Annals of Mathematics and Artificial Intelligence 18 (1996) 133–157

Consistency checking reduced to satisfiability of concepts in terminological systems*

Bernhard Hollunder

Interactive Objects Software GmbH, Basler Str. 63, 79100 Freiburg, Germany
E-mail: hollunder@io.freinet.de

We investigate the inference problem in knowledge representation systems of the KL-ONE family. These systems, also called terminological systems, are equipped with concept languages that are used to express the conceptual knowledge of a problem domain in a structured way. In order to reason with the represented knowledge, terminological systems provide a couple of inference services. In this paper we show that the main reasoning problems in expressive concept languages can be reduced to a particular inference problem, namely checking satisfiability of concepts. This result has two important applications. From a practical point of view, our reduction together with the existence of relatively efficient implementations of satisfiability algorithms strongly simplifies the implementation of inference algorithms in terminological systems. Even from a complexity point of view, the result shows that in the underlying concept language interesting inference problems such as consistency checking or query answering are not harder (in terms of the worst case complexity) than satisfiability checking of concepts.

1. Introduction

Terminological knowledge representation systems in the tradition of KL-ONE [4], for example, BACK [17], CLASSIC [16], KRIS [1], and LOOM [13], are used to represent the taxonomic and conceptual knowledge of a problem domain in a structured way. To describe this kind of knowledge, one starts with primitive concepts and roles, and defines more complex concepts using the operations provided by a so-called concept language. Concepts are usually interpreted as sets of individuals and roles as binary relations between individuals, which means that concepts (roles) can also be viewed as unary (binary) predicates. Additionally, names for individuals (or objects) of a concrete world can be introduced by stating that an individual is an instance of a concept, or that two individuals are related by a role.

To give an example, assume that person, female, and rich are primitive concepts and that child is a role. If connectives such as *concept conjunction*, *concept disjunction*,

* This work has been carried out while the author was an employee of the German Research Center for AI (DFKI GmbH), Saarbrücken, Germany.

© J.C. Baltzer AG, Science Publishers

and *concept negation* are present in the concept language, one can describe "persons who are female or not rich" by the expression person $\sqcap$ (female $\sqcup \neg$rich). Since concepts are interpreted as sets, concept conjunction, concept disjunction, and concept negation are usually interpreted as set intersection, set union, and set complement, respectively. In addition to these operations on sets concept languages provide concept forming operations that employ roles. *Value-restrictions* and *exists-restrictions* can be used, for instance, to express "individuals with rich children only" by $\forall$child.rich and "individuals who have some child that is rich" by $\exists$child.rich. To impose cardinality restrictions on roles one can use *qualifying number restrictions*: For example, the concept ($\leqslant 2$ child $\neg$rich) denotes "individuals with at most two children which are not rich", and ($\geqslant 4$ child (female $\sqcap$ rich)) denotes "individuals with at least four rich daughters".

The so-called assertional formalism of a terminological system can be used to define instances of concepts and roles. This means that one can for instance express that Mary is a female person by Mary : (person $\sqcap$ female), or that Tom is a child of Mary by (Mary, Tom) : child.

Of course, terminological systems should not only be able to represent knowledge but should also provide facilities to reason with the represented knowledge. In fact, the systems mentioned before are equipped with a couple of inference services. The most important inference capabilities concerning concepts are the check whether a concept is more specific than another one (*subsumption of concepts*), and whether a concept can have an instance at all (*satisfiability of concepts*). Basic inference services taking the assertions on individuals into account are *consistency checking* and *instance checking*. Consistency checking is concerned with the question whether the knowledge base, i.e., the concept definitions together with the assertions on individuals, is consistent. Instance checking means to test whether an individual is instance of a concept, and can therefore be viewed as the basic inference task for retrieving information on individuals.

In the present paper we show how the consistency and instance checking problem can be reduced to the satisfiability problem of concepts. This result has two important applications. From an algorithmic point of view, one can exploit algorithmic techniques, which have already been developed for solving the satisfiability problem of concepts (see, e.g., [6, 10, 11]), to implement consistency and instance checking procedures. Also this method is a good basis for the investigation of complexity results. In fact, the proposed reduction in general yields consistency and instance checking procedures which are optimal with respect to the worst case complexity of the corresponding problems.

2. The representation formalism

In this section we formally introduce the formalism for representing knowledge in terminological systems. We start with defining a particular *concept language*, called

$\mathcal{ALCQ}$[1], which can be used to define the relevant concepts of a problem domain. The assertional formalism to be introduced thereafter allows one to describe objects of the problem domain with respect to their relation to concepts and their interrelation with each other.

Definition 1 (concept language $\mathcal{ALCQ}$). We assume two disjoint alphabets of symbols, called *primitive concepts* and *primitive roles*. The special primitive concepts $\top$ and $\bot$ are called the *top* and *bottom* concept. The sets of *concepts* and *roles* are inductively defined as follows. Every primitive concept is a concept. Now let C, D be concepts and R be a role already defined, and let n be a nonnegative integer. Then $C \sqcap D$ (conjunction), $C \sqcup D$ (disjunction), $\neg C$ (negation), $\forall R.C$ (value-restriction), $\exists R.C$ (exists-restriction), $(\geqslant n\ R\ C)$ and $(\leqslant n\ R\ C)$ (qualifying number restriction) are concepts and $R \mid C$ (range restriction) is a role of the language $\mathcal{ALCQ}$.

Assume, for example, that person, female, and rich are primitive concepts and that child is a primitive role. The concept person $\sqcap$ (rich $\sqcup$ $\neg$female) denotes the set of all "persons who are rich or not female". Value-restrictions and exists-restrictions can, for instance, be used to describe "individuals having rich children only" by the expression $\forall$child.rich and "individuals who have some rich child" by $\exists$child.rich. To impose cardinality restrictions on roles one can use qualifying number restrictions. For example, the concept $(\geqslant 2$ child $\top)$ denotes the set of all "individuals having at least two children", and the concept $(\leqslant 3$ (child | female) rich) all "individuals having at most three female children that are rich".

The following definition gives a formal semantics to our concept language.

Definition 2 (interpretation). An *interpretation* $\mathcal{I}$ of our concept language consists of a set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$). The interpretation function maps every primitive concept A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and every primitive role P to a subset $P^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The special primitive concepts $\top$ and $\bot$ are interpreted as $\Delta^{\mathcal{I}}$ and the empty set, respectively.

The interpretation function – which gives an interpretation for the primitive concepts and primitive roles – can be extended to arbitrary concepts and roles as follows. Let C, D be concepts, let R be a role, and let n be a nonnegative integer. Assume that $C^{\mathcal{I}}$, $D^{\mathcal{I}}$ and $R^{\mathcal{I}}$ are already defined. Then

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\
(\forall R.C)^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}} \right\},
\end{aligned}
$$

[1] This abbreviation stands for "Attributive concept Languages with Complements and Qualifying number restrictions".

$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\},$$
$$(\geqslant n\ R\ C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}| \geqslant n\},$$
$$(\leqslant n\ R\ C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}| \leqslant n\},$$
$$(R \mid C)^{\mathcal{I}} = \{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\},$$

where $|X|$ denotes the cardinality of a set X.

Qualifying number restrictions are relatively expressive language constructs: On the one hand, they generalize (ordinary) number restrictions, which are of the form $(\geqslant n\ R)$ and $(\leqslant n\ R)$, respectively, as follows: A specified number of role fillers for a role can be restricted to arbitrary concepts rather than only to the top concept. On the other hand, qualifying number restrictions can be viewed as generalizations of value- and exists-restrictions: A value-restriction $\forall R.C$ expresses that *all* role fillers of a role R belong to the concept C, whereas the concept $(\leqslant n\ R\ \neg C)$ states that *all but* n role fillers of R belong to C. Conversely, an "at least" qualifying number restriction $(\geqslant n\ R\ C)$ generalizes an exists-restriction $\exists R.C$ such that one can state that at least n fillers (rather than at least one) of the role R are in the concept C.

It can easily be verified that $(\geqslant n\ (R \mid C)\ D)^{\mathcal{I}} = (\geqslant n\ R\ (C \sqcap D))^{\mathcal{I}}$ and $(\leqslant n\ (R \mid C)\ D)^{\mathcal{I}} = (\leqslant n\ R\ (C \sqcap D))^{\mathcal{I}}$ for every interpretation $\mathcal{I}$, which shows that the role-forming operator $R \mid C$ can be eliminated within qualifying number restrictions. This – together with the fact that value- and exists-restrictions are special cases of qualifying number restrictions – shows that every concept C of the introduced concept language can be transformed to a concept D such that (1) D contains only the concept-forming operators conjunction, disjunction, negation, qualifying number restrictions and primitive roles, and (2) $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every interpretation $\mathcal{I}$.

The *assertional formalism* provided by most terminological systems allows the introduction of particular objects: One can state that objects belong to certain concepts, and that pairs of objects are instances of roles.

Definition 3 (assertion, ABox). Assume an alphabet of symbols, called *individuals*, disjoint from primitive concepts and roles to be given. A *concept assertion* is of the form $a\!:\!C$, and a *role assertion* is of the form $(a,b)\!:\!R$, where a, b are individuals, C is a concept, and R is a role. In the latter case we also say that b is an *R-successor of a*. An *assertion* is either a concept assertion or a role assertion, and a finite set of assertions is called an *ABox*.

For example, the assertions $(\mathsf{Mary}, \mathsf{Tom}) : \mathsf{child}$ and $\mathsf{Tom} : (\mathsf{male} \sqcap \mathsf{person})$ state that Mary has some child, Tom, who is a rich person.

Definition 4 (model for an ABox). The interpretation function $\cdot^{\mathcal{I}}$ of an interpretation $\mathcal{I}$ is extended to individuals by mapping them to elements of $\Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$. An interpretation $\mathcal{I}$ *satisfies* a concept assertion $a\!:\!C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and satisfies a role assertion $(a,b)\!:\!R$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. We say an interpretation $\mathcal{I}$ is a *model for an ABox $\mathcal{A}$* iff $\mathcal{I}$ satisfies all assertions in $\mathcal{A}$.

This restriction on the interpretation function ensures that individuals with different names are interpreted as distinct domain elements. This property is called *unique name assumption*, and is usually assumed in terminological systems.

Before we introduce the main inference services provided by terminological systems, we note that ABoxes can be viewed as a finite set of first-order formulas (for details see, e.g., [3]). However, formulas that correspond to ABox assertions belong to a restricted subclass of all first-order formulas for which important inference problems are decidable.

3. Inference services

This section introduces the basic inference services provided by terminological representation systems such as BACK [17], CLASSIC [16], KRIS [1], and LOOM [13].

In order to organize a set of concepts in a taxonomy with respect to their generality, subsumption between pairs of concepts plays an important role.

Subsumption of concepts: A concept C subsumes a concept D, written $D \sqsubseteq C$, iff $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ holds for all interpretation $\mathcal{I}$.

For instance, the concept $(\leqslant 1 \text{ child } (\text{female} \sqcap \neg \text{rich}))$ subsumes the concept $(\geqslant 2 \text{ child } (\text{female} \sqcap \text{rich})) \sqcap (\leqslant 3 \text{ child female})$. To see this, consider an individual having at least two female and rich children and at most three female children. But this means that the individual has at most one child which is female and not rich, which in fact shows that the former concept subsumes the latter.

In the past years the subsumption problem in concept languages has been thoroughly investigated. As a result, subsumption algorithms for various concept languages as well as their computational complexity are known. The seminal paper was due to Schmidt-Schauß and Smolka [18] in which the exact complexity of the subsumption problem in the language $\mathcal{ALC}$ has been determined.[2] Subsequently, it has been shown that the algorithmic technique developed in [18] could be applied to other concept languages in order to obtain complexity results as well as subsumption algorithms (see, e.g., [5–7, 10, 11]. These papers, however, do not directly describe subsumption algorithms but algorithms that solve the following, closely related problem.

Satisfiability of concepts: A concept C is satisfiable iff there is some interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \varnothing$.

Since C subsumes D if and only if $D \sqcap \neg C$ is not satisfiable, a satisfiability algorithm can in fact be used to solve the subsumption problem.

A satisfiability algorithm can also be used to check *equivalence* and *disjointness* of concepts, i.e., whether $C^{\mathcal{I}} = D^{\mathcal{I}}$ (equivalence) and $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \varnothing$ (disjointness) hold for every interpretation $\mathcal{I}$ and concepts C, D.

[2] $\mathcal{ALC}$ is the sublanguage of $\mathcal{ALCQ}$ in which neither qualifying number restrictions nor range restrictions are allowed.

Let us now turn to the reasoning facilities where individuals of ABoxes are involved. An obvious requirement on the represented knowledge is that it should be consistent since everything would be deducible from inconsistent knowledge (from a logical point of view). This is all the more important for terminological systems such as KRIS, in which – as we will see immediately – most inference problems are reduced to consistency checking. The underlying model-theoretic semantics allows a clear and intuitive definition of consistency.

Consistency of an ABox: An ABox is consistent iff it has a model.

Observe that the consistency problem of ABoxes is at least as hard as the satisfiability problem of concepts. In fact, a concept C is satisfiable if and only if the ABox $\{a : C\}$ is consistent where a is an arbitrary individual. This, of course, shows that the consistency problem of ABoxes can be viewed as a generalization of the satisfiability problem for concepts.

In addition to the consistency problem, another basic reasoning task is concerned with the retrieval of information. Due to the object-centered representation of information most terminological systems do not allow arbitrary (first-order) formulas as queries (as in the case of general theorem proving), but restrict the query language to ABox assertions. This kind of inferencing is usually called instance checking.

Instance checking: An assertion α is implied by an ABox $\mathcal{A}$, written $\mathcal{A} \models \alpha$, iff all models of $\mathcal{A}$ satisfy α.

Since our concept language allows for negation on concepts the problem of instance checking can be reduced to the consistency problem of an ABox. If α is a concept assertion $a : C$, then $\mathcal{A}$ implies $a : C$ iff $\mathcal{A} \cup \{a : \neg C\}$ is inconsistent. Now assume that α is a role assertion $(a, b) : R$. If R is a primitive role, $\mathcal{A}$ implies α iff $\alpha \in \mathcal{A}$. Otherwise, R is of the form

$$\big(\ldots((P \,|\, C_1) \,|\, C_2) \,|\, \ldots C_k\big),$$

where P is a primitive role. In this case $\mathcal{A}$ implies α iff (1) $(a, b) : P \in \mathcal{A}$, and (2) $\mathcal{A}$ implies $b : C_1 \sqcap \cdots \sqcap C_k$.

Most terminological representation systems provide their users with reasoning facilities for more compact information retrieval such as realization. Realization (which is sometimes also referred to as *recognition*) means the problem of finding those concept names of a so-called terminology (i.e., set of concept definitions, see section 7) a given individual is instance of.[3] Conversely, the *retrieval problem* means

[3] Actually, the systems do not return all concept names with this property, but only those which are minimal with respect to subsumption. These concepts, usually called *most specific concepts*, describe an individual most accurately with respect to the terminology.

to determine all individuals in the ABox that are instance of a given concept. Obviously, both the realization and retrieval problem can be implemented by iterated applications of an instance checking algorithm.

To sum up, we have seen that the introduced inference problems concerning concepts and ABoxes can be reduced to the consistency problem of ABoxes if, of course, the concept language provides concept negation as in the case of $\mathcal{ALCQ}$. Furthermore, the reductions can be computed in polynomial time, which shows that the consistency and instance checking problem are in the same complexity class (with respect to the classes P, NP, and PSPACE).

Note that this is in general not true for restricted concept languages. In [8] the complexity of the above mentioned inference problems has been determined for several sublanguages of $\mathcal{ALCQ}$ that do not contain general concept negation. It turned out that for a particular language the consistency problem is coNP-complete whereas the instance checking problem is PSPACE-complete.

The complexity results presented in [8] also show that for the investigated sublanguages of $\mathcal{ALCQ}$ consistency checking of ABoxes and satisfiability checking of concepts are in the same complexity class. For the language $\mathcal{ALC}$, this result follows immediately from the algorithm presented in [2]. In the following we show that the consistency problem is not harder than the satisfiability problem for the language $\mathcal{ALCQ}$. The idea behind the reduction is basically the same as the one mentioned in [2]. However, as we will see in the next section, the presence of qualifying number restrictions makes things much more complicated.

Before we present a reduction from the consistency problem of ABoxes to the satisfiability problem of concepts, we single out a special class of concepts. In the following we assume that concepts occurring in ABoxes are built up using primitive concepts, primitive roles, concept conjunction, concept disjunction, concept negation, as well as qualifying number restrictions.

Without loss of generality we assume that the concepts are in negation normal form, i.e., negation occurs immediately in front of primitive concepts different from $\top$ and $\bot$. Negation normal forms can be generated using the following, equivalence preserving rules: $\neg\top \to \bot$, $\neg\bot \to \top$, $\neg(C \sqcap D) \to \neg C \sqcup \neg D$, $\neg(C \sqcup D) \to \neg C \sqcap \neg D$, $\neg\neg C \to C$, $\neg(\geqslant 0\ R\ C) \to \bot$, $\neg(\geqslant n\ R\ C) \to (\leqslant (n-1)\ R\ C)$ for $n > 0$, as well as $\neg(\leqslant n\ R\ C)$ and $(\geqslant (n+1)\ R\ C)$. A concept which is obtained from a concept C by applying as long as possible the previous rules from left to right in a top-down manner is called the *negation normal form* of C and is denoted by $NNF(C)$.

In order to prove certain properties of concepts by induction, we need the notion of size. Let C be a concept containing primitive roles only. The *size* of C, denoted by $|C|$, is inductively defined as $|A| = 1$ for all primitive concepts A, $|\neg C| = |C|$, $|C \sqcap D| = |C \sqcup D| = |C| + |D|$, and $|(\geqslant n\ R\ C)| = |(\leqslant n\ R\ C)| = 1 + |C|$.

4. The reduction

The consistency problem of ABoxes can be viewed as a generalization of the satisfiability problem of concepts in the following sense: an ABox $\mathcal{A}$ is consistent only

if, for each individual a in $\mathcal{A}$, the concept $C_a := C_1 \sqcap \cdots \sqcap C_n$ is satisfiable, where $a{:}C_1, \ldots, a{:}C_n$ are all concept assertions on the individual a in $\mathcal{A}$. Moreover, if an ABox does not contain any role assertions, this strategy yields a sound and complete consistency algorithm, i.e., an ABox is consistent if and only if for all individuals a in $\mathcal{A}$ the concept C_a is satisfiable. Intuitively, this is due to the fact that – in the absence of role assertions – different individuals cannot influence each other.

In the presence of role assertions, however, this strategy may not detect that an ABox is inconsistent. As an example consider the ABox

$$\mathcal{A}_1 = \{\mathsf{Mary}{:}\forall \mathsf{child}.\mathsf{doctor},\ (\mathsf{Mary}, \mathsf{Tom}){:}\mathsf{child},\ \mathsf{Tom}{:}\neg\mathsf{doctor}\}.$$

Although both concepts $C_{\mathsf{Mary}} = \forall \mathsf{child}.\mathsf{doctor}$ and $C_{\mathsf{Tom}} = \neg\mathsf{doctor}$ are satisfiable, ABox $\mathcal{A}_1$ is inconsistent. The reason why this strategy does not detect the inconsistency in the example is that it ignores the interaction of role assertions with value-restrictions, i.e., it does not take into account that Tom must be a doctor (which follows from the fact that he is a child of Mary). To overcome this problem, the idea is to extend the ABox by certain concept assertions in such a way that role assertions do not carry any additional information and can therefore be discarded. In fact, if $\mathcal{A}_1'$ is obtained from ABox $\mathcal{A}_1$ by adding the assertion $\mathsf{Tom} : \mathsf{doctor}$, the strategy is now able to detect that $\mathcal{A}_1'$ (and hence $\mathcal{A}_1$) is inconsistent because the concept term $C_{\mathsf{Tom}} = \mathsf{doctor} \sqcap \neg\mathsf{doctor}$ is obviously not satisfiable.

Thus, if an ABox contains a role assertion $(a, b) : R$ and a concept assertion $a : \forall R.C$ we add the assertion $b : C$ (if it is not already in the ABox). Since the concept $\forall R.C$ is equivalent to $(\leqslant 0\ R\ \neg C)$ the rule treats appropriately this restricted form of "at most" qualifying number restrictions. To motivate how this rule can be generalized to deal with arbitrary "at most" restrictions let us consider the ABox

$$\mathcal{A}_2 = \{a{:}(\leqslant 2\ R\ A),\ a{:}(\leqslant 1\ R\ \neg A),\ (a,b){:}R,\ (a,c){:}R\}.$$

It can easily be verified that $\mathcal{A}_2$ is consistent. Furthermore, observe that each model for $\mathcal{A}_2$ satisfies either the assertion $b : A$ or $c : A$, which is due to the interaction of $a{:}(\leqslant 1\ R\ \neg A)$ with $(a, b){:}R$ and $(a, c){:}R$. This, however, shows that neglecting the role assertions in $\mathcal{A}_2$ means losing this information.

Thus, the idea is again to extend $\mathcal{A}_2$ by certain concept assertions such that the role assertions become redundant to some extent. In the above example this means that – in order to check whether the "at most" restrictions imposed on a can be satisfied – one has to know for each R-successor of a whether or not it is an instance of A (resp., $\neg A$). Thus, the approach is simply to nondeterministically choose the appropriate alternative, i.e., *if* $a{:}(\leqslant n\ R\ A)$ *and* $(a, b){:}R$ *are contained in an ABox then either add the assertion* $b : A$ *or* $b : \neg A$ (if it is not already in the ABox). In the previous example, we therefore extend in a first step $\mathcal{A}_2$ either by $b : A$ or $b : \neg A$, and in a second step either by $c : A$ or $c : \neg A$, which means that one gets four completions of $\mathcal{A}_2$. It can easily be verified that $\mathcal{A}_2$ is consistent if and only if one of the completions is consistent.

The obtained completions have the nice property that they can be checked on consistency purely by performing satisfiability tests for concepts, i.e., role assertions are (to some extent) redundant and can therefore be discarded. However, when removing role assertions from a completion, one has to decrease numbers in number restrictions. Consider, for example, the ABox $\mathcal{A}_2' = \mathcal{A}_2 \cup \{b : A,\ c : \neg A\}$. Note that c is an R-successor of a which is in $\neg A$. Therefore a does not have another R-successor different from c which is in $\neg A$ because $a{:}(\leqslant 1\ R\ \neg A)$ is in $\mathcal{A}_2'$. But this means that – when removing $(a,c){:}R$ from $\mathcal{A}_2'$ – one has to replace $a{:}(\leqslant 1\ R\ \neg A)$ by $a : (\leqslant 0\ R\ \neg A)$. Analogously, the assertion $a : (\leqslant 2\ R\ A)$ is substituted by $a{:}(\leqslant 1\ R\ A)$ (because b is an R-successor of a which is in A). Thus we end up with an ABox

$$\mathcal{B} = \big\{a{:}(\leqslant 1\ R\ A),\ a{:}(\leqslant 0\ R\ \neg A),\ b{:}A,\ c{:}\neg A\big\}.$$

Since $\mathcal{B}$ does not contain any role assertions, it is easy to verify that $\mathcal{B}$ is consistent if and only if the concepts $C_a = (\leqslant 1\ R\ A) \sqcap (\leqslant 0\ R\ \neg A)$, $C_b = A$, and $C_c = \neg A$ are satisfiable.

The need for a rule that treats "at least" number restrictions analogously, i.e., if $a{:}(\geqslant n\ R\ A)$ and $(a,b){:}R$ are contained in an ABox then either add the assertion $b{:}A$ or $b{:}\neg A$, is motivated by the following example. Assume that ABox $\mathcal{A}_3$ is given by

$$\big\{a{:}(\leqslant 2\ R\ \top),\ a{:}(\geqslant 1\ R\ A),\ a{:}(\geqslant 1\ R\ \neg A),\ (a,b){:}R\big\}.$$

The "at most" rule just described either adds the assertion $b : \top$ or $b : \neg\top$. Note that the second alternative yields immediately an inconsistent ABox. Thus consider $\mathcal{A}_3' = \mathcal{A}_3 \cup \{b{:}\top\}$. It can easily be verified that $\mathcal{A}_3'$, and hence $\mathcal{A}_3$, are consistent. However, the ABox

$$\big\{a{:}(\leqslant 1\ R\ \top),\ a{:}(\geqslant 1\ R\ A),\ a{:}(\geqslant 1\ R\ \neg A)\big\},$$

which is obtained from $\mathcal{A}_3'$ by removing the role assertion $(a,b){:}R$ and by replacing $a{:}(\leqslant 2\ R\ \top)$ by $a{:}(\leqslant 1\ R\ \top)$ is obviously inconsistent. The problem is that b is either in the concept A or $\neg A$, which means that either $a{:}(\geqslant 1\ R\ A)$ or $a{:}(\geqslant 1\ R\ \neg A)$ is satisfied. The idea to overcome the deficit is to *add either the assertion $b{:}A$ or $b{:}\neg A$ if the ABox contains $a{:}(\geqslant n\ R\ A)$ and $(a,b){:}R$ where $n > 0$*. In the example this means that we obtain $\mathcal{A}_3' \cup \{b{:}A\}$ or $\mathcal{A}_3' \cup \{b{:}\neg A\}$. Now, in order to check whether these ABoxes are consistent it suffices to invoke satisfiability tests for concepts. In fact, it can be easily checked that $\mathcal{A}_3' \cup \{b{:}A\}$ is consistent

iff the ABox $\{a{:}(\leqslant 1\ R\ \top),\ a{:}(\geqslant 0\ R\ A),\ a{:}(\geqslant 1\ R\ \neg A),\ b{:}A\}$ is consistent

iff the concepts $C_a = (\leqslant 1\ R\ \top) \sqcap (\geqslant 1\ R\ \neg A)$ and $C_b = A$ are satisfiable.

Thus we have motivated and informally described two rules (the "at most" and "at least" rule, respectively). However, these two rules alone are not sufficient. Suppose an ABox contains $(a,b) : R$ and $a : D \sqcap D'$ where D (resp., D') is of the form $(\geqslant n\ R\ A)$ or $(\leqslant n\ R\ A)$. In order to apply the rules just described we have

to decompose conjunctive concepts, i.e., ABoxes containing assertions of the form $a:D \sqcap D'$ are extended by $a:D$ and $a:D'$. For similar reasons we need a further rule that deals with disjunctive concepts. We will see that these four rules are sufficient to transform – in the so-called preprocessing step – an arbitrary ABox into an ABox where role assertions are redundant. In a second step, such ABoxes are checked on consistency by only testing satisfiability of concepts.

Definition 1 (preprocessing rules). The *preprocessing rules* are defined as follows:

1. $\mathcal{A} \to_{\sqcap} \{a:C_1,\ a:C_2\} \cup \mathcal{A}$

   if $a:C_1 \sqcap C_2$ is in $\mathcal{A}$, not both $a:C_1$ and $a:C_2$ are in $\mathcal{A}$.

2. $\mathcal{A} \to_{\sqcup} \{a:D\} \cup \mathcal{A}$

   if $a:C_1 \sqcup C_2$ is in $\mathcal{A}$, neither $a:C_1$ nor $a:C_2$ is in $\mathcal{A}$, and $D = C_1$ or $D = C_2$.

3. $\mathcal{A} \to_{\geqslant} \{b:D\} \cup \mathcal{A}$

   if $a:(\geqslant n\ R\ C), (a,b):R$ are in $\mathcal{A}$, neither $b:C$ nor $b:NNF(\neg C)$ is in $\mathcal{A}$, and $D = C$ or $D = NNF(\neg C)$.

4. $\mathcal{A} \to_{\leqslant} \{b:D\} \cup \mathcal{A}$

   if $a:(\leqslant n\ R\ C), (a,b):R$ are in $\mathcal{A}$, neither $b:C$ nor $b:NNF(\neg C)$ is in $\mathcal{A}$, and $D = C$ or $D = NNF(\neg C)$.

The first two rules are obvious, in fact they are defined as in [18]. Since concepts occurring in ABoxes are assumed to be in negation normal form, in the third and fourth rule we first compute a negation normal form $NNF(\neg C)$ when adding the fact that b is in the concept $\neg C$.

The following proposition shows that the preprocessing rules are correct.

Proposition 2. Let $\mathcal{A}$, $\mathcal{A}'$ be ABoxes.

1. If $\mathcal{A}'$ is obtained from $\mathcal{A}$ by application of the $\to_{\sqcap}$-rule then $\mathcal{A}'$ is consistent if and only if $\mathcal{A}$ is consistent.

2. If $\mathcal{A}'$ is obtained from $\mathcal{A}$ by application of the $\to_{\sqcup}$-, $\to_{\geqslant}$-, or $\to_{\leqslant}$-rule, then $\mathcal{A}$ is consistent if $\mathcal{A}'$ is consistent. Furthermore, if such a rule applies to $\mathcal{A}$, it can be applied in such a way that it yields an ABox $\mathcal{A}'$ such that $\mathcal{A}'$ is consistent if and only if $\mathcal{A}$ is consistent.

Proof. The claim for the $\to_{\sqcap}$-rule (resp., $\to_{\sqcup}$-rule) follows immediately from the fact that an interpretation $\mathcal{I}$ satisfies $a : C_1 \sqcap C_2$ iff $\mathcal{I}$ satisfies $a : C_1$ as well as $a : C_2$ ($\mathcal{I}$ satisfies $a : C_1 \sqcup C_2$ iff $\mathcal{I}$ satisfies $a : C_1$ or $a : C_2$). For the $\to_{\geqslant}$- and $\to_{\leqslant}$-rule nothing has to be shown because $b^{\mathcal{I}} \in D^{\mathcal{I}}$ or $b^{\mathcal{I}} \in (\neg D)^{\mathcal{I}} = (NNF(\neg D))^{\mathcal{I}}$ holds for every interpretation $\mathcal{I}$. □

To prove termination of the preprocessing rules, i.e., that there are no infinite chains of applications of preprocessing rules issuing from an ABox, we need the following definition.

Definition 3 (subconcepts). Let C be a concept. The *subconcepts of* C, denoted by $Sub(C)$, are recursively defined as follows. For a primitive concept A we define $Sub(A) = \{A\}$. If C is of the form

- $D_1 \sqcap D_2$ or $D_1 \sqcup D_2$ then $Sub(C) = \{C\} \cup Sub(D_1) \cup Sub(D_2)$,
- $\neg D$, $(\geqslant n\ R\ D)$, or $(\leqslant n\ R\ D)$ then $Sub(C) = \{C\} \cup Sub(D)$.

For an ABox $\mathcal{A}$ we define the *subconcepts of* $\mathcal{A}$, $Sub(\mathcal{A})$, by $\bigcup_{a:C \in \mathcal{A}} Sub(C)$.

Termination of the preprocessing rules is then an immediate consequence of the following observation. If an ABox $\mathcal{A}'$ is obtained from an ABox $\mathcal{A}$ by application of a preprocessing rule, then $\mathcal{A}' = \mathcal{A} \cup \{b : D\}$ where $b \in Ind(\mathcal{A})$, $D \in Sub(\mathcal{A})$ or $D = NNF(\neg C)$ for some $C \in Sub(\mathcal{A})$, and $b : D \notin \mathcal{A}$. Since the set of individuals, $Ind(\mathcal{A})$, and the set of subconcepts, $Sub(\mathcal{A})$, are finite for a given ABox $\mathcal{A}$, there are only finitely many assertions which could be added. Furthermore, every application of a preprocessing rule adds at least one assertion which is not yet contained in the ABox. This shows that one ends up after finitely many applications with an ABox to which no rule is applicable. Moreover, the length of every derivation issuing from an ABox $\mathcal{A}$ is bounded by $|Ind(\mathcal{A})| * (2 * |Sub(\mathcal{A})|) \leqslant 2 * |\mathcal{A}|^2$, where $|\mathcal{A}|$, the size of $\mathcal{A}$, is defined to be the number of assertions in $\mathcal{A}$. Hence we have the following result.

Proposition 4. Let $\mathcal{A}$ be an ABox. After at most $\mathcal{O}(|\mathcal{A}|^2)$ applications of preprocessing rules one obtains an ABox to which no preprocessing rule is applicable.

Definition 5 (preprocessing complete ABox). An ABox is called *preprocessing complete* iff none of the preprocessing rules can be applied.

Note that there may exist exponentially many preprocessing complete ABoxes issuing from an ABox $\mathcal{A}$, which however can be enumerated using polynomial space in the size of $\mathcal{A}$.

Proposition 6. Let $\mathcal{A}$ be an ABox.

1. If $\mathcal{A}$ is consistent, then there exists a consistent preprocessing complete ABox $\mathcal{A}'$ derivable from $\mathcal{A}$ by application of preprocessing rules.
2. If $\mathcal{A}$ is inconsistent, then every preprocessing complete ABox issuing from $\mathcal{A}$ is inconsistent.

Proof. 1. Suppose that $\mathcal{A}$ is consistent. Since applications of the $\rightarrow_{\sqcap}$-rule preserve consistency and inconsistency, and the $\rightarrow_{\sqcup}$-, $\rightarrow_{\geqslant}$-, and $\rightarrow_{\leqslant}$-rules can be applied in such a way that they yield a consistent ABox (Proposition 2), there exists a preprocessing complete ABox $\mathcal{A}'$ issuing from $\mathcal{A}$ (Proposition 4) that is consistent.

2. Suppose that there exists a preprocessing complete ABox $\mathcal{A}'$ issuing from $\mathcal{A}$ which is consistent. Then $\mathcal{A}$ is obviously consistent because $\mathcal{A} \subseteq \mathcal{A}'$. □

Note that the order in which the preprocessing rules are applied is *don't care* nondeterministic. Thus, to obtain all, possibly exponentially many, preprocessing complete ABoxes issuing from an ABox $\mathcal{A}$, we start with $\mathcal{A}$ and choose an arbitrary rule that is applicable to $\mathcal{A}$. If there is no such rule, $\mathcal{A}$ is already preprocessing complete and we are done. Otherwise, we generate all ABoxes that can be obtained from $\mathcal{A}$ by an application of the chosen rule. We again apply some preprocessing rule to each of these ABoxes etc., until we end up with preprocessing complete ABoxes. If we use a depth-first strategy, all preprocessing complete ABoxes issuing from $\mathcal{A}$ can be enumerated using polynomial space in the size of $\mathcal{A}$. This is due to the fact that both the length of every derivation and the branching factor, i.e., the number of ABoxes obtained by application of a single preprocessing rule to an ABox that is obtained from $\mathcal{A}$, is polynomially bounded by the size of $\mathcal{A}$.

Preprocessing complete ABoxes have the nice property that they can easily be transformed into ABoxes in which role assertions no longer carry any additional information. More precisely, assume that $\mathcal{A}$ is a preprocessing complete ABox. We show how to construct (in polynomial time) an ABox $\mathcal{A}'$ from $\mathcal{A}$ such that (1) $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent, and (2) $\mathcal{A}'$ does not contain role assertions. But this means that the consistency of $\mathcal{A}'$, and hence of $\mathcal{A}$, can be checked by performing only satisfiability tests for concepts.

The transformation, however, does not apply to any ABox, but only to those which do not contain obvious contradictions involving role assertions and "at most" restrictions.

Definition 7 (clash). We say an ABox $\mathcal{A}$ contains a *clash* iff

$$\{a{:}(\leqslant n\ R\ C),\ (a,b_1){:}R,\ \ldots,\ (a,b_m){:}R,\ b_1{:}C,\ \ldots,\ b_m{:}C\} \subseteq \mathcal{A},$$

where $n < m$ for some individual a, distinct individuals $b_1, \ldots, b_m$, some concept C, and some role R. An ABox without clash is called *clash-free*.

Since distinct individuals occurring in ABoxes are interpreted as distinct domain elements (which is due to the unique name assumption) we have the following fact.

Proposition 8. Any ABox containing a clash is inconsistent.

Now let us formally describe how to translate preprocessing complete, clash-free ABoxes into ones without role assertions.

Definition 9 (notation: $\mathcal{C}_a^{\mathcal{A}}$ and $\mathcal{A}_a$). For an ABox $\mathcal{A}$ and an individual a in $Ind(\mathcal{A})$ let $\mathcal{C}_a^{\mathcal{A}}$ be a set of concepts defined as follows: $C \in \mathcal{C}_a^{\mathcal{A}}$ iff

- $a{:}C \in \mathcal{A}$ where C is a primitive concept or a negated primitive concept, or

- $a{:}(\geqslant n\ R\ D) \in \mathcal{A}$, $m = n_{R,D,\mathcal{A}}(a)$, $n > m$, and $C = (\geqslant (n-m)\ R\ D)$, or
- $a{:}(\leqslant n\ R\ D) \in \mathcal{A}$, $m = n_{R,D,\mathcal{A}}(a)$, $n \geqslant m$, and $C = (\leqslant (n-m)\ R\ D)$,

where

$$n_{R,D,\mathcal{A}}(a) = \big|\{b \in \mathit{Ind}(\mathcal{A}) \mid (a,b){:}R \text{ and } b{:}D \text{ are in } \mathcal{A}\}\big|.$$

For an individual a in $\mathit{Ind}(\mathcal{A})$ we define $\mathcal{A}_a = \{a{:}C \mid C \in \mathcal{C}_a^{\mathcal{A}}\}$.

In other words, an ABox $\mathcal{A}_a$ can be viewed as the restriction of $\mathcal{A}$ to concept assertions imposed on a. Since we apply the definition to preprocessing complete ABoxes only, there is no need to carry over assertions of the form $a : C \sqcap D$ and $a{:}C \sqcup D$. Assume that $n_{R,D,\mathcal{A}}(a) \geqslant n$ for some ABox $\mathcal{A}$, some role R, some concept C, and some integer n. Then any assertion of the form $a{:}(\geqslant n'\ R\ D)$ with $n' \leqslant n$ is satisfied, and is therefore omitted. Finally, notice that if $\mathcal{A}$ is clash-free the number $n - n_{R,D,\mathcal{A}}(a)$ is nonnegative for every assertion $a{:}(\leqslant n\ R\ D)$ in $\mathcal{A}$.

Consider, for example, the preprocessing complete ABox

$$\mathcal{A} = \big\{a{:}(\leqslant 2\ R\ A),\ a{:}(\geqslant 4\ R\ B),\ (a,b){:}R,\ (a,c){:}R,\ b{:}A,\ b{:}B,\ c{:}\neg A,\ c{:}B\big\}.$$

Then $\mathcal{C}_a^{\mathcal{A}} = \{(\leqslant 1\ R\ A),\ (\geqslant 2\ R\ B)\}$, $\mathcal{C}_b^{\mathcal{A}} = \{A, B\}$, and $\mathcal{C}_c^{\mathcal{A}} = \{\neg A, B\}$, and hence $\mathcal{A}_a = \{a{:}(\leqslant 1\ R\ A),\ a{:}(\geqslant 2\ R\ B)\}$, $\mathcal{A}_b = \{b{:}A,\ b{:}B\}$, and $\mathcal{A}_c = \{c{:}\neg A,\ c{:}B\}$.

The following two sections provide a soundness and completeness proof for this transformation, i.e., we show that *a clash-free, preprocessing complete ABox $\mathcal{A}$ is consistent iff, for every a in $\mathit{Ind}(\mathcal{A})$, the ABox $\mathcal{A}_a$ is consistent.* The direction from left to right, which we refer to as soundness (of the reduction), is shown in section 5; the opposite direction is shown in section 6.

5. Soundness

An immediate idea to show the soundness of the translation could be the following. Suppose that $\mathcal{A}$ is a preprocessing complete ABox that is consistent. Then take an interpretation $\mathcal{I}$ which satisfies $\mathcal{A}$, and construct an interpretation $\mathcal{I}'$ from $\mathcal{I}$ by eliminating, for every role R, a tuple $(a^{\mathcal{I}}, b^{\mathcal{I}})$ from $R^{\mathcal{I}}$ if $\mathcal{A}$ contains an assertion $(a,b){:}R$. Consider, for example, the preprocessing complete ABox

$$\mathcal{A} = \big\{a{:}(\geqslant 2\ R\ A),\ (a,b){:}R,\ b{:}A,\ c{:}A\big\}.$$

It can easily be verified that the interpretation $\mathcal{I}$ defined by

$$\Delta^{\mathcal{I}} = \{a,b,c\},\ A^{\mathcal{I}} = \{b,c\},\ R^{\mathcal{I}} = \{(a,b),(a,c),(c,a)\},\ a^{\mathcal{I}} = a,\ b^{\mathcal{I}} = b,\ c^{\mathcal{I}} = c$$

is a model for $\mathcal{A}$. The interpretation $\mathcal{I}'$ is obtained from $\mathcal{I}$ by removing the tuple (a,b) from the set $R^{\mathcal{I}}$. By definition, we have $\mathcal{A}_a = \{a{:}(\geqslant 1\ R\ A)\}$, $\mathcal{A}_b = \{b{:}A\}$, $\mathcal{A}_c = \{c{:}A\}$, which in fact shows that $\mathcal{I}'$ is a model for $\mathcal{A}_a$, $\mathcal{A}_b$, and $\mathcal{A}_c$.

Although the approach yields the correct behavior for this example, it is in general problematic, i.e., the interpretation $\mathcal{I}'$ may not be a model for an ABox $\mathcal{A}_a$, where a is an individual occurring in $Ind(\mathcal{A})$.

To see this, let $\mathcal{A}'$ be the ABox which is obtained from $\mathcal{A}$ by adding the assertion $c:(\geqslant 1\ R\ (\geqslant 2\ R\ \top))$. Note that c does not have an R-successor in $\mathcal{A}'$, which means that $\mathcal{A}'$ is preprocessing complete. Furthermore observe that the above interpretation $\mathcal{I}$ satisfies the additional assertion (because the $R^{\mathcal{I}}$-filler a of c itself has two $R^{\mathcal{I}}$-fillers, b and c), which means that $\mathcal{I}$ is a model for $\mathcal{A}'$. However, the interpretation $\mathcal{I}'$ is not a model for $\mathcal{A}'_c = \{c:A,\ c:(\geqslant 1\ R\ (\geqslant 2\ R\ \top))\}$ (because the only $R^{\mathcal{I}'}$-filler a of c itself has exactly one $R^{\mathcal{I}'}$-filler).

The problem is that the interpretation $\mathcal{I}$ is too specific, i.e., it relates the images of two individuals of the ABox by some role (in the previous example the tuples (a, c) and (c, a), respectively), although this constraint is not necessarily enforced by the ABox. This observation leads to the following definition, which singles out a certain class of interpretations (models) for ABoxes.

Definition 10 (simple interpretation and model). Let $\mathcal{I}$ be an interpretation (resp., model) for an ABox $\mathcal{A}$. We say that $\mathcal{I}$ is a *simple interpretation* (resp., *simple model*) for $\mathcal{A}$ iff for all b in $Ind(\mathcal{A})$, $(d, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ implies that there is some a in $Ind(\mathcal{A})$ such that $a^{\mathcal{I}} = d$ and $(a, b):R \in \mathcal{A}$.

The following proposition shows that the above mentioned approach is valid for simple models.

Proposition 11. Let $\mathcal{I}$ be a simple model for a preprocessing complete ABox $\mathcal{A}$. Furthermore let $\mathcal{I}'$ be the interpretation defined as follows:

- $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$,
- $A^{\mathcal{I}'} = A^{\mathcal{I}}$ where A is a primitive concept,
- $R^{\mathcal{I}'} = R^{\mathcal{I}} \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid (a, b):R \in \mathcal{A}\}$ where R is a primitive role, and
- $a^{\mathcal{I}'} = a^{\mathcal{I}}$ where a is an individual name.

For every a in $Ind(\mathcal{A})$, $\mathcal{I}'$ is a model for the ABox $\mathcal{A}_a$.

Before we can prove the proposition we need the lemma:

Lemma 12. Let d be an element of $\Delta^{\mathcal{I}}$ such that $d \neq b^{\mathcal{I}}$ for all b in $Ind(\mathcal{A})$. If $d \in C^{\mathcal{I}}$ then $d \in C^{\mathcal{I}'}$ for every concept C.

Proof. We prove the lemma by induction on the size of C, where we assume that C is in negation normal form. If C is a primitive concept, then $C^{\mathcal{I}} = C^{\mathcal{I}'}$, and thus $d \in C^{\mathcal{I}}$ implies $d \in C^{\mathcal{I}'}$.

Now let $C = \neg A$ for a primitive concept A. Then $d \in \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} = \Delta^{\mathcal{I}'} \setminus A^{\mathcal{I}'}$, which shows that $d \in (\neg A)^{\mathcal{I}'} = C^{\mathcal{I}'}$.

If $C = C_1 \sqcap C_2$ then $d \in C_1^{\mathcal{I}}$ and $d \in C_2^{\mathcal{I}}$. By induction we conclude that $d \in C_1^{\mathcal{I}'}$ and $d \in C_2^{\mathcal{I}'}$, and therefore $d \in (C_1 \sqcap C_2)^{\mathcal{I}'}$. The case $C = C_1 \sqcup C_2$ can be shown similarly.

If $C = (\geqslant n\ R\ D)$ there exist elements $d_1, \ldots, d_n$ ($d_i \neq d_j$ if $i \neq j$) in $\Delta^{\mathcal{I}}$ such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$. Since we assumed $d \neq b^{\mathcal{I}}$ for all b in $Ind(\mathcal{A})$, we can conclude that $d_i \neq b^{\mathcal{I}}$ for all b in $Ind(\mathcal{A})$ (because $\mathcal{I}$ is a simple model for $\mathcal{A}$). But this means that $(d, d_i) \in R^{\mathcal{I}'}$ (definition of $R^{\mathcal{I}'}$) as well as $d_i \in D^{\mathcal{I}'}$ (induction). Thus $d \in (\geqslant n\ R\ D)^{\mathcal{I}'}$.

Finally suppose that $C = (\leqslant n\ R\ D)$. Let d' be an element of $\Delta^{\mathcal{I}'}$ such that $(d, d') \in R^{\mathcal{I}'}$ and $d' \in D^{\mathcal{I}'}$. We show that $(d, d') \in R^{\mathcal{I}}$ and $d' \in D^{\mathcal{I}}$. It then follows immediately that $d \in (\leqslant n\ R\ D)^{\mathcal{I}'}$. If $(d, d') \in R^{\mathcal{I}'}$ then $(d, d') \in R^{\mathcal{I}}$ (definition of $R^{\mathcal{I}'}$). To verify that $d' \in D^{\mathcal{I}}$, suppose to the contrary that $d' \in D^{\mathcal{I}'}$ and $d' \notin D^{\mathcal{I}}$. Thus $d' \in \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}$. But this means that $d' \in (NNF(\neg D))^{\mathcal{I}}$ and, by the induction hypothesis[4], we can conclude that $d' \in (NNF(\neg D))^{\mathcal{I}'}$. Hence $d' \notin D^{\mathcal{I}'}$ contradicting our assumption. Therefore we know that $d' \in D^{\mathcal{I}}$ if $d' \in D^{\mathcal{I}'}$. Since there exist at most n elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$, we can conclude that there are at most n elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}'}$ such that $(d, d_i) \in R^{\mathcal{I}'}$ and $d_i \in D^{\mathcal{I}'}$. This shows that $d \in (\leqslant n\ R\ D)^{\mathcal{I}'}$. □

Proof of Proposition 11. We prove the claim by showing that $\mathcal{I}'$ satisfies every assertion in the ABox $\mathcal{B} = \bigcup_{a \in Ind(\mathcal{A})} \mathcal{A}_a$. From this it follows immediately that $\mathcal{I}'$ is a model for $\mathcal{A}_a$ for every a in $Ind(\mathcal{A})$. By definition, $\mathcal{B}$ contains neither role assertions nor concept assertions of the form $a : C_1 \sqcap C_2$ and $a : C_1 \sqcup C_2$, respectively, for any individual a (definition of $\mathcal{A}_a$). We show that $\mathcal{I}'$ satisfies $a : C$ in $\mathcal{B}$.

First assume that C is a primitive concept. Since $a : C$ is contained in $\mathcal{A}$ and $\mathcal{I}$ satisfies $\mathcal{A}$, we can conclude that $a^{\mathcal{I}} \in C^{\mathcal{I}}$. But this shows that $a^{\mathcal{I}'} = a^{\mathcal{I}} \in C^{\mathcal{I}} = C^{\mathcal{I}'}$, which means that $\mathcal{I}'$ satisfies $a : C$.

If $C = \neg A$ for some primitive concept A, then $\mathcal{A}$ contains the assertion $a : \neg A$. Since $\mathcal{I}$ satisfies $\mathcal{A}$ and $A^{\mathcal{I}} = A^{\mathcal{I}'}$ for every primitive concept A, we can conclude that $a^{\mathcal{I}'} \in \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} = \Delta^{\mathcal{I}'} \setminus A^{\mathcal{I}'}$. Thus the assertion $a : \neg A$ is satisfied by $\mathcal{I}'$.

Now assume that $C = (\geqslant n\ R\ D)$. Then the assertion $a : (\geqslant (n+m)\ R\ D)$ is contained in $\mathcal{A}$ where $m = n_{R,D,\mathcal{A}}(a)$. Observe that $\mathcal{A}$ is preprocessing complete, which means that either the assertion $b : D$ or $b : NNF(\neg D)$ is contained in $\mathcal{A}$ if $(a, b) : R \in \mathcal{A}$. Since $\mathcal{I}$ is a model for $\mathcal{A}$, we have $b^{\mathcal{I}} \in D^{\mathcal{I}}$ if $b : D \in \mathcal{A}$ and $b^{\mathcal{I}} \in (\neg D)^{\mathcal{I}}$ if $b : NNF(\neg D) \in \mathcal{A}$. But this means that there exist n distinct elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d_i) \in R^{\mathcal{I}}$, $d_i \in D^{\mathcal{I}}$, and $d_i \neq b^{\mathcal{I}}$ for all b in $Ind(\mathcal{A})$ (because $\mathcal{I}$ is a simple model for $\mathcal{A}$). Thus we can conclude that $(a^{\mathcal{I}'}, d_i) \in R^{\mathcal{I}'}$ (definition of $\mathcal{I}'$), and $d_i \in D^{\mathcal{I}'}$ (Lemma 12). Therefore $a^{\mathcal{I}'} \in (\geqslant n\ R\ D)^{\mathcal{I}'}$.

Finally let $C = (\leqslant n\ R\ D)$. We first claim that there are at most n elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that (1) $(a^{\mathcal{I}}, d_i) \in R^{\mathcal{I}}$, $d_i \in D^{\mathcal{I}}$, and (2) $d_i \neq b^{\mathcal{I}}$ for all b

[4] Observe that $|NNF(\neg D)| \leqslant |\neg D| = |D| < |(\leqslant n\ R\ D)| = |C|$, which shows that the size of $NNF(\neg D)$ is strictly smaller than the size of C.

in $Ind(\mathcal{A})$. This follows immediately from the facts that (1) $\mathcal{A}$ contains an assertion $a:(\leqslant (n+m)\ R\ D)$ where $m = n_{R,D,\mathcal{A}}(a)$, (2) the $\rightarrow_{\leqslant}$-rule does not apply to $\mathcal{A}$, and (3) $\mathcal{I}$ is simple. Thus it remains to be shown that $(a^{\mathcal{I}'}, d') \in R^{\mathcal{I}'}$ and $d' \in D^{\mathcal{I}'}$ implies that $(a^{\mathcal{I}}, d') \in R^{\mathcal{I}}$ and $d' \in D^{\mathcal{I}}$. Observe that $d' \neq b^{\mathcal{I}}$ for all b in $Ind(\mathcal{A})$. If $(a^{\mathcal{I}'}, d') \in R^{\mathcal{I}'}$ then $(a^{\mathcal{I}}, d') \in R^{\mathcal{I}}$ (definition of $R^{\mathcal{I}'}$). To see that $d' \in D^{\mathcal{I}}$ suppose to the contrary that $d' \in D^{\mathcal{I}'}$ and $d' \notin D^{\mathcal{I}}$. Thus $d' \in \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}} = NNF(\neg D)^{\mathcal{I}}$. By Lemma 12 we have $d' \in \Delta^{\mathcal{I}'} \setminus D^{\mathcal{I}'}$. Hence $d' \notin D^{\mathcal{I}'}$, contradicting the assumption $d' \in D^{\mathcal{I}'}$, which shows that $d' \in D^{\mathcal{I}}$. Thus $a^{\mathcal{I}'} \in (\leqslant n\ R\ D)^{\mathcal{I}'}$ holds. □

In order to give the final proof for the soundness of our approach, i.e., if $\mathcal{A}$ is consistent then, for every individual a, the ABox $\mathcal{A}_a$ is also consistent, we prove that an ABox has a model if and only if it has a simple model. We do this by showing that a given model $\mathcal{I}$ for an ABox $\mathcal{A}$ can be transformed into a simple one for $\mathcal{A}$. The idea behind the transformation is as follows: If $\mathcal{I}$ is already a simple model, nothing has to be done. Otherwise, there is some individual b in $Ind(\mathcal{A})$ such that the tuple $(d, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, where d is some element in $\Delta^{\mathcal{I}}$ and R is some role, violates the condition of a simple model, i.e., there is no a in $Ind(\mathcal{A})$ such that $(a,b):R \in \mathcal{A}$ and $a^{\mathcal{I}} = d$. The idea is to replace, for every role S, the element $b^{\mathcal{I}}$ in tuples $(e, b^{\mathcal{I}}) \in S^{\mathcal{I}}$ which violate the condition of a simple model by a new domain element which is a "copy" of $b^{\mathcal{I}}$.

To illustrate this idea consider the ABox

$$\mathcal{A} = \{a:(\geqslant 2\ R\ A),\ (a,b):R,\ b:A,\ c:A,\ c:(\geqslant 1\ R\ (\geqslant 2\ R\ \top))\}$$

and the model $\mathcal{I}$ for $\mathcal{A}$ given by

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

Observe that $\mathcal{I}$ is not a simple model for $\mathcal{A}$ because $R^{\mathcal{I}}$ contains the tuples (a,c) and (c,a). Thus, in a first step, we generate a new element, say c', which behaves exactly like c with respect to concept membership. This requirement can be met by putting c' into those primitive concepts A which contain c, and by adding a tuple (c', d) to $R^{\mathcal{I}}$ if (c,d) is in $R^{\mathcal{I}}$. This yields the interpretation $\mathcal{I}_1$ where

$$\begin{aligned}
\Delta^{\mathcal{I}_1} &= \{a,b,c,c'\},\\
A^{\mathcal{I}_1} &= \{b,c,c'\},\\
R^{\mathcal{I}_1} &= \{(a,b),(c,a),(a,c'),(c',a)\},\\
d^{\mathcal{I}_1} &= d^{\mathcal{I}} \text{ for all individuals } d \text{ in } Ind(\mathcal{A}).
\end{aligned}$$

Observe that $R^{\mathcal{I}_1}$ does not contain the tuple (a,c). It can easily be verified that $\mathcal{I}_1$ is a model for $\mathcal{A}$. However, $\mathcal{I}_1$ is not simple, because the tuples (c,a), (c',a) are

contained in $R^{\mathcal{I}_1}$. Therefore, in the second step, we introduce a new element, say a', which behaves like a, and we obtain the interpretation $\mathcal{I}_2$ where

$$\Delta^{\mathcal{I}_2} = \{a, b, c, c', a'\},$$
$$A^{\mathcal{I}_2} = \{b, c, c'\},$$
$$R^{\mathcal{I}_2} = \{(a, b), (a, c'), (c, a'), (c', a'), (a', b), (a', c')\},$$
$$d^{\mathcal{I}_2} = d^{\mathcal{I}_1} \text{ for all individuals } d \text{ in } \mathit{Ind}(\mathcal{A}).$$

Again, the thus obtained interpretation $\mathcal{I}_2$ is a model for $\mathcal{A}$, but is not a simple one (because (a', b) is an element of $R^{\mathcal{I}_2}$). We therefore construct the interpretation $\mathcal{I}_3$ by adding a copy, b', for b. This yields

$$\Delta^{\mathcal{I}_3} = \{a, b, c, c', a', b'\},$$
$$A^{\mathcal{I}_3} = \{b, c, c', b'\},$$
$$R^{\mathcal{I}_3} = \{(a, b), (a, c'), (c, a'), (c', a'), (a', c'), (a', b')\},$$
$$d^{\mathcal{I}_3} = d^{\mathcal{I}_2} \text{ for all individuals } d \text{ in } \mathit{Ind}(\mathcal{A}).$$

One can easily verify that $\mathcal{I}_3$ is a simple model for $\mathcal{A}$.

To sum up, we have constructed a chain $\mathcal{I}$, $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$ of models for $\mathcal{A}$ such that the final element of the chain, $\mathcal{I}_3$, is a simple model.

However, to guarantee that there is always a last element in such a chain of models for an ABox we have to refine the above idea. Consider, for example, a model $\mathcal{I}$ for an ABox $\mathcal{A}$ such that $R^{\mathcal{I}} = \{(d, e), (e, e)\}$. Assume that $a^{\mathcal{I}} = e$ for some a in $\mathit{Ind}(\mathcal{A})$, $(a, a) : R$ is in $\mathcal{A}$, and $b^{\mathcal{I}} \neq d$ for all b in $\mathit{Ind}(\mathcal{A})$. Note that $\mathcal{I}$ is not a simple model for $\mathcal{A}$ because $(d, e) \in R^{\mathcal{I}}$. When applying the above idea we construct a model, $\mathcal{I}_1$, by introducing a copy, say e_1, for e, which means that

$$R^{\mathcal{I}_1} = \{(d, e_1), (e, e), (e_1, e)\}.$$

Since $\mathcal{I}_1$ is not simple (because of (e_1, e)), we introduce a further copy of e, say e_2. Thus we get a model, $\mathcal{I}_2$, where

$$R^{\mathcal{I}_2} = \{(d, e_1), (e, e), (e_1, e_2), (e_2, e)\}.$$

But this shows that in proceeding this way we would never end up with a model, say $\mathcal{I}_n$, that is simple. In fact, since $R^{\mathcal{I}_n}$ contains a tuple (e_n, e), where e_n is a new domain element, we again have to generate a new element, say e_{n+1}, which is a copy of e_n. Hence we get $(e_{n+1}, e) \in R^{\mathcal{I}_{n+1}}$ for the next interpretation $\mathcal{I}_{n+1}$. The reason for this kind of looping is that we would generate infinitely many copies of the element e. However, it turns out that it suffices to generate at most *one* copy for every element. In the above example, this means that we take e_1 as a copy of e rather than introducing the new element e_2 when moving from the model $\mathcal{I}_1$ to $\mathcal{I}_2$. This modification yields $R^{\mathcal{I}_2} = \{(d, e_1), (e, e), (e_1, e_1)\}$, which in fact means that $\mathcal{I}_2$ is simple.

Instead of introducing a copy for *exactly* one element when moving from one interpretation to the next one, we generate copies for all elements, which are images of ABox individuals, at once.

Definition 13 (straight interpretation). Let $\mathcal{I}$ be an interpretation for an ABox $\mathcal{A}$. For each element $d \in \Delta^{\mathcal{I}}$ such that $d = a^{\mathcal{I}}$ for some $a \in \mathit{Ind}(\mathcal{A})$ we introduce a new element $\widehat{d}$. The set of newly introduced elements is denoted by $\mathcal{O}$. The *straight interpretation* $\mathcal{I}'$ for $\mathcal{I}$ is defined as follows:

- $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}} \cup \mathcal{O}$,
- $A^{\mathcal{I}'} = A^{\mathcal{I}} \cup \{\widehat{d} \in \mathcal{O} \mid d \in A^{\mathcal{I}}\}$ where A is a primitive concept,
- $a^{\mathcal{I}'} = a^{\mathcal{I}}$ where a is an individual name in $\mathit{Ind}(\mathcal{A})$.

To define the set $R^{\mathcal{I}'}$ for a primitive role R, we need the set $R^{\mathcal{I}+}$ that is given by

$$\begin{aligned} R^{\mathcal{I}+} = &\{(e,f) \mid (e,f) \in R^{\mathcal{I}},\ \exists b\ b^{\mathcal{I}} = f,\ \exists a\ a^{\mathcal{I}} = e \wedge (a,b) : R \in \mathcal{A}\} \\ &\cup \{(e,\widehat{f}) \mid (e,f) \in R^{\mathcal{I}},\ \exists b\ b^{\mathcal{I}} = f,\ \nexists a\ a^{\mathcal{I}} = e \wedge (a,b) : R \in \mathcal{A}\} \\ &\cup \{(e,f) \mid (e,f) \in R^{\mathcal{I}},\ \nexists b\ b^{\mathcal{I}} = f\}. \end{aligned}$$

Then

- $$\begin{aligned} R^{\mathcal{I}'} = R^{\mathcal{I}+} &\cup \{(\widehat{e},\widehat{f}) \mid (e,f) \in R^{\mathcal{I}+},\ \exists b\ b^{\mathcal{I}} = f\} \\ &\cup \{(\widehat{e},f) \mid (e,f) \in R^{\mathcal{I}+},\ \nexists b\ b^{\mathcal{I}} = f\} \end{aligned}$$

for every primitive role R.

Thus, in order to achieve that a copy $\widehat{d}$ behaves exactly like its original d with respect to concept instanceship we put $\widehat{d}$ exactly into the primitive concepts d belongs to; also we introduce appropriate role successors for $\widehat{d}$. Of course, d is replaced by $\widehat{d}$ in those tuples contained in $R^{\mathcal{I}}$ which contain d as second component and which violate the condition of a simple model. As a consequence, the straight interpretation $\mathcal{I}'$ is a simple interpretation.

Suppose that an interpretation $\mathcal{I}$ is a model for an ABox $\mathcal{A}$. Before we can prove that the straight interpretation $\mathcal{I}'$ for $\mathcal{I}$ is also a model for $\mathcal{A}$, we need the following lemma:

Lemma 14. Let $\mathcal{I}'$ be the straight interpretation for $\mathcal{I}$. Then:

1. If $d \in \Delta^{\mathcal{I}}$ then, for all concepts C, $d \in C^{\mathcal{I}}$ implies $d \in C^{\mathcal{I}'}$.
2. If $\widehat{d} \in \mathcal{O}$ then, for all concepts C, $d \in C^{\mathcal{I}}$ implies $\widehat{d} \in C^{\mathcal{I}'}$.

Proof. We prove the statements by induction on the size of C.

Base cases:

1. Assume that $d \in C^{\mathcal{I}}$ for a primitive concept C. Since $C^{\mathcal{I}} \subseteq C^{\mathcal{I}'}$ we have $d \in C^{\mathcal{I}'}$. If $C = \neg A$ for some primitive concept A, we know that $d \in \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$ and hence $d \notin A^{\mathcal{I}}$. Since $d \notin \mathcal{O}$ we have $d \notin A^{\mathcal{I}} \cup \mathcal{O}$. Thus we can conclude that $d \in \Delta^{\mathcal{I}'} \setminus (A^{\mathcal{I}} \cup \mathcal{O}) \subseteq \Delta^{\mathcal{I}'} \setminus A^{\mathcal{I}'}$, which shows that $d \in (\neg A)^{\mathcal{I}'}$.

2. Assume that $\widehat{d} \in \mathcal{O}$. If $d \in C^{\mathcal{I}}$ for some primitive concept C then $\widehat{d} \in C^{\mathcal{I}'}$. Now assume that $d \in C^{\mathcal{I}} = (\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$ for some primitive concept A. Thus $d \notin A^{\mathcal{I}}$, which means that $\widehat{d} \notin A^{\mathcal{I}'}$. Therefore $\widehat{d} \in \Delta^{\mathcal{I}'} \setminus A^{\mathcal{I}'} = (\neg A)^{\mathcal{I}'}$.

Induction step for 1.

If $C = C_1 \sqcap C_2$ then $d \in C_1^{\mathcal{I}}$ and $d \in C_2^{\mathcal{I}}$. By induction we conclude that $d \in C_1^{\mathcal{I}'}$ and $d \in C_2^{\mathcal{I}'}$, which shows that $d \in (C_1 \sqcap C_2)^{\mathcal{I}'}$. The case $C = C_1 \sqcup C_2$ can be shown similarly.

Now suppose that $C = (\geqslant n\ R\ D)$. There exist distinct elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$. We show that there is for each d_i an element d_i' ($d_i' \neq d_j'$ if $i \neq j$) such that $(d, d_i') \in R^{\mathcal{I}'}$ and $d_i' \in D^{\mathcal{I}'}$. If $(d, d_i) \in R^{\mathcal{I}'}$ for some i we set $d_i = d_i'$. By induction by can conclude that $d_i' \in D^{\mathcal{I}'}$. Now assume that $(d, d_i) \notin R^{\mathcal{I}'}$. Then $(d, \widehat{d_i}) \in R^{\mathcal{I}^+}$, and thus $(d, \widehat{d_i}) \in R^{\mathcal{I}'}$. With 2. it follows that $\widehat{d_i} \in D^{\mathcal{I}'}$ if $d_i \in D^{\mathcal{I}}$.

Thus it remains to be shown that $d_i' \neq d_j'$ if $i \neq j$. Suppose to the contrary that $d_i' = d_j'$ for some i, j, $i \neq j$. First suppose that $d_i' = d_i$ and $d_j' = d_j$. But this means that $d_i = d_j$ for $i \neq j$, thus contradicting our assumption that $d_1, \ldots, d_n$ are distinct elements. Secondly, suppose that $d_i' \neq d_i$ and $d_j' \neq d_j$. Thus $d_i' = \widehat{d_i}$ and $d_j' = \widehat{d_j}$. Hence, $d_i = d_j$ for $i \neq j$, which again yields a contradiction. Finally, suppose that $d_i' \neq d_i$ and $d_j' = d_j$ (the symmetric case $d_j' \neq d_j$ and $d_i' = d_i$ can be treated similarly). Since $d_i' \neq d_i$ we know that $d_i' = \widehat{d_i}$, which shows that $d_i' \notin \Delta^{\mathcal{I}}$. Since $d_j' \in \Delta^{\mathcal{I}}$, we also have a contradiction.

Finally let $C = (\leqslant n\ R\ D)$. We show that for each d_i' such that $(d, d_i') \in R^{\mathcal{I}'}$ and $d_i' \in D^{\mathcal{I}'}$, there is some d_i such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$, where $d_i \neq d_j$ if $d_i' \neq d_j'$. Since $d \in (\leqslant n\ R\ D)^{\mathcal{I}}$ it then follows immediately that $d \in (\leqslant n\ R\ D)^{\mathcal{I}'}$. Thus assume that $(d, d_i') \in R^{\mathcal{I}'}$ and $d_i' \in D^{\mathcal{I}'}$. If $(d, d_i') \in R^{\mathcal{I}}$ we set $d_i = d_i'$. Since $d_i' \in D^{\mathcal{I}'}$ we can conclude (by induction and using the contrapositive) that $d_i \in D^{\mathcal{I}}$. Now assume that $(d, d_i') \notin R^{\mathcal{I}}$. Then $d_i' = \widehat{e}$ for some $\widehat{e} \in \mathcal{O}$, and we set $d_i = e$. With 2. it follows that, for all concepts E, $e^{\mathcal{I}} \in E^{\mathcal{I}}$ implies $\widehat{e} \in E^{\mathcal{I}'}$. Since $\widehat{e} \in D^{\mathcal{I}'}$ this also shows (using the contrapositive) that $e \in D^{\mathcal{I}}$.

Similarly to the previous case, one can show that $d_j \neq d_i$ if $d_i' \neq d_j'$, which completes the proof.

Induction step for 2.

Assume that $\widehat{d} \in \mathcal{O}$. If $d \in C^{\mathcal{I}} = (C_1 \sqcap C_2)^{\mathcal{I}}$ then $d \in C_1^{\mathcal{I}}$ and $d \in C_2^{\mathcal{I}}$. By induction we conclude that $\widehat{d} \in C_1^{\mathcal{I}'}$ and $\widehat{d} \in C_2^{\mathcal{I}'}$, which shows that $\widehat{d} \in (C_1 \sqcap C_2)^{\mathcal{I}'}$. The case $C = C_1 \sqcup C_2$ can be shown similarly.

Next suppose that $C = (\geqslant n\ R\ D)$. There exist distinct elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$. By definition of $R^{\mathcal{I}'}$, there are distinct elements $d_1', \ldots, d_n'$ in $\Delta^{\mathcal{I}'}$ such that $(\widehat{d}, d_i') \in R^{\mathcal{I}'}$. Thus it remains to be shown that

$d_i' \in D^{\mathcal{I}'}$. If $d_i' = d_i$ then we can conclude with 1. that $d_i' \in D^{\mathcal{I}'}$. Thus suppose that $d_i' \neq d_i$. This means that $d_i' = \widehat{d_i}$, and we can conclude by induction that $d_i' \in D^{\mathcal{I}'}$.

The case $C = (\leqslant n\ R\ D)$ can be treated similarly. □

Proposition 15. Let $\mathcal{I}$ be a model for an ABox $\mathcal{A}$. The straight interpretation $\mathcal{I}'$ for $\mathcal{I}$ is a model for $\mathcal{A}$.

Proof. Let $\mathcal{I}$ be a model for $\mathcal{A}$, and let $\mathcal{I}'$ be the straight interpretation for $\mathcal{I}$. To show that $\mathcal{I}'$ is a model for $\mathcal{A}$ let α be an assertion in $\mathcal{A}$.

Case 1. α is of the form $a : C$ for some a in $Ind(\mathcal{A})$ and some concept C. Since $\mathcal{I}$ is a model for $\mathcal{A}$ we have $a^{\mathcal{I}} \in C^{\mathcal{I}}$. By Lemma 14 we can conclude that $a^{\mathcal{I}} = a^{\mathcal{I}'} \in C^{\mathcal{I}'}$. Thus $\mathcal{I}'$ satisfies $a : C$.

Case 2. α is of the form $(a, b) : R$ for some $a, b \in Ind(\mathcal{A})$ and some role R. Then $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Since $(a, b) : R \in \mathcal{A}$, we know that $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in R^{\mathcal{I}'}$ (definition of straight interpretation). Thus $\mathcal{I}'$ satisfies $(a, b) : R$.

Both cases together show that $\mathcal{I}'$ is a model for $\mathcal{A}$. □

Thus we can prove the following theorem:

Theorem 16. An ABox has a model if and only if it has a simple model.

Proof. One direction is trivial, since a simple model is in particular a model. In order to show the other direction let $\mathcal{I}$ be a model for an ABox $\mathcal{A}$ that is not simple. The straight interpretation $\mathcal{I}'$ for $\mathcal{I}$ is a model for $\mathcal{A}$ (cf. Proposition 15). Since $\mathcal{I}'$ is a simple interpretation, we have proved the claim. □

Let $\mathcal{A}$ be a preprocessing complete ABox that is consistent. By the previous theorem we know that $\mathcal{A}$ has a simple model. But this means that we can apply Proposition 11 which shows that, for every $a \in Ind(\mathcal{A})$, the ABox $\mathcal{A}_a$ is consistent. Thus we have established the main result of this section.

Corollary 17. Let $\mathcal{A}$ be a preprocessing complete ABox that is consistent. Then, for each a in $Ind(\mathcal{A})$, the ABox $\mathcal{A}_a$ is consistent.

6. Completeness

To prove completeness of the transformation, i.e., a preprocessing complete clash-free ABox $\mathcal{A}$ is consistent if, for every a in $Ind(\mathcal{A})$, the ABox $\mathcal{A}_a$ is consistent, we proceed as follows. To show that $\mathcal{A}$ is consistent, we construct a model for $\mathcal{A}$ out of the simple models for the ABoxes $\mathcal{A}_a$. This idea leads to the following definition.

Definition 18 (composed interpretation). Let $\mathcal{A}$ be a preprocessing complete clash-free ABox. Furthermore, let $\mathcal{I}_a$ be simple models for the ABoxes $\mathcal{A}_a$ ($a \in Ind(\mathcal{A})$)

such that $a^{\mathcal{I}_a} = a$, and $\Delta^{\mathcal{I}_a} \cap \Delta^{\mathcal{I}_b} = \varnothing$ for all a, b, $b \neq a$. The *composed interpretation* $\mathcal{I}$ for $\mathcal{A}$ is defined by:

- $\Delta^{\mathcal{I}} := \bigcup_{a \in Ind(\mathcal{A})} \Delta^{\mathcal{I}_a}$,
- $A^{\mathcal{I}} := \bigcup_{a \in Ind(\mathcal{A})} A^{\mathcal{I}_a}$ where A is a primitive concept,
- $R^{\mathcal{I}} := \bigcup_{a \in Ind(\mathcal{A})} R^{\mathcal{I}_a} \cup \{(a, b) \mid (a, b) : R \in \mathcal{A}\}$ where R is a role,
- $a^{\mathcal{I}} := a^{\mathcal{I}_a} = a$ where $a \in Ind(\mathcal{A})$.

Note that $\Delta^{\mathcal{I}_a} \cap Ind(\mathcal{A}) = \{a\}$ for every a in $Ind(\mathcal{A})$. Before we can prove that the composed interpretation $\mathcal{I}$ is a model for $\mathcal{A}$ (Proposition 20) we need the following lemma. It shows that the elements of $\Delta^{\mathcal{I}_a}$ different from a behave similarly in $\mathcal{I}_a$ and $\mathcal{I}$ with respect to concept membership.

Lemma 19. Let $\mathcal{I}$ be the composed interpretation for a preprocessing complete clash-free ABox $\mathcal{A}$. Furthermore, let a be an individual of $Ind(\mathcal{A})$. If $d \in \Delta^{\mathcal{I}_a}$, $d \neq a$, then $d \in C^{\mathcal{I}_a}$ implies $d \in C^{\mathcal{I}}$ for every concept C.

Proof. Assume that $d \neq a$. We prove by induction on the structure of the concept C that $d \in C^{\mathcal{I}_a}$ implies $d \in C^{\mathcal{I}}$. Without loss of generality we assume that C is in negation normal form. First assume that C is a primitive concept. Since $C^{\mathcal{I}_a} \subseteq C^{\mathcal{I}}$ we immediately get $d \in C^{\mathcal{I}}$.

Now let $C = \neg A$ for a primitive concept A. Then $d \in \Delta^{\mathcal{I}_a} \setminus A^{\mathcal{I}_a}$ and hence $d \notin A^{\mathcal{I}_a}$. Since $\Delta^{\mathcal{I}_a} \cap \Delta^{\mathcal{I}_b} = \varnothing$ for all b, $b \neq a$, we have $d \notin \Delta^{\mathcal{I}_b}$, and therefore $d \notin A^{\mathcal{I}_b}$ for $a \neq b$. But this means that $d \notin \bigcup_{c \in Ind(\mathcal{A})} A^{\mathcal{I}_c} = A^{\mathcal{I}}$, and thus $d \in \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} = (\neg A)^{\mathcal{I}} = C^{\mathcal{I}}$.

If $C = C_1 \sqcap C_2$ then $d \in C_1^{\mathcal{I}_a}$ and $d \in C_2^{\mathcal{I}_a}$. By induction we can conclude that $d \in C_1^{\mathcal{I}}$ and $d \in C_2^{\mathcal{I}}$, and therefore $d \in (C_1 \sqcap C_2)^{\mathcal{I}}$. The case $C = C_1 \sqcup C_2$ can be shown similarly.

Now suppose that $C = (\geqslant n\ R\ D)$. There exist distinct elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}_a}$ such that $(d, d_i) \in R^{\mathcal{I}_a}$ and $d_i \in D^{\mathcal{I}_a}$. Since $R^{\mathcal{I}_a} \subseteq R^{\mathcal{I}}$ we have $(d, d_i) \in R^{\mathcal{I}}$. We observe that $d_i \neq a$ (because $\mathcal{I}_a$ is a simple model for $\mathcal{A}_a$). Thus we can conclude by induction that $d_i \in D^{\mathcal{I}}$. Therefore we have $d \in (\geqslant n\ R\ D)^{\mathcal{I}}$.

For $C = (\leqslant n\ R\ D)$ we have to show that there are at most n elements $d_1, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$. Therefore we verify that $(d, d_i) \in R^{\mathcal{I}_a}$ and $d_i \in D^{\mathcal{I}_a}$ if $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$. If $(d, d_i) \in R^{\mathcal{I}}$ then $(d, d_i) \in R^{\mathcal{I}_a}$ because $d \in \Delta^{\mathcal{I}_a}$, $d \neq a$, and $\Delta^{\mathcal{I}_a} \cap \Delta^{\mathcal{I}_b} = \varnothing$ for $a \neq b$. To see that $d_i \in D^{\mathcal{I}_a}$ suppose to the contrary that $d_i \in D^{\mathcal{I}}$ and $d_i \notin D^{\mathcal{I}_a}$. Then $d_i \in \Delta^{\mathcal{I}_a} \setminus D^{\mathcal{I}_a} = (NNF(\neg D))^{\mathcal{I}_a}$. By induction we conclude that $d_i \in (NNF(\neg D))^{\mathcal{I}}$ which contradicts the assumption $d_i \in D^{\mathcal{I}}$. Thus we have shown that for each d_i such that $(d, d_i) \in R^{\mathcal{I}}$ and $d_i \in D^{\mathcal{I}}$ we also have $(d, d_i) \in R^{\mathcal{I}_a}$ and $d_i \in D^{\mathcal{I}_a}$. Since $d \in (\leqslant n\ R\ D)^{\mathcal{I}_a}$ we can therefore conclude that $d \in (\leqslant n\ R\ D)^{\mathcal{I}}$. □

Now we are able to prove the completeness of the transformation.

Proposition 20. Let $\mathcal{I}$ be the composed interpretation for a preprocessing complete clash-free ABox $\mathcal{A}$. Then $\mathcal{I}$ satisfies $\mathcal{A}$.

Proof. To prove the claim we show that the interpretation $\mathcal{I}$ satisfies every assertion α in $\mathcal{A}$. If α is of the form $(a,b):R$ then $\mathcal{I}$ satisfies α by definition of $\mathcal{I}$. Now assume that α is of the form $a:C$. We prove by induction on the structure of C that $\mathcal{I}$ satisfies α. First assume that C is a primitive concept. By definition of $\mathcal{A}_a$ we know that $a:C$ is contained in $\mathcal{A}_a$. Since $\mathcal{I}_a$ is a model for $\mathcal{A}_a$ it follows that $a \in C^{\mathcal{I}_a}$. Since $C^{\mathcal{I}_a} \subseteq C^{\mathcal{I}}$ we can conclude that $a \in C^{\mathcal{I}}$.

If $C = \neg A$ for a primitive concept A, we have $a \in \Delta^{\mathcal{I}_a} \setminus A^{\mathcal{I}_a}$ (because $a:\neg A \in \mathcal{A}_a$), and therefore $a \notin A^{\mathcal{I}_a}$. Since $\Delta^{\mathcal{I}_a} \cap \Delta^{\mathcal{I}_b} = \varnothing$ for all b, $b \neq a$, it follows that $a \notin \Delta^{\mathcal{I}_b}$, and therefore $a \notin A^{\mathcal{I}_b}$ for all b, $b \neq a$. Thus $a \notin \bigcup_{c \in Ind(\mathcal{A})} A^{\mathcal{I}_c} = A^{\mathcal{I}}$, which means that $a \in \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} = (\neg A)^{\mathcal{I}} = C^{\mathcal{I}}$.

Assume that $C = C_1 \sqcap C_2$. Since the $\rightarrow_\sqcap$-rule does not apply to $\mathcal{A}$ (because $\mathcal{A}$ is preprocessing complete) we know that $a:C_1$ and $a:C_2$ are in $\mathcal{A}$. By induction we conclude that $\mathcal{I}$ satisfies $a:C_1$ and $a:C_2$, and therefore the assertion $a:(C_1 \sqcap C_2)$ is satisfied by $\mathcal{I}$. The case $C = C_1 \sqcup C_2$ can be shown similarly.

Now let $C = (\geqslant n\ R\ D)$. For $m = n_{R,D,\mathcal{A}}(a)$ the assertion $a:(\geqslant (n-m)\ R\ D)$ is contained in $\mathcal{A}_a$. Since $\mathcal{I}_a$ satisfies $\mathcal{A}_a$ there exist distinct elements $d_1, \ldots, d_{n-m}$ in $\Delta^{\mathcal{I}_a}$ such that $(a,d_i) \in R^{\mathcal{I}_a}$ and $d_i \in D^{\mathcal{I}_a}$. But this means that $(a,d_i) \in R^{\mathcal{I}}$ (definition of $\mathcal{I}$) and $d_i \in D^{\mathcal{I}}$ (Lemma 19). Thus $|\{d' \in \Delta^{\mathcal{I}_a} \mid (d,d') \in R^{\mathcal{I}} \text{ and } d' \in D^{\mathcal{I}}\}| \geqslant n-m$. Now consider $b \in Ind(\mathcal{A})$ such that $(a,b):R$ and $b:D$ are in $\mathcal{A}$. Since $b:D \in \mathcal{A}$ we can conclude by induction that $b \in D^{\mathcal{I}}$. Furthermore, $(a,b) \in R^{\mathcal{I}}$, which follows from the definition of $\mathcal{I}$. But this means that $|\{b \in Ind(\mathcal{A}) \mid (a,b) \in R^{\mathcal{I}} \text{ and } b \in D^{\mathcal{I}}\}| \geqslant m$. Therefore $|\{d \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \text{ and } b \in D^{\mathcal{I}}\}| \geqslant n$, which shows that $a \in (\geqslant n\ R\ D)^{\mathcal{I}}$.

Finally assume that $C = (\leqslant n\ R\ D)$. If $m = n_{R,D,\mathcal{A}}(a)$, we have $n \geqslant m$ because $\mathcal{A}$ is clash-free. There are individuals $b_1, \ldots, b_m$ in $Ind(\mathcal{A})$ such that $(a,b_i) \in R^{\mathcal{I}}$ (definition of $\mathcal{I}$) and $b_i \in D^{\mathcal{I}}$ (by induction). We show that there are at most $n-m$ elements $e_1, \ldots, e_{n-m}$ different from b_j (for all j, $1 \leqslant j \leqslant m$) such that $(a,e_i) \in R^{\mathcal{I}}$ and $e_i \in D^{\mathcal{I}}$. Let e_i be such an element.

We first observe that $e_i \neq c$ for all individuals c in $Ind(\mathcal{A})$. Suppose to the contrary that $e_i = c$ for some $c \in Ind(\mathcal{A})$. This means that $(a,e_i):R \in \mathcal{A}$. Since $\mathcal{A}$ is preprocessing complete either $e_i:D$ or $e_i:NNF(\neg D)$ is in $\mathcal{A}$. If $e_i:D \in \mathcal{A}$ then $e_i = b_j$ for some j, which contradicts our assumption that $e_i \neq b_j$ for all j, $1 \leqslant j \leqslant m$. If $e_i:NNF(\neg D) \in \mathcal{A}$ we can conclude by induction that $e_i \notin D^{\mathcal{I}}$ contradicting the assumption that $e_i \in D^{\mathcal{I}}$.

Thus we have shown that $e_i \neq c$ for all $c \in Ind(\mathcal{A})$. Hence we know that $e_i \in \Delta^{\mathcal{I}_a}$. Therefore $(a,e_i) \in R^{\mathcal{I}_a}$ and, as can easily be verified with Lemma 19, $e_i \in D^{\mathcal{I}_a}$. Since $a \in (\leqslant (n-m)\ R\ D)^{\mathcal{I}_a}$ (because $a:(\leqslant (n-m)\ R\ D)$ is contained in $\mathcal{A}_a$), there exist at most $n-m$ elements $e_1, \ldots, e_{n-m}$ different from b_j (for all j, $1 \leqslant j \leqslant m$) such that $(a,e_i) \in R^{\mathcal{I}}$ and $e_i \in D^{\mathcal{I}}$. Thus we can conclude that $a \in (\leqslant n\ R\ D)^{\mathcal{I}}$. □

This result together with the results from the previous section yields a reduction from the consistency problem of ABoxes to the satisfiability problem of concepts. This establishes the main result of the paper.

Theorem 21. The consistency problem of ABoxes can be reduced to the satisfiability problem of concepts using polynomial space in the size of the ABox.

Proof. Let $\mathcal{A}$ be an ABox. First assume that $\mathcal{A}$ is consistent. By Proposition 6 there is a preprocessing complete ABox $\mathcal{A}'$ such that (1) $\mathcal{A}'$ is consistent and (2) $\mathcal{A}'$ is derived from $\mathcal{A}$ by application of preprocessing rules. But this means that, for every $a \in Ind(\mathcal{A})$, the ABox $\mathcal{A}'_a$ is consistent (Corollary 17). Since $\mathcal{A}'_a = \{a:C_1, \ldots, a:C_n\}$ for some concepts $C_1, \ldots, C_n$, we know that $\mathcal{A}'_a$ is consistent if and only if the concept $C_1 \sqcap \cdots \sqcap C_n$ is satisfiable.

Now assume that $\mathcal{A}$ is inconsistent. Then every preprocessing complete ABox $\mathcal{A}'$ issuing from $\mathcal{A}$ is inconsistent (Proposition 6). We now show that $\mathcal{A}'$ either contains a clash or there is some $a \in Ind(\mathcal{A})$ such that $\mathcal{A}'_a$ is inconsistent. Suppose to the contrary that $\mathcal{A}'$ is clash-free and $\mathcal{A}'_a$ is consistent for every $a \in Ind(\mathcal{A})$. By Theorem 16 we know that $\mathcal{A}'_a$ has a simple model. But this means that $\mathcal{A}'$ is consistent (Proposition 20), which is not possible.

Since all preprocessing complete ABoxes issuing from $\mathcal{A}$ can be enumerated using polynomial space in the size of $\mathcal{A}$ this yields the claim. □

7. Terminologies

When modeling the relevant notions of a problem domain with the help of a concept language it is convenient to refer to already introduced concepts, rather than defining every concept from scratch by using primitive concepts and roles only. Almost all terminological systems therefore allow their users to introduce abbreviations for concepts and roles. For example, the concept definition mother $\doteq$ woman $\sqcap$ $\exists$child.person introduces the concept name mother as an abbreviation for the concept occurring on the right hand side of the definition. Using this abbreviation the concept parent can simply be defined by parent $\doteq$ mother $\sqcup$ father if father is appropriately defined. Analogously, we may introduce abbreviations for roles as, for instance, daughter $\doteq$ child | female. The so-called *terminology* (for short *TBox*) is given by a set of such concept and role definitions together with statements which introduce the primitive concepts and roles.

Almost all terminological systems do not allow arbitrary terminologies, but only those that satisfy the following two additional conditions: Every primitive concept and primitive role may appear at most once at the left hand side of a definition, i.e., every concept and role has a unique definition. And secondly, the terminology must not contain cyclic definitions, i.e., the right hand side of a concept (role) definition must not refer directly or indirectly to the concept (role) name to be defined.

It is a well-known fact that an ABox that refers to a terminology that satisfies both conditions can be transformed into an "equivalent" *expanded ABox* that does no longer refer to the terminology (see, e.g., [14]). The idea behind the transformation is to enlarge the ABox by the facts expressed in the terminology. That is, if $A \doteq C$ is a concept definition of the terminology, and $a : D$ is a concept assertion in the ABox such that A occurs in D, we substitute each occurrence of A in D by C. This process is iterated until the rewrite rule just described is no longer applicable. Obviously, this process terminates if the concept and role definitions do not contain any cycle.

An expanded ABox thus obtained may be exponential in the size of the terminology as mentioned in [14]. But this might imply that the consistency problem of an ABox w.r.t. a terminology has a higher complexity than the consistency problem of an ABox. In fact, Nebel [15] has proved that for the restricted concept language $\mathcal{FL}^-$ (which includes concept conjunction, value-restriction, and the restricted exists-restriction $\exists R.\top$) subsumption becomes co-NP-hard if the concepts are given by a terminology, while subsumption between a pair of $\mathcal{FL}^-$-concepts can be checked in polynomial time [12]. For the rather expressive concept language $\mathcal{ALCQ}$ of the present paper this however is no longer true. When successively expanding concept definitions during the consistency check as described in [9, Chapter 5] (rather than starting with the completely expanded concept definitions), one can show that the consistency problem of an ABox w.r.t. a terminology can be decided with polynomial space. But this means that this inference problem is not harder than the consistency problem of ABoxes in $\mathcal{ALCQ}$.

8. Conclusion

We have presented a reduction from the consistency problem of ABoxes to the satisfiability problem of concepts for the concept language $\mathcal{ALCQ}$. This result has two important applications. From a practical point of view, our reduction together with the existence of relatively efficient implementations of satisfiability algorithms strongly simplifies the implementation of consistency and instance checking algorithms. Even from a complexity point of view, the result shows that consistency checking is not harder (in terms of the worst case complexity) than satisfiability checking in our concept language.

Acknowledgements

I would like to thank Franz Baader, Armin Laux, and Jörg Siekmann for helpful comments on an earlier version of this paper. In particular, Franz' remarks helped me to simplify notations.

This work has been supported by the German Ministry for Research and Technology (BMFT) under research contracts ITW 8903 0 and ITW 9201.

References

[1] F. Baader and B. Hollunder, *KRIS*: Knowledge Representation and Inference System, *SIGART Bulletin* 2(3) (1991) 8–14.

[2] F. Baader and B. Hollunder, A terminological knowledge representation system with complete inference algorithms, in: *International Workshop on Processing Declarative Knowledge*, Vol. 567, eds. M. Richter and H. Boley (Springer, 1991).

[3] F. Baader and B. Hollunder, Embedding defaults into terminological knowledge representation formalisms, in: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Mass. (1992).

[4] R.J. Brachman and J.G. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive Science* 9(2) (1985) 171–216.

[5] F. Donini, B. Hollunder, M. Lenzerini, A.M. Spaccamela, D. Nardi and W. Nutt, The complexity of existential quantification in concept languages, *Artificial Intelligence* 2–3 (1992) 309–327.

[6] F. Donini, M. Lenzerini, D. Nardi and W. Nutt, The complexity of concept languages, in: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Mass. (1991).

[7] F. Donini, M. Lenzerini, D. Nardi and W. Nutt, Tractable concept languages, in: *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia (1991).

[8] F.M. Donini, M. Lenzerini, D. Nardi and A. Schaerf, From subsumption to instance checking, Technical Report 15.92, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza" (1992).

[9] B. Hollunder, *Algorithmic Foundations of Terminological Knowledge Representation Systems*, PhD Thesis, University of Saarbrücken (1994).

[10] B. Hollunder and F. Baader, Qualifying number restrictions in concept languages, in: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, Mass. (1991).

[11] B. Hollunder, W. Nutt and M. Schmidt-Schauß, Subsumption algorithms for concept description languages, in: *Proceedings of the 9th European Conference on Artificial Intelligence*, Stockholm, Sweden (1990) pp. 348–353.

[12] H.J. Levesque and R.J. Brachman, Expressiveness and tractability in knowledge representation and reasoning, *Computational Intelligence* 3 (1987) 78–93.

[13] R. MacGregor, Inside the LOOM description classifier, *SIGART Bulletin* 2(3) (1991) 88–92.

[14] B. Nebel, *Reasoning and Revision in Hybrid Representation Systems*, LNAI 422 (Springer-Verlag, Berlin, Germany, 1990).

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

[16] P.F. Patel-Schneider, D.L. McGuinness, R.J. Brachman, L.A. Resnick and A. Borgida, The CLASSIC knowledge representation system: Guiding principles and implementation rational, *SIGART Bulletin* 2(3) (1991) 108–113.

[17] C. Peltason, The BACK system – an overview, *SIGART Bulletin* 2(3) (1991) 114–119.

[18] M. Schmidt-Schauß and G. Smolka, Attributive concept descriptions with complements, *Artificial Intelligence* 47 (1991).