a *graph component-based ABox modularization for* $\mathcal{A}$, denoted $MC^{\mathcal{A}}$, is an ABox Modularization $MC^{\mathcal{A}} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ for $\mathcal{A}$, such that $\mathcal{A}_i \in MC^{\mathcal{A}} \iff \mathbb{G}^{\mathcal{A}_i}$ is a component in $\mathbb{G}^{\mathcal{A}}$.

The components of a graph can be obtained in linear time[HT73]. In the following, we often refer to the term graph component-based with the term *component-based.*
Please note that the graph component-based ABox modularization for an ABox $\mathcal{A}$ is unique. An example of a component-based ABox modularization is shown in Example 5.

Example 5 (Example for ABox Modularization by Graph Components)**:**
Given the ontology $\mathcal{O}_{Ex5} = \langle \mathcal{T}_{Ex5}, \mathcal{R}_{Ex5}, \mathcal{A}_{Ex5} \rangle$, such that

$$\begin{aligned}
\mathcal{T}_{Ex5} &= \{Chair \equiv \exists headOf.Department\} \\
\mathcal{R}_{Ex5} &= \{headOf \sqsubseteq memberOf\} \\
\mathcal{A}_{Ex5} &= \{Department(cs), Professor(ann), headOf(ann, cs), \\
&\qquad Department(ee), Professor(mae), headOf(mae, ee)\},
\end{aligned}$$

the graph component-based ABox modularization is $M_{Ex5} = \{\mathcal{A}_{Ex5,1}, \mathcal{A}_{Ex5,2}\}$, such that

$$\begin{aligned}
\mathcal{A}_{Ex5,1} =& \{Department(cs), Professor(ann), headOf(ann, cs)\} \\
\mathcal{A}_{Ex5,2} =& \{Department(ee), Professor(mae), headOf(mae, ee)\}.
\end{aligned}$$

It is easy to see that the component-based ABox modularization $MC^{\mathcal{A}} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ for $\mathcal{A}$ is sound for instance retrieval in $\mathcal{O}$ (by monotonicity).
Next, we show a proof for completeness with respect to $\mathcal{SHI}$ and give a negative result for the Description Logic $\mathcal{SHOQ}$.

Lemma 1 ($\mathcal{SHI}$-Instance Retrieval over Component-based ABox Modularizations is Complete)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and a component-based ABox modularization $MC^{\mathcal{A}} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$ for $\mathcal{A}$, the ABox modularization $MC^{\mathcal{A}}$ is complete for instance retrieval in $\mathcal{O}$.

Proof of Lemma 1. We have to show that for all atomic concept descriptions $C \in \mathbf{AtCon}$ and all individuals $a \in NInd(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, MC^{\mathcal{A}} \rangle \vDash C(a)$. By contraposition: We have to show $\langle \mathcal{T}, \mathcal{R}, MC^{\mathcal{A}} \rangle \nvDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash C(a)$. Assume that $\langle \mathcal{T}, \mathcal{R}, MC^{\mathcal{A}} \rangle \nvDash C(a)$. Thus, for all $\mathcal{A}_i \in MC^{\mathcal{A}}$, $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \cup \{\neg C(a)\} \rangle$ is consistent. Let $\mathcal{A}_j$ be the ABox module, such that $a \in \mathcal{A}_j$. There exists only one such module, by Definition 3. We can conclude that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_1 \cup \mathcal{A}_2 \cup ... \mathcal{A}_{j-1} \cup (\mathcal{A}_j \cup \{\neg C(a)\}) \cup \mathcal{A}_{j+1} \cup ... \cup \mathcal{A}_n \rangle$ is consistent as well. Since $\mathcal{A} \cup \{\neg C(a)\} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup ... \mathcal{A}_{j-1} \cup (\mathcal{A}_j \cup \{\neg C(a)\}) \cup \mathcal{A}_{j+1} \cup ... \cup \mathcal{A}_n$, the ontology $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a)\} \rangle$ is consistent, and thus $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash C(a)$. □

Theorem 1 (Instance Retrieval over Component-based ABox Modularizations is Sound and Complete for $\mathcal{SHI}$)**:**
Instance Retrieval over component-based ABox modularizations is sound and complete for $\mathcal{SHI}$-ontologies.

Proof of Theorem 1. Soundness is immediate (subsets of the original ABox) and in Lemma 1 we show completeness. □

Actually Theorem 1 is true for any standard Description Logic without nominals. Unfortunately, the completeness result does not hold for ontologies containing nominals, e.g. $\mathcal{SHOQ}$-ontologies. It is not always possible to ensure individual disjointness of tableau runs, because TBox axioms can introduce individuals and these individuals cannot be renamed without changing the semantics and result of the tableau run. While all modules of an ABox modularization are consistent, the complete ontology might be inconsistent. Thus, graph component-based ABox modularization techniques cannot be applied directly to ontologies containing nominals.

The effectiveness of component-based modularization techniques is usually quite low (e.g. only one big module is obtained), since in most ontologies each individual is related to many other individuals, either directly or indirectly.

3.3. Intensional-based Modularization

Component-based modularization alone can be too naive for the modularization of real world ontologies. Usually, most individuals in an ABox are connected by paths of role assertions to many other individuals. Thus, the number of modules obtained by component-based ABox modularizations can be quite small and the average module size is usually quite big. In the following section, we discuss how to compute smaller modules by splitting up role assertions whenever possible. After the splitting process is finished, we can apply component-based modularization techniques on the result. Please note again that, during the modularization process, we are interested in preserving entailment of atomic concept descriptions for each named individual.

The idea is to analyze the terminological part of the ontology (hence called intensional-based modularization) to find out in which ways role assertions are used in the ontology. It is important to note that we only use a purely syntactical analysis of the TBox. Otherwise, for complex ontologies, a more sophisticated analysis could turn out to be too complex. In order to illustrate the idea of intensional-based modularization in a more detailed way, an example ontology is given in Example 6.

Example 6 (Example Ontology)**:**
Let $\mathcal{O}_{Ex6} = \langle \mathcal{T}_{Ex6}, \mathcal{R}_{Ex6}, \mathcal{A}_{Ex6} \rangle$ be as follows:

$$\begin{aligned}
\mathcal{T}_{Ex6} =& \{\top \sqsubseteq \forall takes.Course\} \\
\mathcal{R}_{Ex6} =& \{\} \\
\mathcal{A}_{Ex6} =& \{Course(c5), Student(zoe), \\
& \quad takes(zoe, c5), teaches(mae, c5)\}.
\end{aligned}$$

Looking closer at the ontology defined in Example 6 reveals the following details about the role assertions in $\mathcal{A}_{Ex6}$:

- $teaches(mae, c5)$: The role $teaches$ is not used (mentioned) anywhere in the TBox or RBox of the ontology $\mathcal{O}_{Ex6}$. Thus, no information can be propagated in a tableau algorithm from mae to $c5$ and vice versa, and it might be safe to ignore/remove the role assertion to obtain more fine grained ABox modularization in some cases.
- $takes(zoe, c5)$: Although the role $takes$ is mentioned in $\mathcal{T}_{Ex6}$, we can see that it is only used to propagate the concept description $Course$. Since individual $c5$ is already known to be an instance of $Course$, because that fact is directly asserted in $\mathcal{A}_{Ex6}$, we might further split up this role assertion in some cases.

3.3.1. Technical Preliminaries

In the following, we define necessary criteria for identifying concept descriptions which are propagated over role descriptions in the worst-case during the application of a tableau algorithm. Since we only allow atomic concept assertions in ABoxes, we can focus on the syntactical analysis of the TBox to obtain this set of concept descriptions. First, we revisit normal forms of general concept inclusions and TBoxes. A general concept inclusion axiom is in *normal form* if it has the shape $\top \sqsubseteq C$, such that C is a concept description in negation normal form. Please note that every concept description can be transformed into an equivalent concept description in negation normal form. A TBox $\mathcal{T}$ is in *normal form* (or *normalized*) if all general concept inclusion axioms in $\mathcal{T}$ are in normal form.

In Definition 4, we formally define a structure which associates the worst-case set of propagated concept descriptions with each role description. The idea is to extract subconcept descriptions of all $\forall$-concept descriptions from the closure of the input TBox.

Definition 4 ($\forall$-info structure)**:**

A $\forall$-*info structure for* a TBox $\mathcal{T}$ in normal form is a function $info^{\forall}_{\mathcal{T}}$: $\mathbf{Rol} \rightarrow \wp(\mathbf{Con})$, such that we have $C \in info^{\forall}_{\mathcal{T}}(R)$ if and only if $\forall R.C \in clos(\mathcal{T})$.

Example 7 (Example for a $\forall$-info structure)**:**

Let

$$\mathcal{T}_{Ex7} = \{\top \sqsubseteq \forall takes.Course, \exists takes.Course \sqsubseteq Student, \exists memberOf.\top \sqsubseteq Person,$$
$$GraduateStudent \sqsubseteq Student, UndergraduateStudent \sqsubseteq Student\},$$

then one TBox in normal form is

$$\mathcal{T}_{Ex7norm} = \{\top \sqsubseteq \forall takes.Course, \top \sqsubseteq \forall takes.\neg Course \sqcup Student, \top \sqsubseteq \forall memberOf.\bot \sqcup Person,$$
$$\top \sqsubseteq \neg GraduateStudent \sqcup Student, \top \sqsubseteq \neg UndergraduateStudent \sqcup Student\}$$

and the $\forall$-info structure for $\mathcal{T}_{Ex7norm}$ is:

$$info^{\forall}_{\mathcal{T}}(R) = \begin{cases} \{Course, \neg Course\} & \text{if } R = takes, \\ \{\bot\} & \text{if } R = memberOf, \\ \emptyset & \text{otherwise.} \end{cases}$$

The $\forall$-info structure helps us to check, which concept descriptions are (in the worst case) propagated over role assertions during application of tableau rules in tableau proofs. First, we prove a general property of concept descriptions in tableau runs.

Given the above results, we define an operation which splits up role assertions in such a way that we can apply graph component-based modularization techniques over the outcome of the split (or a series of splits). Then we show that under some conditions the operation retains soundness and completeness for instance checking/retrieval.

Definition 5 (ABox Split)**:**

Given

- a role description R,
- two distinct named individuals a and b,
- two distinct anonymous individuals c and d, and,
- an ABox $\mathcal{A}$,

an *ABox split* is a function $\downarrow_{c,d}^{R(a,b)}$: $\mathbf{SA} \to \mathbf{SA}$, defined as follows:

- If $R(a,b) \in \mathcal{A}$ and $\{c,d\} \nsubseteq Ind(\mathcal{A})$, then

$$\begin{aligned}\downarrow_{c,d}^{R(a,b)}(\mathcal{A}) =& \mathcal{A} \setminus \{R(a,b)\} \cup \\ & \{R(a,d), R(c,b)\} \cup \{C(c) \mid C(a) \in \mathcal{A}\} \cup \{C(d) \mid C(b) \in \mathcal{A}\}\end{aligned}$$

- Else

$$\downarrow_{c,d}^{R(a,b)}(\mathcal{A}) = \mathcal{A}.$$

The intuition of Definition 5 is depicted in Fig. 1. We split up a role assertion and keep the concept assertions for each fresh individual copy. The reason for keeping the asserted concept descriptions is explained below. If the ABox does not contain the role assertion in question, then the split returns the unchanged ABox.

In Definition 6, we define soundness and completeness of ABox splits. While soundness of ABox splits is shown by simply applying Lemma 2, the proof of completeness is harder and depends on several criteria.

Definition 6 (Sound, Complete and Valid ABox Split)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, we say that

- $\downarrow_{c,d}^{R(a,b)}$ is *sound with respect to* $\mathcal{O}$ if for all individuals $a_1 \in NInd(\mathcal{A})$ and all atomic concept descriptions $C \in \mathbf{AtCon}$:

$$\exists \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \vDash C(a_1) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a_1),$$

- $\downarrow_{c,d}^{R(a,b)}$ is *complete with respect to* $\mathcal{O}$ if for all individuals $a_1 \in NInd(\mathcal{A})$ and all atomic concept descriptions $C \in \mathbf{AtCon}$:

$$\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a_1) \implies \exists \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \vDash C(a_1),$$

- $\downarrow_{c,d}^{R(a,b)}$ is *valid with respect to* $\mathcal{O}$ if $\downarrow_{c,d}^{R(a,b)}$ is sound and complete with respect to $\mathcal{O}$.

Lemma 2 (Soundness of ABox Splits)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is sound with respect to $\mathcal{O}$.

Proof of Lemma 2. We have to show that for all individuals $a_1 \in NInd(\mathcal{A})$ and all atomic concept descriptions $C \in \mathbf{AtCon}$: $\exists \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \vDash C(a_1) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a_1)$. Assume that $\exists \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \vDash C(a_1)$. Without loss of generality, let $\mathcal{A}_X \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}$ be the ABox module which makes $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_X \cup \{\neg C(a_1)\} \rangle$ inconsistent. Furthermore, let $\mathcal{A}_* = \mathcal{A}_X \cup \{\neg C(a_1)\}$. It is easy to see that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \rangle$ is inconsistent and we have to show that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a_1)\} \rangle$ is inconsistent. By contraposition: We show that if $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a_1)\} \rangle$ is consistent, then $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \rangle$ is consistent. Assuming that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a_1)\} \rangle$ is consistent, there exists an interpretation $\mathcal{I}$, such that $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a_1)\} \rangle$. It is easy to see that for the interpretation $\mathcal{I}_{new}$, an extension of $\mathcal{I}$ by setting $c^{\mathcal{I}_{new}} = a^{\mathcal{I}}$ and $d^{\mathcal{I}_{new}} = b^{\mathcal{I}}$, $\mathcal{I}_{new} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \rangle$ and thus, $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \rangle$ is consistent. □

The criteria for ensuring completeness of ABox splits are introduced below and proven step-wise for the Description Logic $\mathcal{ALC}$ and extensions up to $\mathcal{SHI}$. We define a set of consistency-preserving ABox splits, for which, informally speaking, the split-up role assertion can be added to the outcome of the ABox split without changing consistency. This is formally defined in Definition 7.

Definition 7 (Consistency-preserving ABox Split)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, we say that $\downarrow_{c,d}^{R(a,b)}$ is a *consistency-preserving ABox split for* $\mathcal{O}$ if for all atomic concept descriptions C and all individuals $e \in NInd(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \rangle$ is consistent $\implies \langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \cup \{R(a,b)\} \rangle$ is consistent.

Lemma 3 (Completeness of Consistency-preserving ABox Splits)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is complete with respect to $\mathcal{O}$ if $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\mathcal{O}$.

Proof of Lemma 3. We have to show that for all named individuals $a_1 \in NInd(\mathcal{A})$ and all atomic concept descriptions $C \in \mathbf{AtCon}$:

$$\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \models C(a_1) \implies \exists \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \models C(a_1).$$

By contraposition: We have to show that $\forall \mathcal{A}_i \in MC^{\downarrow_{c,d}^{R(a,b)}(\mathcal{A})}.\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \rangle \not\models C(a_1) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \not\models C(a_1)$. Assume that all $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_i \cup \{\neg C(a_1)\} \rangle$ are consistent. Let $\mathcal{A}_j$ be the ABox module, such that $a_1 \in NInd(\mathcal{A}_j)$. There exists only one such module, by Definition 3. Let $\mathcal{A}_* = \mathcal{A}_1 \cup \mathcal{A}_2 \cup ... \mathcal{A}_{j-1} \cup (\mathcal{A}_j \cup \{\neg C(a)\}) \cup \mathcal{A}_{j+1} \cup ... \cup \mathcal{A}_n$. We know that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \rangle$ is consistent. Since $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, we know that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}_* \cup \{R(a,b)\} \rangle$ is consistent, because $\mathcal{A}_* =\downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(a_1)\}$. Since $\mathcal{A} \cup \{\neg C(a_1)\} \subseteq \mathcal{A}_* \cup \{R(a,b)\}$, we can conclude that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a_1)\} \rangle$ is consistent as well, and thus $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \not\models C(a_1)$. □

Lemma 3 and Definition 7 help us to identify complete ABox splits, by finding consistency-preserving ABox splits. We identify classes of these consistency-preserving ABox splits below. Please note that consistency-preserving ABox splits do not affect the blocking of individuals, i.e. adding the role assertions does not change the blocking condition for any individual.
We distinguish the following three scenarios as candidate criteria for consistency-preserving ABox splits $\downarrow_{c,d}^{R(a,b)}$:

1. No concept descriptions are propagated over R.
2. Only the concept description $\bot$ is propagated over R.
3. Only atomic concept descriptions are propagated over R, such that each propagation, informally speaking, either yields redundant information or an obvious clash.

Each scenario is discussed in detail for the Description Logic $\mathcal{ALC}$ below.

3.3.2. Consistency-preserving ABox Splits for $\mathcal{ALC}$

Below, we discuss three cases for consistency-preserving ABox splits. First, in Lemma 4, we prove that an ABox split is consistency-preserving, if no concept descriptions can be propagated over the role assertion of the ABox split during the application of a tableau algorithm to the ontology.

Lemma 4 (Propagationless ABox Splits)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\mathcal{O}$ if $info_{\mathcal{T}}^{\forall}(R) = \emptyset$.

Proof of Lemma 4. We have to show that for all atomic concept descriptions C and all individuals $e \in NInd(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \rangle$ is consistent $\implies \langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \cup \{R(a,b)\} \rangle$ is consistent. Assume that $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \rangle$ is consistent. Thus, there exists a satisfying tableau run RUN for $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \rangle$. It is easy to see that the new tableau run

$RUN^{+\{R(a,b)\}}$ (obtained by adding $R(a,b)$ to all the ABoxes in the original tableau run) is a tableau run for $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e) \cup \{R(a,b)\}\rangle$. The only tableau rule which could become applicable due to the role assertion addition is the $\forall$-tableau rule. But since we assume $info_{\mathcal{T}}^{\forall}(R) = \emptyset$, we can conclude that the individual a cannot be labeled with a $\forall$-constraint on role R. Thus, the $\forall$-tableau rule is not applicable either. The new role assertion yields no immediate clash. Thus, we have a satisfying tableau run and $\langle \mathcal{T}, \mathcal{R}, \downarrow_{c,d}^{R(a,b)} (\mathcal{A}) \cup \{\neg C(e)\} \cup \{R(a,b)\}\rangle$ is consistent. □

Next, we discuss consistency-preserving ABox splits with role assertions, such that only direct contradictions are propagated, i.e. given an $\downarrow_{c,d}^{R(a,b)}$, we have $info_{\mathcal{T}}^{\forall}(R) = \{\bot\}$.

Lemma 5 (Clash-Propagation ABox Splits)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\mathcal{O}$ if $info_{\mathcal{T}}^{\forall}(R) = \{\bot\}$.

Proof of Lemma 5. In the same style as the proof of Lemma 4. Please note that, if the $\forall$-tableau rule becomes applicable for $R(a,b)$, then it must have been already applicable in RUN for the role assertion $R(a,d)$. Since RUN is satisfying, d does not contain a direct clash, and thus the $\forall$-tableau rule was not applicable to $R(a,d)$ in RUN and it cannot be applicable to $R(a,b)$ either. □

In the following, we discuss completeness of ABox splits with role assertions, such that only chosen atomic concepts are propagated. These atomic concepts are special in such a way that they will either only propagate redundant information or yield a direct clash during the application of a tableau algorithm. First, we discuss the propagation of redundant information. The terminological knowledge can be used to avoid the worst-case propagation over the role assertion of concern.

Lemma 6 (Redundant Propagation ABox Splits)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\mathcal{O}$ if $info_{\mathcal{T}}^{\forall}(R) = \{C_1\}$ and there exists a concept description C_2, with $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C_1$.

Proof of Lemma 6. In the same style as the proof of Lemma 4. Please note that the only tableau rule which could become applicable due to the role assertion addition is the $\forall$-tableau rule. We have $C_1(b) \in \mathcal{A}_{leaf}$ (since $\mathcal{A}_{leaf}$ is a complete ABox and $\mathcal{T} \vDash C_2 \sqsubseteq C_1$), and thus the $\forall$-tableau rule cannot become applicable for the new role assertion $R(a,b)$ and concept description $\forall R.C_1$. □

We discuss the propagation of directly contradicting information next. If a propagation will only yield a direct clash due to disjointness information, we can break up the role assertion as well.

Lemma 7 (Redundant Contradiction-Propagation ABox Splits)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is a consistency-preserving ABox split for $\mathcal{O}$ if $info_{\mathcal{T}}^{\forall}(R) = \{C_1\}$ and there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_1 \sqcap C_2 \sqsubseteq \bot$.

Proof of Lemma 7. In the same style as the proof of Lemma 6. Since the leaf ABox $\mathcal{A}_{leaf}$ of RUN is complete, the only tableau rule which could become applicable due to the ABox extension is the $\forall$-tableau rule. However, if the $\forall$-tableau rule becomes applicable for $R(a,b)$, then it must have been already applicable in RUN for the role assertion $R(a,d)$ and we must have $C_1(d) \in \mathcal{A}_{leaf}$. This must have yielded a clash, since $\mathcal{T} \vDash C_1 \sqcap C_2 \sqsubseteq \bot$ and $C_2(d) \in \mathcal{A}_{leaf}$.
Since RUN is satisfying, d does not contain that clash, and thus the $\forall$-tableau rule was not applicable to $R(a,d)$ in RUN and it cannot be applicable to $R(a,b)$ either. Thus the $\forall$-tableau rule is not applicable. □

In Theorem 2, we summarize the above results about decision criteria for ABox splits over $\mathcal{ALC}$-ontologies.

Theorem 2 (Decision Criteria for ABox Splits in $\mathcal{ALC}$-ontologies)**:**
Given an $\mathcal{ALC}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is valid for $\mathcal{O}$ if for each $C \in \mathit{info}_{\mathcal{T}}^{\forall}(R)$

- $C = \bot$ or
- there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or
- there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$.

Proof of Theorem 2. Direct consequence of Lemma 2 (soundness), Lemma 3, Lemma 4, Lemma 5, Lemma 6 and Lemma 7. □

3.3.3. Consistency-preserving ABox Splits for $\mathcal{ALCH}$

In the following, we extend our results for valid ABox splits step-by-step from $\mathcal{ALC}$-ontologies to $\mathcal{SHI}$-ontologies. First, we add role hierarchies to $\mathcal{ALC}$.

In presence of role hierarchies, the ∀-info structure needs to be extended in order to handle role subsumptions, because propagations of concept descriptions can now occur over subsumed role descriptions.

Definition 8 (Extended ∀-info Structure)**:**
Given a TBox $\mathcal{T}$ in normal form and a RBox $\mathcal{R}$, an *extended ∀-info structure for $\mathcal{T}$ and $\mathcal{R}$* is a function $\mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall} : \mathbf{Rol} \rightarrow \wp(\mathbf{Con})$, such that we have $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R)$ if and only if there exists a role $R_2 \in \mathbf{Rol}$, such that $\mathcal{R} \vDash R \sqsubseteq R_2$ and $\forall R_2.C \in \mathit{clos}(\mathcal{T})$.

Example 8 (Example for an Extended ∀-info Structure)**:**
Let

$$\mathcal{T}_{Ex8} = \{Chair \sqsubseteq \forall headOf.Department, \exists memberOf.\top \sqsubseteq Person, GraduateStudent \sqsubseteq Student\}$$

and $\mathcal{R}_{Ex8} = \{headOf \sqsubseteq memberOf\}$, then the TBox in normal form is

$$\begin{aligned} \mathcal{T}_{Ex8norm} = \{ & \\ & \top \sqsubseteq \neg Chair \sqcup \forall headOf.Department, \top \sqsubseteq \forall memberOf.\bot \sqcup Person, \\ & \top \sqsubseteq \neg GraduateStudent \sqcup Student \\ \} & \end{aligned}$$

and the extended ∀-info structure for $\mathcal{T}_{Ex8norm}$ and $\mathcal{R}_{Ex8}$ is:

$$\mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R) = \begin{cases} \{Department, \bot\} & \text{if } R = headOf, \\ \{\bot\} & \text{if } R = memberOf, \\ \emptyset & \text{otherwise.} \end{cases}$$

The extended ∀-info structure allows us to check which concept descriptions are (worst-case) propagated over role assertions in $\mathcal{ALCH}$-ontologies. Please note that the definition of the extended ∀-info structure corresponds to the definition of the usual ∀-tableau rule in presence of role hierarchies.

Theorem 3 (Decision Criteria for ABox Splits in $\mathcal{ALCH}$-ontologies)**:**
Given an $\mathcal{ALCH}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}$, $\downarrow_{c,d}^{R(a,b)}$ is valid with respect to $\mathcal{O}$ if for each $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R)$

- $C = \bot$ or
- there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or

- there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$.

Proof of Theorem 3. Since the tableau rules for $\mathcal{ALCH}$ do not change compared to $\mathcal{ALC}$, but only the definition of neighbor relationships, the proof is a direct consequence of the results for $\mathcal{ALC}$ (Theorem 2). □

3.3.4. Consistency-preserving ABox Splits for $\mathcal{ALCHI}$

In presence of inverse roles, concept descriptions can be propagated in two directions over a role assertion $R(a_1, a_2)$. The extension of Theorem 3 to $\mathcal{ALCHI}$-ontologies is shown in Theorem 4.

Theorem 4 (Decision Criteria for ABox Splits in $\mathcal{ALCHI}$-ontologies)**:**
Given an $\mathcal{ALCHI}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow_{c,d}^{R(a,b)}, \downarrow_{c,d}^{R(a,b)}$ is valid with respect to $\mathcal{O}$ if

1. for each $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R)$
   - $C = \bot$ or
   - there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or
   - there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$

   and
2. for each $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R^-)$
   - $C = \bot$ or
   - there exists a concept description C_2, such that $C_2(a) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or
   - there exists a concept description C_2, such that $C_2(a) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$.

Proof of Theorem 4. Since the tableau rules for $\mathcal{ALCHI}$ do not change compared to $\mathcal{ALCH}$, but only the definition of neighbor relationships, the proof is a direct consequence of the results for $\mathcal{ALCH}$ (Theorem 3). □

3.3.5. Consistency-preserving ABox Splits for $\mathcal{SHI}$

We discuss the extension to transitive roles next. Please note that the additional $\forall_+$-tableau rule can only become applicable for role assertions with transitive roles. We formally define a class of $\mathcal{SHI}$-splittable role assertions, and prove that each ABox split for these role assertions is valid in $\mathcal{SHI}$-ontologies.

Definition 9 ($\mathcal{SHI}$-splittability of Role Assertions)**:**
Given a $\mathcal{SHI}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and a role assertion $R(a, b)$, we say that $R(a, b)$ is *$\mathcal{SHI}$-splittable with respect to $\mathcal{O}$* if

1. there exists no transitive role R_2 with respect to $\mathcal{R}$, such that $\mathcal{R} \vDash R \sqsubseteq R_2$,
2. for each $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R)$
   - $C = \bot$ or
   - there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or
   - there exists a concept description C_2, such that $C_2(b) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$

   and
3. for each $C \in \mathit{extinfo}_{\mathcal{T},\mathcal{R}}^{\forall}(R^-)$
   - $C = \bot$ or
   - there exists a concept description C_2, such that $C_2(a) \in \mathcal{A}$ and $\mathcal{T} \vDash C_2 \sqsubseteq C$ or
   - there exists a concept description C_2, such that $C_2(a) \in \mathcal{A}$ and $\mathcal{T} \vDash C \sqcap C_2 \sqsubseteq \bot$.

Please note that, although we have defined $\mathcal{SHI}$-splittability based on TBox-reasoning (for subsumption and disjointness tests), these structures do not have to be complete. Thus, only sound TBox-reasoning (e.g. syntactical analysis with some closure) is requiered in order to compute modularizations. Informally, the *more complete* the actual TBox analysis is, the more role assertions can possibly split up. In our experiments so far, most role assertions can be already split up based on simple analysis of direct (told) subsumptions in the TBox.

Lemma 8 (Consistency-preserving $\mathcal{SHI}$-ABox Splits)**:**
Given a $\mathcal{SHI}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow^{R(a,b)}_{c,d}$, $\downarrow^{R(a,b)}_{c,d}$ is a consistency-preserving ABox split for $\mathcal{O}$ if $R(a, b)$ is $\mathcal{SHI}$-splittable with respect to $\mathcal{O}$.

Proof of Lemma 8. In the same style as the proof of Lemma 6. We know that the only rule which could become applicable due to the role assertion addition is the $\forall_+$-tableau rule. However, since we assume that R does not have a transitive subsuming role ($\mathcal{SHI}$-splittability of R), we can conclude that the $\forall_+$-tableau rule is not applicable either. The same argumentation as before ($\mathcal{ALCHI}$) is true for non-applicability of the $\forall$-tableau rule. □

Theorem 5 is the extension of Theorem 4 from $\mathcal{ALCHI}$ to $\mathcal{SHI}$-ontologies.

Theorem 5 (Decision Criteria for ABox Splits in $\mathcal{SHI}$-ontologies)**:**
Given a $\mathcal{SHI}$-ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an ABox split $\downarrow^{R(a,b)}_{c,d}$, $\downarrow^{R(a,b)}_{c,d}$ is valid with respect to $\mathcal{O}$ if $R(a, b)$ is $\mathcal{SHI}$-splittable with respect to $\mathcal{O}$.

Proof of Theorem 4. By Lemma 2 we have soundness and by Lemma 8 we have completeness. □

In Example 9, we define an example ontology and then derive one intensional-based ABox modularization step by step.

Example 9 (Example Ontology for Intensional Modularization)**:**
The example ontology $\mathcal{O}_{Ex9} = \langle \mathcal{T}_{Ex9}, \mathcal{R}_{Ex9}, \mathcal{A}_{Ex9} \rangle$ is defined as follows

$$
\begin{aligned}
\mathcal{T}_{Ex9} = \{ & \\
& Chair \equiv \exists headOf.Department, Student \equiv \exists takes.Course, \\
& UndergraduateCourse \sqsubseteq Course, \\
& Course \sqcap Chair \sqsubseteq \bot, \top \sqsubseteq \forall takes.Course, \\
& \top \sqsubseteq \forall teaches.Course, \exists memberOf.\top \sqsubseteq Professor \\
\} &
\end{aligned}
$$

$$\mathcal{R}_{Ex9} = \{headOf \sqsubseteq memberOf, teaches \equiv isTaughtBy^{-}, Trans(suborgOf)\}$$

$$
\begin{aligned}
\mathcal{A}_{Ex9} = \{ & \\
& Department(cs), Department(ee), Professor(ann), Professor(eve), \\
& Professor(mae), UndergraduateCourse(c1), UndergraduateCourse(c4), \\
& UndergraduateCourse(c5), GraduateCourse(c2), GraduateCourse(c3), \\
& Student(ani), Student(ean), Student(eva), Student(noa), \\
& Student(sam), Student(sue), Student(zoe), \\
& headOf(ann, cs), memberOf(eve, cs), headOf(mae, ee), \\
& teaches(ann, c1), teaches(eve, c2), teaches(eve, c3), \\
& teaches(mae, c4), teaches(mae, c5), \\
& suborgOf(r, cs), suborgOf(cs, u1), suborgOf(ee, u1), \\
& takes(ani, c1), takes(ean, c1), takes(ean, c2), takes(eva, c3), \\
& takes(noa, c3), takes(sam, c4), takes(sue, c5), takes(zoe, c5) \\
& \}.
\end{aligned}
$$

Please note that absence of the concept inclusion axiom $GraduateCourse \sqsubseteq Course$ in $\mathcal{T}_{Ex9}$. Absence and presence of that axiom makes the impact of TBox modeling for $\mathcal{SHI}$-splittability clear. We add the axiom later again.
The extended $\forall$-info structure for $\mathcal{T}_{Ex9}$ and $\mathcal{R}_{Ex9}$ is:

$$
extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(R) = \begin{cases}
\{\neg Department, \bot\} & \text{if } R = headOf, \\
\{Course\} & \text{if } R = isTaughtBy^{-}, \\
\{\bot\} & \text{if } R = memberOf, \\
\{\neg Course, Course\} & \text{if } R = takes, \\
\{Course\} & \text{if } R = teaches, \\
\emptyset & \text{otherwise.}
\end{cases}
$$

We have for instance $\bot \in extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(headOf)$, because of the subsumption relationship between $headOf$ and $memberOf$ in the RBox $\mathcal{R}_{Ex9}$. Given the extended $\forall$-info structure for $\mathcal{O}_{Ex9}$, we can decide $\mathcal{SHI}$-splittability for each role assertion in $\mathcal{A}_{Ex9}$. For instance, the role assertion $memberOf(eve, cs)$ is $\mathcal{SHI}$-splittable because of

- $extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(memberOf) = \{\bot\}$ and
- $extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(memberOf^{-}) = \{\}$.

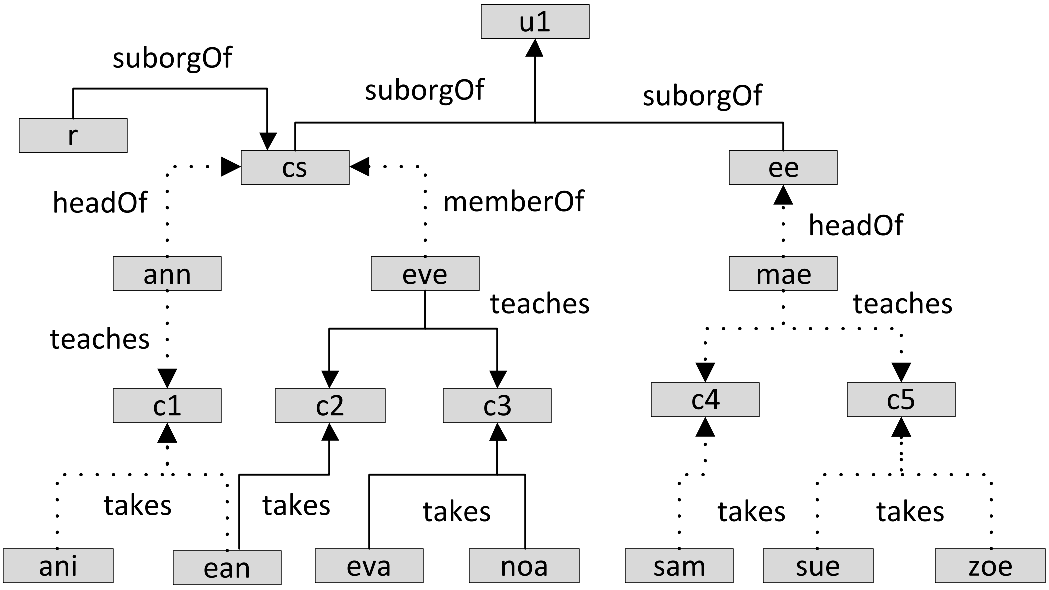
The role assertion $takes(noa, c3)$ is not $\mathcal{SHI}$-splittable because of

- $extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(takes) = \{\neg Course, Course\}$ and
- $extinfo^{\forall}_{\mathcal{T}_{Ex9},\mathcal{R}_{Ex9}}(takes^{-}) = \{\}$.

The problem is that the concept description $\neg Course$ can be propagated via role description $takes$. Since we only know that individual $c3$ is an instance of the concept description $GraduateCourse$, we cannot find an obvious propagation and neither a direct clash. Please note that this role assertion would be $\mathcal{SHI}$-splittable, if we had a subsumption axiom between $GraduateCourse$ and $Course$, because then the propagation of $\neg Course$ will be identified as a direct clash. Furthermore, all transitive $suborgOf$-role assertions are not $\mathcal{SHI}$-splittable.

In Fig. 3, we show all role assertions in $\mathcal{A}_{Ex9}$ and their $\mathcal{SHI}$-splittability. All $\mathcal{SHI}$-splittable role assertions are shown with dashed lines and all $\mathcal{SHI}$-unsplittable role assertions are shown with normal lines.

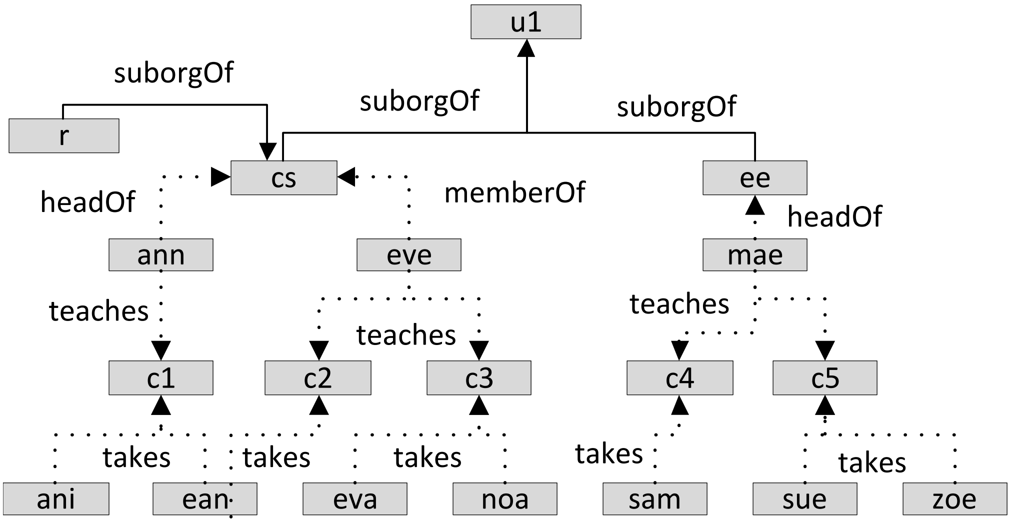
Fig. 3. $\mathcal{SHI}$-splittability for Example 9



In Fig. 4, we show all role assertions in $\mathcal{A}_{Ex9}$ and their $\mathcal{SHI}$-splittability, if the concept inclusion axiom $GraduateCourse \sqsubseteq Course$ was present. With the axiom included, all role assertions without transitive role descriptions in $\mathcal{A}_{Ex9}$ become $\mathcal{SHI}$-splittable. This simple example shows, how important the correct modeling of (maybe obvious) information can be for intensional modularization. Our experiments below show similar results for real world ontologies.

Without providing a formal proof, a more detailed look at ontology $\mathcal{O}_{Ex9}$ shows that the $suborgOf$-role assertions have only an influence on relation checking and retrieval, since there is no $\forall$-propagation over $suborgOf$ possible. Thus, for instance checking and retrieval, even transitive role assertions could be split up here. We discuss this special case in below again.

Fig. 4. $\mathcal{SHI}$-splittability for Example 9 with subsumption



We have shown until here that for reasoning over individuals in an ontology, only small modules might suffice. In the next section, we show how to use ABox modularization techniques in order to define algorithms for efficient instance checking over $\mathcal{SHI}$-ontologies.

4. Individual Islands

So far, we have introduced approaches to modularization of the assertional part of an ontology. In the following, we use these modularization techniques to define structures for efficient reasoning over ontologies.

We formally define a subset of assertions, called an individual island, which is worst-case necessary, i.e. possibly contains more assertions than really necessary, in order to have sound and complete instance checking. Informally speaking, we take the graph view of an ABox and, starting from a given individual, follow all role assertions in the graph until we reach a $\mathcal{SHI}$-splittable role assertion. We show that this strategy is sufficient for entailment of atomic concepts.

Usually, instance checking over ontologies is performed on the whole TBox, RBox, and ABox. Our goal is to formally identify a subset of assertions, called *individual island*, which is worst-case sufficient to perform sound and complete instance checking for a given individual. The formal foundations for these subsets of assertions have been set up before, where we show that, under some conditions, role assertions can be broken up while preserving soundness and completeness of instance checking algorithms. First, in Definition 10, we formally define an individual island candidate with an arbitrary subset of the original ABox. The concrete computation of the subset is then further defined below.

Definition 10 (Individual Island Candidate)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and a named individual $a \in Ind(\mathcal{A})$, an *individual island candidate*, is a tuple $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle$, such that $\mathcal{A}^{isl} \subseteq \mathcal{A}$. Given an individual island candidate $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle$ and an interpretation $\mathcal{I}$, we say that $\mathcal{I}$ is a *model of* ISL_a, denoted $\mathcal{I} \vDash ISL_a$, if $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \rangle$. Given an individual island candidate $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle$, we say that ISL_a *entails a concept assertion* $C(a)$, denoted $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle \vDash C(a)$, if for all interpretations $\mathcal{I}$, we have $\mathcal{I} \vDash ISL_a \implies \mathcal{I} \vDash C(a)$. We say that ISL_a *entails a role assertion* $R(a_1, a_2)$, denoted $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle \vDash R(a_1, a_2)$, if for all interpretations $\mathcal{I}$, we have $\mathcal{I} \vDash ISL_a \implies \mathcal{I} \vDash R(a_1, a_2)$.

Please note that entailment of concept and role assertions can be directly reformulated as a decision problem over ontologies, i.e. we have $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle \vDash C(a) \iff \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \rangle \vDash C(a)$. In order to evaluate the quality of an individual island candidate, we define soundness and completeness criteria for individual island candidates.

Definition 11 (Soundness and Completeness for Island Candidates)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an individual island candidate $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle$, we say that ISL_a is *sound for instance checking in ontology* $\mathcal{O}$ if for all atomic concept descriptions $C \in \mathbf{AtCon}$, $ISL_a \vDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a)$. ISL_a is *complete for instance checking in ontology* $\mathcal{O}$ if for all atomic concept descriptions $C \in \mathbf{AtCon}$, $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a) \implies ISL_a \vDash C(a)$.

We say that ISL_a is *sound for relation checking in ontology* $\mathcal{O}$ if for all role descriptions $R \in \mathbf{Rol}$ and all individuals $a_2 \in NInd(\mathcal{A})$

- $ISL_a \vDash R(a, a_2) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a, a_2)$ and
- $ISL_a \vDash R(a_2, a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a_2, a)$.

ISL_a is *complete for relation checking in ontology* $\mathcal{O}$ if for all role descriptions $R \in \mathbf{Rol}$ and all individuals $a_2 \in NInd(\mathcal{A})$

- $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a, a_2) \implies ISL_a \vDash R(a, a_2)$ and
- $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a_2, a) \implies ISL_a \vDash R(a_2, a)$.

We say that ISL_a is *sound for reasoning in ontology $\mathcal{O}$* if ISL_a is sound for instance and relation checking in $\mathcal{O}$. We say that ISL_a is *complete for reasoning in ontology $\mathcal{O}$* if ISL_a is complete for instance and relation checking in $\mathcal{O}$.

Definition 12 (Individual Island)**:**
Given an individual island candidate $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a\rangle$ for an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$, ISL_a is called *individual island for $\mathcal{O}$* if ISL_a is sound and complete for reasoning in $\mathcal{O}$.

An individual island candidate becomes an individual island if it can be used for sound and complete reasoning. It is easy to see that each individual island candidate is sound for reasoning since it contains a subset of the original ABox assertions.

In Fig. 5, we define an algorithm which computes an individual island starting from a given named individual a. The set $\mathbf{agenda}$ manages the individuals which have to be visited. The set $\mathbf{seen}$ collects already visited individuals. Individuals are visited if they are connected by a chain of $\mathcal{SHI}$-unsplittable role assertions to a. We add the role assertions of all visited individuals and all concept assertions for visited individuals and their direct neighbors.

Fig. 5. Naive algorithm for computation of an individual island

```
Input: Ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$, individual $a \in NInd(\mathcal{A})$
Output: Individual island $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a\rangle$
Algorithm:
    Let agenda $= a$
    Let seen $= \emptyset$
    Let $\mathcal{A}^{isl} = \emptyset$
    While agenda $\neq \emptyset$ do
        Remove $a_1$ from agenda
        Add $a_1$ to seen
        Let $\mathcal{A}^{isl} = \mathcal{A}^{isl} \cup \{C(a_1) \mid C(a_1) \in \mathcal{A}\}$
        For each $R(a_1, a_2) \in \mathcal{A}$
            $\mathcal{A}^{isl} = \mathcal{A}^{isl} \cup \{R(a_1, a_2) \in \mathcal{A}\}$
            If $R(a_1, a_2) \in \mathcal{A}$ is $\mathcal{SHI}$-splittable with respect to $\mathcal{O}$ then
                $\mathcal{A}^{isl} = \mathcal{A}^{isl} \cup \{C(a_2) \mid C(a_2) \in \mathcal{A}\}$
            else agenda $=$ agenda $\cup (\{a_2\} \setminus$ seen$)$
        For each $R(a_2, a_1) \in \mathcal{A}$
            $\mathcal{A}^{isl} = \mathcal{A}^{isl} \cup \{R(a_2, a_1) \in \mathcal{A}\}$
            If $R(a_2, a_1) \in \mathcal{A}$ is $\mathcal{SHI}$-splittable with respect to $\mathcal{O}$ then
                $\mathcal{A}^{isl} = \mathcal{A}^{isl} \cup \{C(a_2) \mid C(a_2) \in \mathcal{A}\}$
            else agenda $=$ agenda $\cup (\{a_2\} \setminus$ seen$)$
```

In Lemma 9, we show that the individual island of an individual suffices to decide entailment of atomic concept assertions for an individual.

Lemma 9 (Individual Island Dependencies)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$, for all named individuals $a \in NInd(\mathcal{A})$ and atomic concept descriptions C, if ISL_a is an individual island and $ISL_a \nvDash C(a)$ then there exists no individual $diff \in NInd(\mathcal{A})$, such that $ISL_{diff} \vDash C(a)$.

Proof of Lemma 9. By contradiction: Assume that $ISL_a \nvDash C(a)$ and there exists an individual island $ISL_{diff} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}_{diff}, diff\rangle$, such that $ISL_{diff} \vDash C(a)$. It is easy to see that $diff \neq a$ and $ISL_{diff} \neq ISL_a$ (ABox is not structurally equivalent). We know that all role assertions for individual a in ISL_{diff} are $\mathcal{SHI}$-splittable. Therefore, the role assertions for individual a can only be used to derive/propagate obvious concept descriptions. Since all the individual islands are consistent initially, we must have $ISL_{diff} \vDash C(a)$ only because of the presence of role assertions for individual a, concept assertions for a and its direct neighbors, and TBox axioms. Since all these axioms occur in ISL_a, we must have $ISL_a \vDash C(a)$ as well. Contradiction. □

Below, we show in Theorem 6 that the computed set of assertions is indeed sufficient for complete reasoning.

Theorem 6 (Island Computation yields Individual Island for Ontologies)**:**
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and an individual $a \in NInd(\mathcal{A})$, the algorithm in Fig. 5 computes an individual island $ISL_a = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl}, a \rangle$ for a.

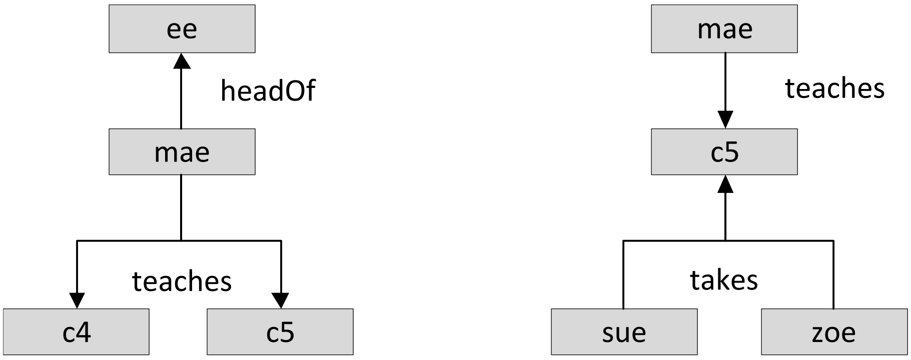
Proof of Theorem 6. The proof is done in four steps, following Definition 11:

- ISL_a is sound for instance checking in $\mathcal{O}$: We have to show that we have for all atomic concept descriptions $C \in \mathbf{AtCon}$ that $ISL_a \vDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a)$. Assuming $ISL_a \vDash C(a)$, it follows that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \rangle \vDash C(a)$, and thus $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \cup \{\neg C(a)\} \rangle$ is inconsistent. We know that $\mathcal{A}^{isl} \cup \{\neg C(a)\} \subseteq \mathcal{A} \cup \{\neg C(a)\}$. We can conclude that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{\neg C(a)\} \rangle$ is inconsistent, and thus $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a)$.
- ISL_a is sound for relation checking in $\mathcal{O}$: We have to show that we have for all role descriptions $R \in \mathbf{Rol}$ and all individuals $a_2 \in NInd(\mathcal{A})$ that
  - ∗ $ISL_a \vDash R(a, a_2) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a, a_2)$: By contraposition: We obtain $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash R(a, a_2) \implies ISL_a \nvDash R(a, a_2)$. Assuming $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash R(a, a_2)$, we know that there exists an interpretation $\mathcal{I}$, such that $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, but $\mathcal{I} \nvDash R(a, a_2)$. We know that $\mathcal{A}^{isl} \subseteq \mathcal{A}$, and thus $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \rangle$. By $\mathcal{I} \nvDash R(a, a_2)$ we can then conclude that $ISL_a \nvDash R(a, a_2)$.
  - ∗ $ISL_a \vDash R(a_2, a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a_2, a)$: By contraposition: We obtain $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash R(a_2, a) \implies ISL_a \nvDash R(a_2, a)$. Assuming $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash R(a_2, a)$, we know that there exists an interpretation $\mathcal{I}$, such that $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, but $\mathcal{I} \nvDash R(a_2, a)$. We know that $\mathcal{A}^{isl} \subseteq \mathcal{A}$, and thus $\mathcal{I} \vDash \langle \mathcal{T}, \mathcal{R}, \mathcal{A}^{isl} \rangle$. By $\mathcal{I} \nvDash R(a_2, a)$ we can then conclude that $ISL_a \nvDash R(a_2, a)$.
- ISL_a is complete for instance checking in $\mathcal{O}$: We have to show that for all atomic concept descriptions $C \in \mathbf{AtCon}$ and all individuals $a \in NInd(\mathcal{A})$ that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash C(a) \implies ISL_a \vDash C(a)$. By contraposition: We have to show that $ISL_a \nvDash C(a) \implies \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash C(a)$. Assume that $ISL_a \nvDash C(a)$. By Lemma 9, we know that no other individual island entails $C(a)$. Please note that the set of all individual islands can be rewritten to a component-based ABox modularization. Thus, by Definition 6 and Theorem 5, we know that $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \nvDash C(a)$.
- ISL_a is complete for relation checking in $\mathcal{O}$: We have to show that for all role descriptions $R \in \mathbf{Rol}$ and all individuals $a_2 \in NInd(\mathcal{A})$ that
  - ∗ $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a, a_2) \implies ISL_a \vDash R(a, a_2)$: There are three (combinations of) reasons for entailment of a role assertion $R(a, a_2)$ in a $\mathcal{SHI}$-ontology:
    - ∗ $R_2(a, a_2) \in \mathcal{A}$ and $\mathcal{O} \vDash R_2 \sqsubseteq R$: It is easy to see that all potentially useful role assertions $R_2(a, a_2)$ are in $\mathcal{A}^{isl}$, since, by the computation of islands in Fig. 5, all role assertions for a are added to $\mathcal{A}^{isl}$.
    - ∗ $R_2(a_2, a) \in \mathcal{A}$ and $\mathcal{O} \vDash R_2^- \sqsubseteq R$: It is easy to see that all potentially useful role assertions $R_2(a_2, a)$ are in $\mathcal{A}^{isl}$, since, by the computation of islands in Fig. 5, all role assertions for a are added to $\mathcal{A}^{isl}$.
    - ∗ a and a_2 are connected by a chain of (subroles of) transitive roles: By the definition of valid ABox splits and $\mathcal{SHI}$-splittability, each role assertion with a transitive superrole connected to an individual is not $\mathcal{SHI}$-splittable, and thus will end up in the $\mathcal{A}^{isl}$ computed by the algorithm in Fig. 5.
  - ∗ $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \vDash R(a_2, a) \implies ISL_a \vDash R(a_2, a)$: symmetric to the previous case.

□

In Fig. 6, we show two example individual islands for individual mae and individual $c5$ from Example 9.

Fig. 6. Example individual island for mae and $c5$ in Example 9



To summarize, we have shown that an individual island can be used for sound and complete instance checks. In the average case, the size of the individual island (with respect to the number of assertion in its ABox) is considerably smaller than the original ABox. In our experiments the size is usually orders of magnitudes smaller. For qualitative and quantitative results, see below. Please note that these modularization techniques allow traditional Description Logic reasoning systems to deal with ontologies which they cannot handle without modulariztions (because the data or the computed model abstraction does not fit into main memory).

5. Preliminary Evaluation

We have used two benchmark ontologies for evaluation of our modularization techniques: one synthetic benchmark introduced in [GPH05] and a real world multimedia annotation ontology used in the CASAM project and introduced in [GMN$^+$09]. The results for both ontologies are outlined below.

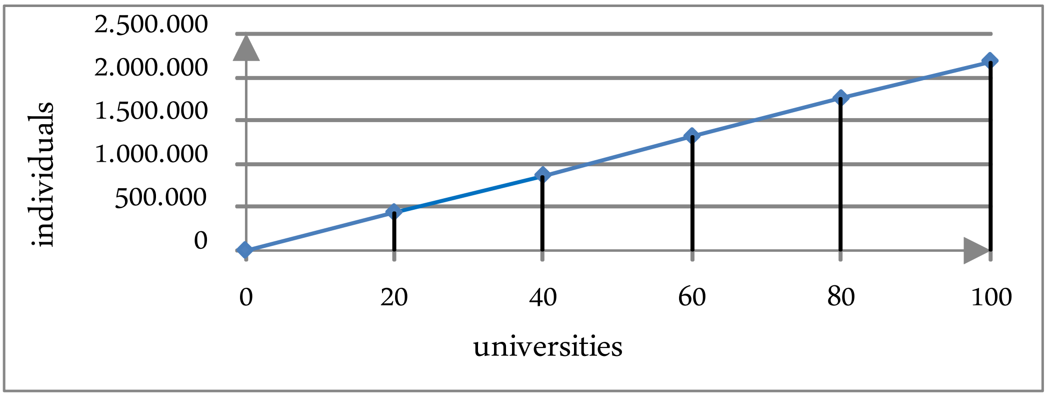
5.1. LUBM

The Lehigh University Benchmark, in short LUBM, is a synthetic ontology developed to benchmark knowledge base systems with respect to large OWL applications. The ontology is situated in the university domain. The background knowledge, i.e. the terminology, is described in a schema called Univ-Bench, see [GPH05] for an overview. The expressivity of the ontology is chosen to be in OWL Lite, which corresponds to the Description Logic $\mathcal{SHIF}$. However, the de facto expressivity is lower. For instance, the ontology does not introduce any functionality expressions on roles.

The terminology defines 43 classes and 32 properties (including 25 object properties and 7 datatype properties). The datatype properties are ignored in our experiments. According to [TV03], this ontology can be categorized as a Description Logic-style ontology which has a moderate number of classes but several restrictions and properties per class. The terminology of LUBM is rather simple.

While the terminological part of LUBM is static, the assertional part is dynamic in size and can be generated as big as necessary/desired. There exists a small tool written in Java, called Univ-Bench Artificial Data Generator. Given a number n as input, the tool generates n different universities, containing information about individuals, e.g. students, professors, publications and courses. The basic unit of a University is a Department. The number of departments varies by university. To make data creation more random, one can manually set a seed number as input to the data generator.

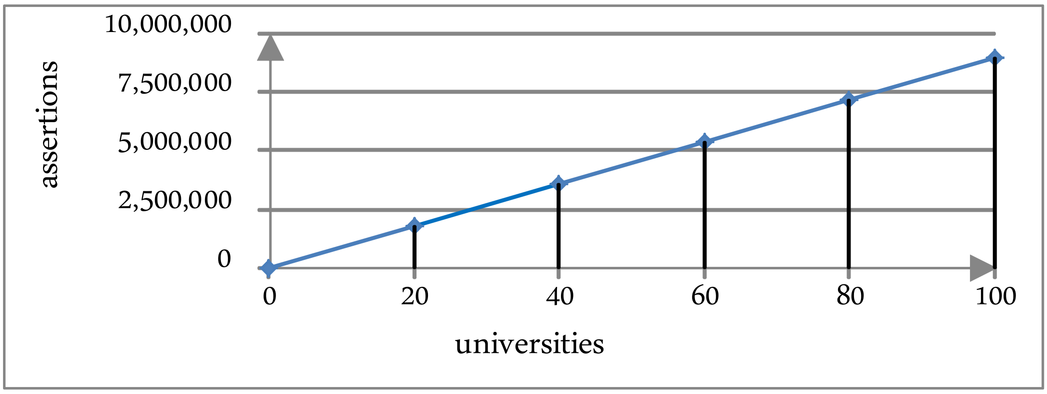
Fig. 7. Number of individuals in LUBM



In Fig. 7, we show the number of individuals in the dataset, for different numbers of universities. It can be seen that the number of individuals increases almost linearly with the number of universities. For details about the data distribution, see [GPH05].

In Fig. 8, the number of ABox assertions is shown. Most of the role assertions in the ontology cover the enrollment into a course (around 45 percent of the role assertions), being a publication author (around 22 percent of the role assertions) or being a member of an organization (around 15 percent of the role assertions). The remaining role assertions cover facts like, for instance, teaching a course or having a master degree from a university.

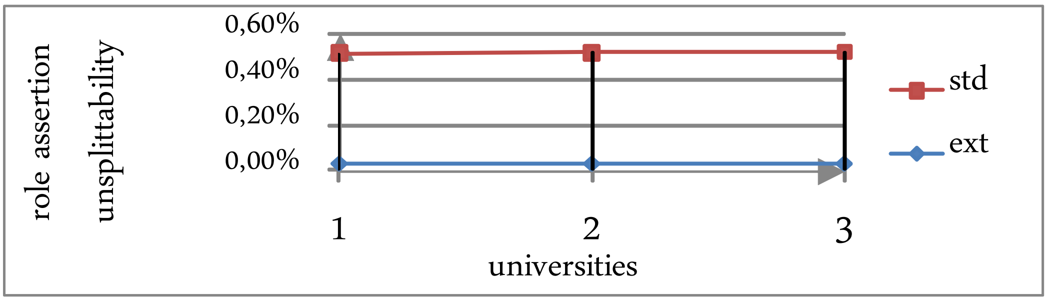
Fig. 8. Number of ABox assertions in LUBM



In the following, we investigate the efficiency of the ABox modularization techniques. The most important measure for efficiency seems to be the amount of $\mathcal{SHI}$-splittable role assertions, i.e. how many of the role assertions can be broken up. First of all, please note that component-based modularization of the assertional LUBM dataset yields one big module, i.e. each individual is connected to each other individual by a chain of role assertions. This is true for any number of universities. The connection between different universities is mainly because of degree-relationships between people and universities. Since only one ABox module is obtained, we do not provide any further statistics for component-based modularization.

The results for $\mathcal{SHI}$-splittability (from Definition 9) with respect to LUBM are shown in Fig. 9 with the label *std*. The dataset for LUBM 1, i.e. only one university, contains 49,336 role assertions, out of which 49,082 are $\mathcal{SHI}$-splittable. This means that only 0.5 percent of the role assertions are $\mathcal{SHI}$-unsplittable. This ratio does not change with a growing number of universities. Almost all $\mathcal{SHI}$-unsplittable role assertions have transitive roles, e.g. the role $suborganizationOf$. In addition, role assertions with the role

Fig. 9. Percentage of unsplittable role assertions in LUBM

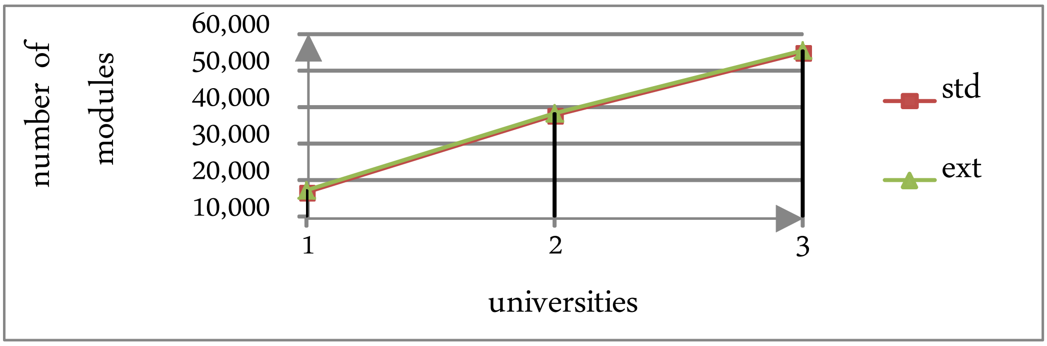


headOf are also $\mathcal{SHI}$-unsplittable, since, for instance, the not obvious concept description *Chair* can be propagated.

We have investigated an extended $\mathcal{SHI}$-splittability criteria, such that role assertions with transitive roles are splittable if all propagated concept descriptions are enforced by simple domain- or range-restrictions. In this case, without further proof, role assertions over transitive roles can be split up as well. The result for this extended splittability criterion are shown in Fig. 9 with the label *ext*. For the extended criterion and one university, only 15 role assertions (out of 49,336, 0.03 percent) turn out to be unsplittable. All these 15 role assertions contain the role *headOf*. For more universities, the ratio of unsplittable role assertions remains the same, since each department introduces exactly one head of the department.

Given the set of splittable role assertions, we can determine the number of ABox modules for different LUBM datasets. The results are shown in Fig. 10.

Fig. 10. Number of modules in LUBM



For component-based modularization, one big module is obtained, since each individual is related to each other individual by a chain of role assertions. With respect to $\mathcal{SHI}$-splittability, we obtain 16,920 modules for one university and 37,748 modules for two universities. With the extended criterion for splittability, i.e. improved handling of transitive roles, the number of modules can be further increased, as expected. For instance, for one university we obtain 17159 modules, instead of 16,920. Please remember that the number of individuals in one university is 17,174. Each ABox module contains in average 1.01 individuals.

In order to further evaluate the quality of ABox modules, we show the average size (measured in number of ABox assertions) of the modules in Fig. 11. For component-based modularization, the module is as big as the whole ABox (not shown). With respect to $\mathcal{SHI}$-splittability, the average size is between three and four ABox assertions per ABox module.

Fig. 11. Average size of modules in LUBM

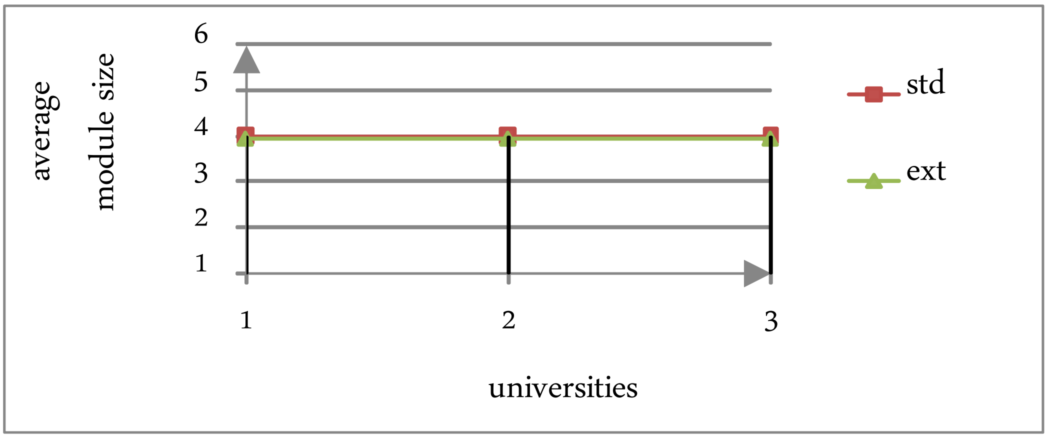
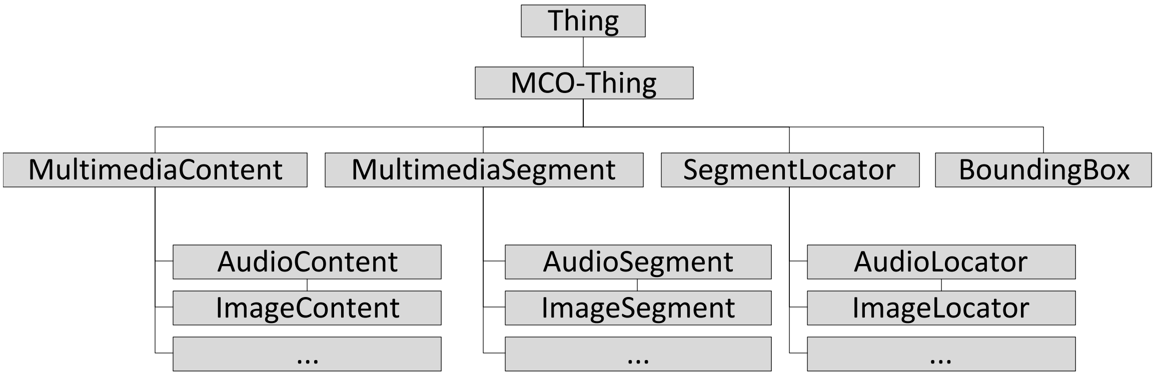


Fig. 12. Excerpt of the MCO concept classification



5.2. *CASAM Multimedia Content Ontology*

The CASAM project is focused on computer-aided semantic annotation of multimedia content. The novelty is the aggregation of human and machine knowledge. For a detailed discussion of the research objectives, see [GMN$^+$10], [PTP10], and [CLHB10]. Within the CASAM project, there is a need to define an expressive annotation language which allows for typical-case reasoning systems. The proposed annotation language is defined by the so-called Multimedia Content Ontology, short MCO, introduced in [GMN$^+$09]. Inspired by the MPEG-7 standard, see [IF02], strictly necessary elements describing the structure of multimedia documents are extracted. The intention is to exploit quantitative and qualitative time information in order to relate co-occurring observations about events in videos. Co-occurrences are detected either within the same or between different modalities, i.e. text, audio and speech, regarding the video shots.
In the following, we present small excerpts of MCO as far as relevant for understanding our evaluation results. A part of the concept classification is shown in Fig. 12.
An excerpt of the role classification is shown in Fig. 13. The role descriptions are used to relate multimedia objects with each other. Please note that role description $correlatesWith$ and its subroles are used to represent quantitative information as qualitative information. The roles d, m, and o are derived from the Allen-relations [All83], and represent *disjoint*, *meets*, and *overlapping* relations, respectively. The role descriptions $depicts$ and *hasInterpretation* map individuals of the MCO to observations/elements of an analysis module. Different interpretations are related for instance by the role description *associatedWith*. For more details about MCO please refer to [GMN$^+$09].

Fig. 13. Excerpt of the MCO role classification

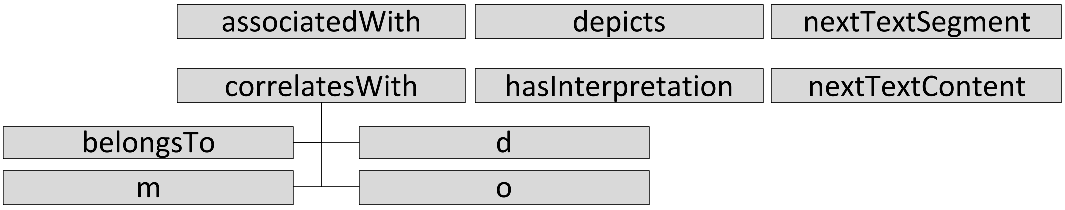
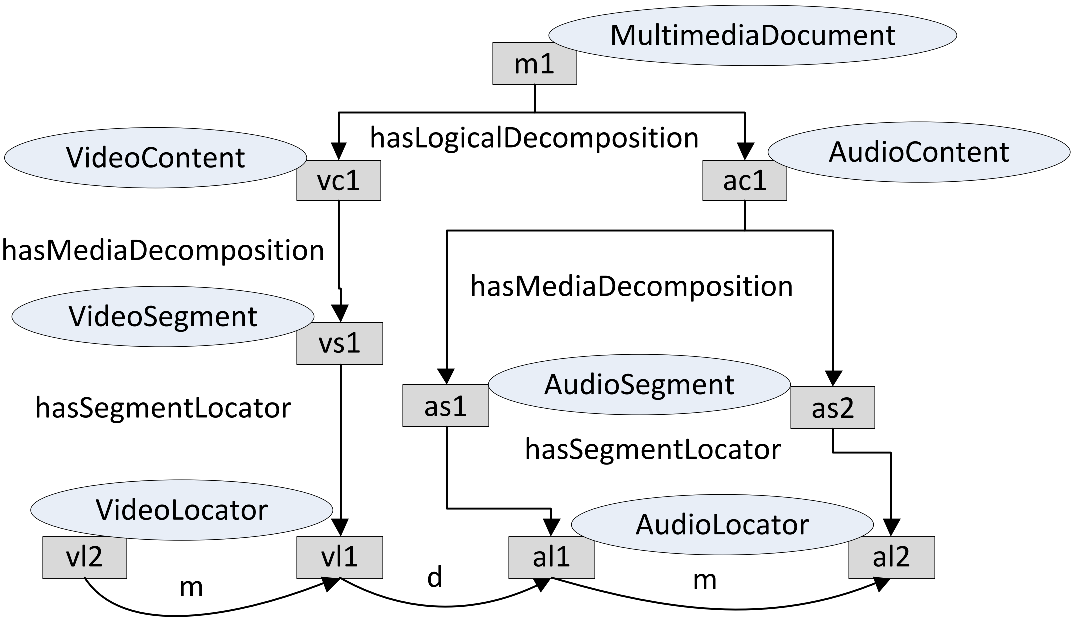


Fig. 14. MCO ABox example



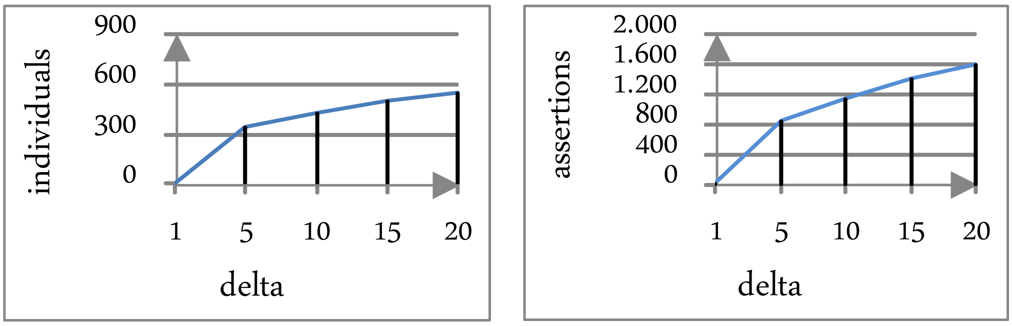
An excerpt of a multimedia document described with MCO is depicted in Fig. 14. The ABox excerpt contains the description of a multimedia document $m1$, which has video and audio content. The video content, named $vc1$, has a video segment $vs1$. The audio content, named $ac1$, is decomposed into several audio segments, such as $as1$ and $as2$. Each segment is associated with a locator and the locators are related by qualitative spatial/temporal relations.

For our evaluation with respect to MCO, we have a number of multimedia documents from the CASAM project. The source ontologies can be found in [CAS]. The set of test ontologies contains documents with identifiers ranging from 1 to 14. Each document is decomposed into several so-called *delta* files. Each delta represents additional information about the document of concern. We evaluated our modularization techniques with respect to all documents. Here we only show the results for Document 1, since for all other documents we obtained very similar statistics.

In Fig. 15, we show the number of individuals in the dataset, with an increasing delta. It can be seen that most individuals are introduced in the first delta files. The remaining delta files only introduce additional ABox assertions about already known individuals. The number of ABox assertions for different delta is also shown in Fig. 15. Please note that the number of individuals, as well as the number of ABox assertions is not linear in the number of delta. Thus, a MCO document cannot be directly used for evaluation purposes. At least one would have to consider the number of individuals up to the delta to read off more clear scalability results.

In the following, we investigate the efficiency of ABox modularization techniques. First of all, please note that component-based modularization of Document 1 yields one big module. This is true for all the other

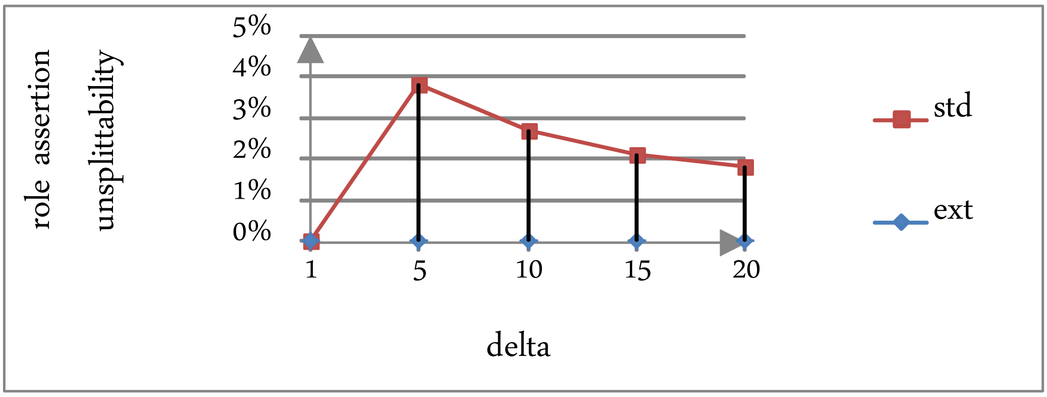
Fig. 15. Number of individuals and ABox assertions in Document 1



documents as well. Since only one ABox module is obtained, we do not provide any further statistics for the component-based modularization.

The results for $\mathcal{SHI}$-splittability (from Definition 9) with respect to MCO Document 1 are shown in Fig. 16 with the label *std*. The dataset for delta 1-5 contains 524 role assertions, out of which 504 are $\mathcal{SHI}$-splittable. This means that only 3 percent of the role assertions are $\mathcal{SHI}$-unsplittable. This ratio decreases with a growing number of deltas, because only $\mathcal{SHI}$-splittable role assertions are added. All $\mathcal{SHI}$-unsplittable role assertions have the transitive role $nextTextContent$. In addition, no other kinds of role assertions are $\mathcal{SHI}$-unsplittable.

Fig. 16. Percentage of unsplittable role assertions in Document 1

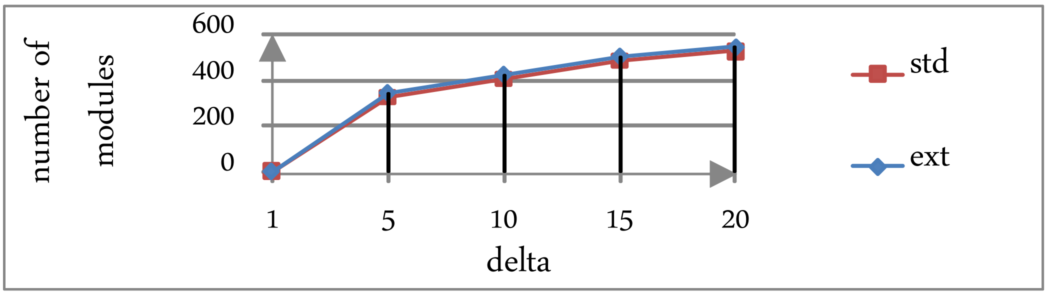


Again, we have investigated an extended $\mathcal{SHI}$-splittability criteria, such that role assertions with transitive roles are splittable if all propagated concept descriptions are enforced by simple domain- or range-restrictions. The result for the extended splittability criterion are shown in Fig. 16 with the label *ext*. For the extended criterion and any delta, no more role assertion is unsplittable, i.e. all role assertions in the ontology can be split up.

Given the set of splittable role assertions, we can determine the number of ABox modules for different delta. The results are shown in Fig. 17. For component-based modularization, one big module is obtained, since each individual is related to each other individual by a chain of role assertions. With respect to $\mathcal{SHI}$-splittability, we obtain 326 modules for five delta and 545 modules for 20 delta. With the extended criterion for splittability, i.e. improved handling of transitive roles, the number of modules can be further increased, as expected. For instance, for five delta we obtain 346 modules, instead of 326. Please remember that the number of individuals with five delta is 346. Each ABox module contains in average one individual.

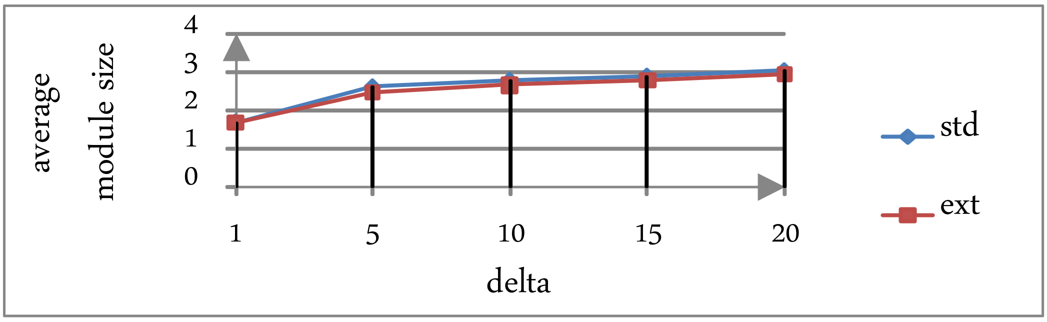
In order to further evaluate the quality of ABox modules, we show the average size (measured in number of ABox assertions) of the modules in Fig. 18. For component-based modularization, the module is as big

Fig. 17. Number of modules in Document 1



as the whole ABox (not shown). With respect to $\mathcal{SHI}$-splittability, the average size is between two and three ABox assertions per ABox module.

Fig. 18. Average size of modules in Document 1



6. Conclusions and Future Work

The main goal of this research was to introduce modularization techniques for ABoxes. We focused on the semi-expressive Description Logic $\mathcal{SHI}$, which can be seen as a first step towards more expressive Description Logics. We have derived criteria, called $\mathcal{SHI}$-splittability, for modularizing the ABox of an input ontology. The main technique used for modularization of ABoxes are ABox-splits, which break up role assertions in an ABox if particular conditions are satisfied. Role assertions can be broken up if, for instance, only obvious information is propagated. A graph component-based modularization can be used to extract a set of modules out of the ABox after breaking up all $\mathcal{SHI}$-splittable role assertions. Traditional Description Logic algorithms can then be used to reason over these ABox modules.

An interesting side effect is that our modularization techniques show that additional axioms in the TBox can help to reduce the average size of ABox modules, and thus, can improve instance checking times. One might think that additional axioms in an ontology always makes reasoning harder.

Based on modularization techniques, we have introduced the notion of individual islands for individuals in ABoxes. These individual islands can be used for sound and complete instance checking. Our evaluation shows that these individual islands are usually quite small and fit easily into main memory.

In [LW10], the authors investigate inseparability of ontologies with respect to a given signature. This technique, for the lightweight Description Logic $\mathcal{EL}$, can possibly be used in order to extract modules from ontologies and also in order to define similarity of modules with respect to a signature. In [KLPW10], the authors apply similar techniques in order to define decompositions of ontologies. It is shown that the decomposition is tractable for the Description Logic $\mathcal{EL}$ and not more complex than concept subsumption for more expressive Description Logics. The main difference to our results is that we focused on ABox modularization directly for semi-expressive ontologies and use pure syntactical analysis in order to define modules (or decompositions). This focus makes the implementation of incremental algorithms (under syntactical ontology updates) more easy. Our first tests are already quite encouraging.

In [TL10], the authors propose an index data structure for RDF data. The intention is to find similarities over instances in the RDF dataset by using bisimulations, i.e. something quite similar to our approach based on graph homomorphisms. The authors group bisimilar graph substructures, in order to reduce the complexity of query answering. The main difference to our modularization techniques is that we take the terminology into account for modularizing ABoxes. Individual islands can be potentially used to find similarities among ABox individuals as well.

In the following, we would like to discuss interesting directions for future work. The effectiveness of our modularization techniques can be further improved. For instance, TBox modularization techniques can contribute to smaller ABox modularizations. If we are able to split up the TBox into different modules, we could create one ABox modularization for each TBox module. Since each TBox module only contains a subset of assertions from the original TBox, it is clear that additional role assertions become $\mathcal{SHI}$-splittable. However, it needs to be shown, whether the overhead of several ABox modularizations in parallel, one for each TBox module, pays off. In addition, we think that further optimizations of our modularization techniques are possible. So far, we ensure entailment of all atomic concept descriptions. The number of $\mathcal{SHI}$-splittable role assertions might increase if the vocabulary is known and restricted beforehand.

An extension from the semi-expressive Description Logic $\mathcal{SHI}$ to $\mathcal{SHIQ}$ should be possible. Although our proof techniques are not directly applicable, we think that a syntactical analysis of the TBox and RBox can be used to identify a set of $\mathcal{SHIQ}$-unsplittable role assertions. Our homomorphism-based similarity criteria for individuals cannot be directly applied in the presence of cardinality restrictions. Further extensions, for instance to $\mathcal{SHOIQ}$, might be possible, but will surely require a lot of work and sophisticated analysis techniques. Furthermore, extensions to more expressive queries should be investigated (instance retrieval and grounded conjunctive queries). For grounded conjunctive queries, one can use cardinalities of the variable bindings (which can be obtained from our approach) in order to compute a preferred join plan over the role query atoms in the query.

Finally, more comprehensive experimental studies are required. Recently published work [SCH10] on new data generation algorithms for synthetic test ontologies might be one good place to start from. In general, we believe that our results carry over to other ontologies. However there exist scenarios, especially extensive use of transitive roles, which make it more hard to find fine-grained ABox modularizations.

References

[ACKZ09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite Family and Relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[Baa99] Franz Baader. Logic-Based Knowledge Representation. In *Artificial Intelligence Today*, pages 13–41. Springer-Verlag, 1999.

[BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In Kendall Clark and Peter F. Patel-Schneider, editors, *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.

[BCM$^+$07] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, New York, NY, USA, 2007.

[BHS05] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description Logics as Ontology Languages for the Semantic Web. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning*, volume 2605 of *Lecture Notes in Computer Science*, pages 228–248. Springer, 2005.

[BM07] Dan Brickley and Libby Miller. The Friend Of A Friend (FOAF) vocabulary specification. http://xmlns.com/foaf/spec/, November 2007.

[BS01] Franz Baader and Ulrike Sattler. An Overview of Tableau Algorithms for Description Logics. *Studia Logica*, 69(1):5–40, 2001.

[BS03] Alexander Borgida and Luciano Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *J. Data Semantics*, 1:153–184, 2003.

[CAS] Test Documents CASAM. MCO test documents. http://http://www.sts.tu-harburg.de/˜wandelt/casamtest.zip.

[CdK08] Ronald Cornet and Nicolette de Keizer. Forty years of SNOMED: a literature review. *BMC Med Inform Decis Mak*, 8 Suppl 1:S2, 2008.

[CLHB10] Chris Creed, Peter Lonsdale, Robert Hendley, and Russell Beale. Synergistic annotation of multimedia content. In *Proceedings of the 2010 Third International Conference on Advances in Computer-Human Interactions*, ACHI '10, pages 205–208, Washington, DC, USA, 2010. IEEE Computer Society.

[CPSK06] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Modularity and web ontologies. In *Proceedings of KR-2006*, pages 198–209. AAAI Press, 2006.

[DFK$^+$07] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Edith Schonberg, Kavitha Srinivas, and Li Ma. Scalable semantic retrieval through summarization and refinement. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 299–304. AAAI Press, 2007.

[DFK$^+$09] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Edith Schonberg, and Kavitha Srinivas. Scalable highly expressive reasoner (SHER). *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):357 – 361, 2009. Semantic Web challenge 2008.

[DS05] Andreas Doms and Michael Schroeder. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, 33(Web-Server-Issue):783–786, 2005.

[FKM$^+$06] Achille Fokoue, Aaron Kershenbaum, Li Ma, Edith Schonberg, and Kavitha Srinivas. The Summary Abox: Cutting Ontologies Down to Size. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 343–356. Springer Berlin / Heidelberg, 2006.

[FS06] Ulrich Furbach and Natarajan Shankar, editors. *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4130 of *Lecture Notes in Computer Science*. Springer, 2006.

[GFW08] Marcos André Gonçalves, Edward A. Fox, and Layne T. Watson. Towards a digital library theory: a formal digital library ontology. *Int. J. on Digital Libraries*, 8(2):91–114, 2008.

[GH06] Yuanbo Guo and Jeff Heflin. A Scalable Approach for Partitioning OWL Knowledge Bases. In *Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2006*, pages 636–641. Springer, 2006.

[GHLS07] Birte Glimm, Ian Horrocks, Carsten Lutz, and Uli Sattler. Conjunctive Query Answering in the Description Logic SHIQ. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007.

[GHS08] Birte Glimm, Ian Horrocks, and Ulrike Sattler. Unions of Conjunctive Queries in SHOQ. In *Proceedings of the 11th International Conference on the Principles of Knowledge Representation and Reasoning (KR-08)*, pages 252–262. AAAI Press/The MIT Press, 2008.

[GMN$^+$09] O. Gries, R. Möller, A. Nafissi, K. Sokolski, and M. Rosenfeld. CASAM Domain Ontology. Technical report, Hamburg University of Technology, 2009.

[GMN$^+$10] Oliver Gries, Ralf Möller, Anahita Nafissi, Maurice Rosenfeld, Kamil Sokolski, and Michael Wessel. A Probabilistic Abduction Engine for Media Interpretation Based on Ontologies. In Pascal Hitzler and Thomas Lukasiewicz, editors, *RR*, volume 6333 of *Lecture Notes in Computer Science*, pages 182–194. Springer, 2010.

[GPH05] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *J. Web Sem.*, 3(2-3):158–182, 2005.

[GR09] Birte Glimm and Sebastian Rudolph. Conjunctive Query Entailment: Decidable in Spite of O, I, and Q. In *Proceedings of the of the 2000 Description Logic Workshop (DL-09)*. CEUR Workshop Proceedings, 2009.

[HH08] Zhisheng Huang and Frank Harmelen. Using Semantic Distances for Reasoning with Inconsistent Ontologies. In *Proceedings of the 7th International Conference on The Semantic Web*, ISWC '08, pages 178–194, Berlin, Heidelberg, 2008. Springer-Verlag.

[HKP$^+$09] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer. W3C Recommendation, World Wide Web Consortium, October 2009.

[HMW04] V. Haarslev, R. Möller, and M. Wessel. Querying the Semantic Web with Racer + nRQL. In *Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL'04), Ulm, Germany, September 24*, 2004.

[HS99] Ian Horrocks and Ulrike Sattler. A Description Logic with Transitive and Inverse Roles and Role Hierarchies. *J. Log. Comput.*, 9(3):385–410, 1999.

[HT73] John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM*, 16:372–378, June 1973.

[HVHT05] Zhisheng Huang, Frank Van Harmelen, and Annette Ten Teije. Reasoning with inconsistent ontologies. In *Proceedings of the 19th International Joint Conference on Artificial intelligence*, pages 454–459, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[IF02] ISO/IEC15938-5FCD. Multimedia Content Description Interface (MPEG-7). http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm, 2002.

[KKS09] Sebastian Ryszard Kruk, Ewelina Kruk, and Katarzyna Stankiewicz. Evaluation of Semantic and Social Technologies for Digital Libraries. In Sebastian Ryszard Kruk and Bill McDaniel, editors, *Semantic Digital Libraries*, pages 203–214. Springer, 2009.

[KLPW10] Boris Konev, Carsten Lutz, Denis Ponomaryov, and Frank Wolter. Decomposing Description Logic Ontologies. In Fangzhen Lin and Ulrike Sattler, editors, *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR2010)*. AAAI Press, 2010.

[LW10] Carsten Lutz and Frank Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 45(2):194–228, 2010.

[Min74] Marvin Minsky. A Framework for Representing Knowledge. Technical report, MIT-AI Laboratory, Cambridge, MA, USA, 1974.

[Mot08] Boris Motik. KAON2 - Scalable Reasoning over Ontologies with Large Data Sets. *ERCIM News*, 2008(72):–1–1, 2008.

[MP06] Lutz Maicher and Jack Park, editors. *Charting the Topic Maps Research and Applications Landscape, First International Workshop on Topic Maps Research and Applications, TMRA 2005, Leipzig, Germany, October 6-7, 2005, Revised Selected Papers*, volume 3873 of *Lecture Notes in Computer Science*. Springer, 2006.

[MW88] David Maier and David Scott Warren. *Computing with Logic: Logic Programming with Prolog*. Benjamin/Cummings, 1988.

[PTP10] Katerina Papantoniou, George Tsatsaronis, and Georgios Paliouras. KDTA: Automated Knowledge-Driven Text Annotation. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *ECML/PKDD (3)*, volume 6323 of *Lecture Notes in Computer Science*, pages 611–614. Springer, 2010.

[PTZ09] Jeff Z. Pan, Edward Thomas, and Yuting Zhao. Completeness Guaranteed Approximations for OWL-DL Query Answering. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Description Logics*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[Qui68] Ross Quillian. Semantic memory. In *Semantic Information Processing*, pages 216–270. MIT Press, 1968.

[RPZ10] Yuan Ren, Jeff Z. Pan, and Yuting Zhao. Soundness Preserving Approximation for TBox Reasoning. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[SCH10] Giorgos Stoilos, Bernardo Cuenca Grau, and Ian Horrocks. How Incomplete is your Semantic Web Reasoner? In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 10)*, pages 1431–1436. AAAI Publications, 2010.

[SPC$^+$07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2):51–53, 2007.

[TL10] Duc Thanh Tran and Guenter Ladwig. Structure Index for RDF Data. In *Proceedings of the Workshop on Semantic Data Management (SemData) at the 36th International Conference on Very Large Databases (VLDB2010)*. VLDB Endowment, September 2010.

[TRKH08] Tuvshintur Tserendorj, Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Approximate OWL-reasoning with Screech. In Diego Calvanese and Georg Lausen, editors, *RR*, volume 5341 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2008.

[TV03] Christoph Tempich and Raphael Volz. Towards a benchmark for Semantic Web reasoners - an analysis of the DAML ontology library. In York Sure and Óscar Corcho, editors, *EON*, volume 87 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.