

Cardinality Restrictions on Concepts

Franz Baader

Lehr- und Forschungsgebiet Theoretische Informatik, RWTH Aachen

Ahornstraße 55, 52074 Aachen, Germany

e-mail: baader@informatik.rwth-aachen.de

Martin Buchheit

Deutsches Forschungszentrum für KI – DFKI GmbH

Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany

e-mail: buchheit@dfki.uni-sb.de

Bernhard Hollunder

iO Software GmbH

Victoria Haus, Basler Straße 63, 79100 Freiburg, Germany

e-mail: hollunder@io.freinet.de

Abstract

The concept description formalisms of existing description logics systems allow the user to express local cardinality restrictions on the fillers of a particular role. It is not possible, however, to introduce global restrictions on the number of instances of a given concept. This article argues that such cardinality restrictions on concepts are of importance in applications such as configuration of technical systems, an application domain of description logics systems that is currently gaining in interest. It shows that including such restrictions in the description language leaves the important inference problems such as instance testing decidable. The algorithm combines and simplifies the ideas developed for the treatment of qualified number restrictions and of general terminological axioms.

1 Introduction

Description logics (DL) systems can be used to represent the conceptual and taxonomic knowledge of an application domain in a structured and semantically well-understood way. To describe this kind of knowledge one starts with atomic concepts

(unary predicates) and roles (binary predicates), and employs the concept description formalism provided by the system to define more complex concepts. In addition to this terminological component (TBox), most systems also have an assertional component (ABox), in which concepts and roles can be instantiated by individual names (constant symbols) representing particular elements of the problem domain.

The reasoning services of DL systems allow the user to retrieve not only the knowledge that is explicitly stored in TBox and ABox, but to access implicitly represented knowledge as well. For a given TBox, the system automatically computes the concept hierarchy according to subconcept-superconcept relationships (subsumption relationships) induced by the structure of the concepts. In addition, it can determine the consistency of the knowledge base (consisting of a TBox and an ABox), and it answers queries regarding the existence of instance relationships between individuals and concepts.

To make these inference services feasible, the description formalism of a DL system must be of limited expressive power. On the other hand, a too severely restricted formalism may turn out to be too weak for certain applications. For this reason, several extensions of “core” concept languages have been investigated in the literature (see, e.g., [3, 11, 1]). In the present paper, we shall consider an extension that is motivated by the use of DL systems for solving configuration tasks, which is an application domain that is currently gaining more and more importance (see, e.g., [15, 5, 12, 21, 9, 10, 13, 19, 20]). Technical domains such as configuration seem to be well suited for DL systems since they usually rely on a large number of terminological conventions, which are in most cases precisely defined. In contrast, more traditional AI applications of DL systems, such as natural language processing [7], often rely on vague notions and incomplete knowledge, which require the representation of beliefs, as well as probabilistic and default information.

Unlike these very demanding, and not yet well-understood extensions of concept description languages, the additional language construct we shall introduce in this paper is more or less along the lines of traditional constructs, albeit of a rather expressive and thus algorithmically hard to handle nature. It allows one to express restrictions on the number of elements a concept may have: $(\geq m C)$ and $(\leq n C)$ respectively express that the (possibly complex) concept C has at least m elements and at most n elements, thus restricting the possible models of the knowledge base.

The traditional language constructs that most closely resemble this new one are the so-called number restrictions, which are present in almost all existing systems. Number restrictions allow one to specify the number of possible role-fillers of a particular role. Such a restriction can, for example, express that an admissible PC may have at most 17 parts, by restricting the number of role-fillers of the **has-part** role to less or equal 17. If one allows for qualified number restrictions [11] (which are not available in most systems), one can also express that the PC must have exactly one CPU and at most four 1MB memory chips, where CPU and 1MB-memory-chip may

be complex concepts. These cardinality restrictions are, however, still localized to the fillers of one particular role. In contrast, the cardinality restrictions on concepts we propose here are global in the sense that they restrict the number of objects belonging to a given concept for the whole domain of interest (e.g., the whole technical system that is configured). For example, one can express that (in a computer) there must be exactly one electrical power supply unit, which supplies all the devices with electrical power. With a conventional concept description language, even one including qualified number restrictions, one can only express that every device must have a power supply, but not that all must have the same (or one out of a specified number n).¹

The expressive power of the new construct is also demonstrated by the fact that it can be used to express terminological axioms of the form $C \doteq D$ (see Section 2 below), which express that the (possibly complex) concepts C and D have exactly the same instances. Such axioms are known to be algorithmically hard to handle (satisfiability is EXP-TIME hard) [17].

In a very restricted setting, cardinality restrictions have been considered by van der Hoek and de Rijke [18]. However, their language is far less expressive than ours (satisfiability is in PSPACE). Like number restrictions and several other concept constructors, cardinality restrictions on concepts can be seen as specific generalized quantifiers [6]. In [16], Quantz proposes an integration of various generalized quantifiers into terminological formalisms in order to cope with problems related to bound anaphora resolution. However, like most of the research on generalized quantifiers in the area of linguistics and philosophical logics, Quantz’s article is only concerned with expressibility issues and not with computability. In fact, for his representation language the important inference problems turn out to be undecidable. The main goal of the present paper is to design sound and complete inference algorithms for the language we propose.

In the following, we shall first formally introduce the terminological formalism considered in this paper, which contains both cardinality restrictions on concepts and qualified number restrictions. Section 2 also defines the relevant reasoning services for terminological knowledge bases consisting of a terminological and an assertional component. In Section 3 we shortly sketch how these services can be utilized in a configuration application. Then we shall develop an algorithm that tests a knowledge base for consistency. This is sufficient since all the other interesting inference services can easily be reduced to this task [4]. The consistency algorithm combines the ideas developed in [2, 8] for handling inclusion axioms (in a language with number restrictions), and in [11] for handling qualified number restrictions.

¹Qualified number restrictions could be used to express cardinality restrictions on concepts if the description formalism contained a symbol for the “top-role” (i.e., the universal role that connects all objects with each other). However, none of the implemented systems allows for such a top-role symbol, and reasoning in a concept language where the top-role and ordinary roles coexist has—to the best of our knowledge—not yet been investigated.

2 The Terminological Formalism

The expressive power of a DL system is determined by the constructs available for building concept descriptions, and by the way these descriptions can be used in the terminological (TBox) and the assertional (ABox) component of the system. The description language \mathcal{ALCQ} defined below coincides with the one introduced in [11]. The new expressivity lies in the TBox, where the usual terminological axioms are replaced by cardinality restrictions on concepts. The assertional component is the standard one.

The description language The *concept descriptions* (for short, concepts) of the language \mathcal{ALCQ} are built from *concept names* and *role names* using the constructors *conjunction* ($C \sqcap D$), *disjunction* ($C \sqcup D$), *negation* ($\neg C$), and *qualified number restrictions* ($(\geq n R C)$ and $(\leq n R C)$), where C, D stand for concepts, R for a role name, and n for a nonnegative integer.

Note that (unqualified) number restrictions, value restrictions ($\forall R.C$) and existential restrictions ($\exists R.C$) are not explicitly included in the language since they can all be expressed with the help of qualified number restrictions.

To define the semantics of concept descriptions, we interpret concepts as subsets of a domain of interest and roles as binary relations over this domain. More precisely, an *interpretation* \mathcal{I} 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 concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and every role name R to a subset $R^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The interpretation function is extended to arbitrary concept descriptions as follows. Let C, D be concept descriptions, R be a role name, n be a nonnegative integer, and assume that $C^{\mathcal{I}}$ and $D^{\mathcal{I}}$ are already defined. Then

$$\begin{aligned} (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, & (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, & (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\ (\geq n R C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \geq n\}, \\ (\leq n R C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \leq n\}, \end{aligned}$$

where $\#X$ denotes the cardinality of a set X .

The terminological component A *terminological axiom* is an expression of the form $C \doteq D$, where C and D are (possibly complex) concept descriptions. A finite set of such axioms is called a *TBox*. The semantics of a TBox is quite obvious: an interpretation \mathcal{I} *satisfies* an axiom $C \doteq D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$, and it is a *model* of a TBox \mathcal{T} iff it satisfies all axioms in \mathcal{T} .

Most systems impose severe restrictions on admissible TBoxes: (1) The concepts on the left-hand sides of axioms must be concept names, (2) concept names occur at most once as left-hand side of an axiom, and (3) there are no cyclic definitions. The

effect of these restrictions is that terminological axioms are just macro definitions (introducing names for large descriptions), which can simply be expanded before starting the reasoning process. Unrestricted terminological axioms are a lot harder to handle algorithmically [17, 2, 8], but they are very useful in expressing important constraints on admissible configurations (see Section 3 below).

Now we introduce a new type of axiom, which we call cardinality restrictions on concepts, and which are even more expressive than unrestricted terminological axioms of the form $C \doteq D$. Such a *cardinality restriction* is an expression of the form $(\geq n C)$ or $(\leq n C)$, where C is a concept description and n a nonnegative integer. An interpretation \mathcal{I} *satisfies* the restriction $(\geq n C)$ iff $\sharp C^{\mathcal{I}} \geq n$ and $(\leq n C)$ iff $\sharp C^{\mathcal{I}} \leq n$.

Obviously, saying that C and D have the same instances is equivalent to stating that the concept $(C \sqcap \neg D) \sqcup (\neg C \sqcap D)$ is empty, i.e., contains at most zero elements. This demonstrates that terminological axioms can be expressed by cardinality restrictions. For this reason, a *TBox* will from now on simply be a finite set of cardinality restrictions. The interpretation \mathcal{I} is a *model* of such a TBox iff it satisfies each of its restrictions.

The assertional component In this component, facts concerning particular objects in the application domain can be expressed as follows. The objects are referred to by *individual names*, and these names may be used in two types of *assertional axioms*: concept assertions $C(a)$ and role assertions $R(a, b)$, where C is a concept description, R is a role name, and a, b are individual names. A finite set of assertions is called *ABox*.

In order to give a semantics to assertions we extend the interpretation function to individuals. Each individual name a is interpreted as an element $a^{\mathcal{I}}$ of the domain such that the mapping from individual names to $\Delta^{\mathcal{I}}$ is 1-1. This restriction is usually called unique name assumption (UNA). The interpretation \mathcal{I} *satisfies* the assertion $C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and the assertion $R(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. We say that an interpretation \mathcal{I} is a *model* of an ABox \mathcal{A} iff \mathcal{I} satisfies every assertion in \mathcal{A} .

The reasoning services A terminological *knowledge base* (KB) $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ consists of an ABox \mathcal{A} and a TBox \mathcal{T} . After representing the relevant knowledge of an application domain in such a KB, one can not just retrieve the information that is explicitly stored. DL systems also provide their users with services that allow to access knowledge that is only implicitly represented in the KB. For example, these reasoning services provide answers to the following queries:

1. *KB-consistency*: Is the given KB consistent? That is, does there exist a model of the KB (i.e., a model of both the ABox and the TBox)?

2. *Concept Satisfiability*: Given a KB and a concept C , does there exist a model of the KB in which C is interpreted as a nonempty set?
3. *Subsumption*: Given a KB and two concepts C and D , does $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ hold for all models \mathcal{I} of the KB? Subsumption detects implicit dependencies among the concepts in the KB.
4. *Instance Checking*: Given a KB, an individual a and a concept C , does $a^{\mathcal{I}} \in C^{\mathcal{I}}$ hold for any model \mathcal{I} of the KB?

Concept satisfiability, subsumption of concepts, and instance checking can be reduced to KB-consistency (or to its complement) in linear time. To make this possible, we must introduce an additional syntactic category in the ABox: individuals that are *not* subjected to the UNA.² The only difference between this new type of individuals (called new individuals) and the usual ones (called old individuals) is that the interpretation function must be a 1–1-mapping only on the old individuals. This means that different new individuals may be interpreted by the same object, and a new individual may also have the same interpretation as an old one.

Proposition 2.1 *Let C, D be concepts, $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ a knowledge base, a an individual, and let o be a new individual, i.e., an individual that does not appear in Σ and is not subjected to the UNA. Then:*

- C is satisfiable iff $\langle \mathcal{A} \cup \{C(o)\}, \mathcal{T} \rangle$ is consistent,
- C is subsumed by D iff $\langle \mathcal{A} \cup \{(C \sqcap \neg D)(o)\}, \mathcal{T} \rangle$ is not consistent,
- a is instance of C iff $\langle \mathcal{A} \cup \{(\neg C)(a)\}, \mathcal{T} \rangle$ is not consistent.

Consequently, it is sufficient to devise an algorithm for the KB-consistency problem.

Note that omitting the UNA for the (new) individual o is crucial for the soundness of the reductions. For example, assume that $\mathcal{T}_1 = \{(\leq 1 C)\}$ and $\mathcal{A}_1 = \{C(a)\}$, where C is a concept name. Obviously, the concept C is satisfiable with respect to the KB $\langle \mathcal{A}_1, \mathcal{T}_1 \rangle$, but the KB $\langle \mathcal{A}_1 \cup \{C(o)\}, \mathcal{T}_1 \rangle$ would not be consistent if o was subjected to the UNA. In fact, the only way to satisfy $\langle \mathcal{A}_1 \cup \{C(o)\}, \mathcal{T}_1 \rangle$ is to interpret o and a by the same object.

Also note that in the reductions given above, the ABox cannot be omitted when considering subsumption and satisfiability of concepts, and the TBox cannot be expanded. This is in contrast to most description logic formalisms, where the following two properties are satisfied: (1) one can get rid of the concept definitions in the TBox by an expansion process, and (2) the ABox does not influence concept satisfiability

²The consistency algorithm presented in Section 4 will also make use of individuals not subjected to the UNA.

and subsumption (see [14]). Due to the presence of cardinality restrictions, this is no longer true for our formalism. Indeed, the (standard) expansion procedure cannot be applied in the presence of unrestricted terminological axioms, and thus property (1) is no longer satisfied. To see that property (2) is violated as well, we extend the ABox \mathcal{A}_1 by the assertion $D(a)$. Because of the cardinality restriction in \mathcal{T}_1 , a is the only element of C . In addition, the new ABox says that a belongs to D , which shows that in all models of \mathcal{T}_1 and the new ABox, C is interpreted as a subset of D . This yields an additional subsumption relationship between C and D , which does not hold with respect to the TBox alone.

3 Application in Configuration

Before describing a consistency algorithm for KBs with cardinality restrictions, we give some ideas of how such an algorithm can be employed to solve configuration tasks. Of course, in order to build a complete configurator based on this approach, further features like truth maintenance, control strategies, interactivity, and an explanation component have to be integrated.

Figure 1 contains some parts of the description of a SPARCstation 2 in our terminological formalism. The first three axioms of the TBox are traditional concept definitions, which (in a top-down manner) introduce names for complex descriptions. A SPARCstation 2 is defined to have four obligatory parts, namely system unit, monitor, keyboard, and mouse and pad. In addition, it may have as optional parts terminals and printers, but no other parts are admissible. The concepts standing for the parts are again defined by descriptions. In the example, we have just given the (simplified) descriptions of the system unit, and of the main logic board, which is a part of this unit. Note that $(= n R C)$ is an abbreviation for $(\geq n R C) \sqcap (\leq n R C)$.

The next five axioms are inclusion axioms of the form $C \sqsubseteq D$, which should be read as abbreviations of the corresponding cardinality restrictions $(\leq 0 C \sqcap \neg D)$. The (complex) concepts main logic board, hard drive, and diskette drive are required to have a power supply, and certain types of terminals need specific cables. The qualified number restrictions in these inclusion axioms express that each part has exactly one power supply, but different parts can still have different power supplies. The last terminological axiom, which is a cardinality restriction on the concept power supply, makes sure that all parts use the same power supply. It seems to be impossible to express such a constraint in a traditional terminological formalism unless one allows for role-value maps (which would, however, cause undecidability).

Configuration checking The instance test of a DL system can be employed to check whether a computer configuration is admissible. This idea has, for example,

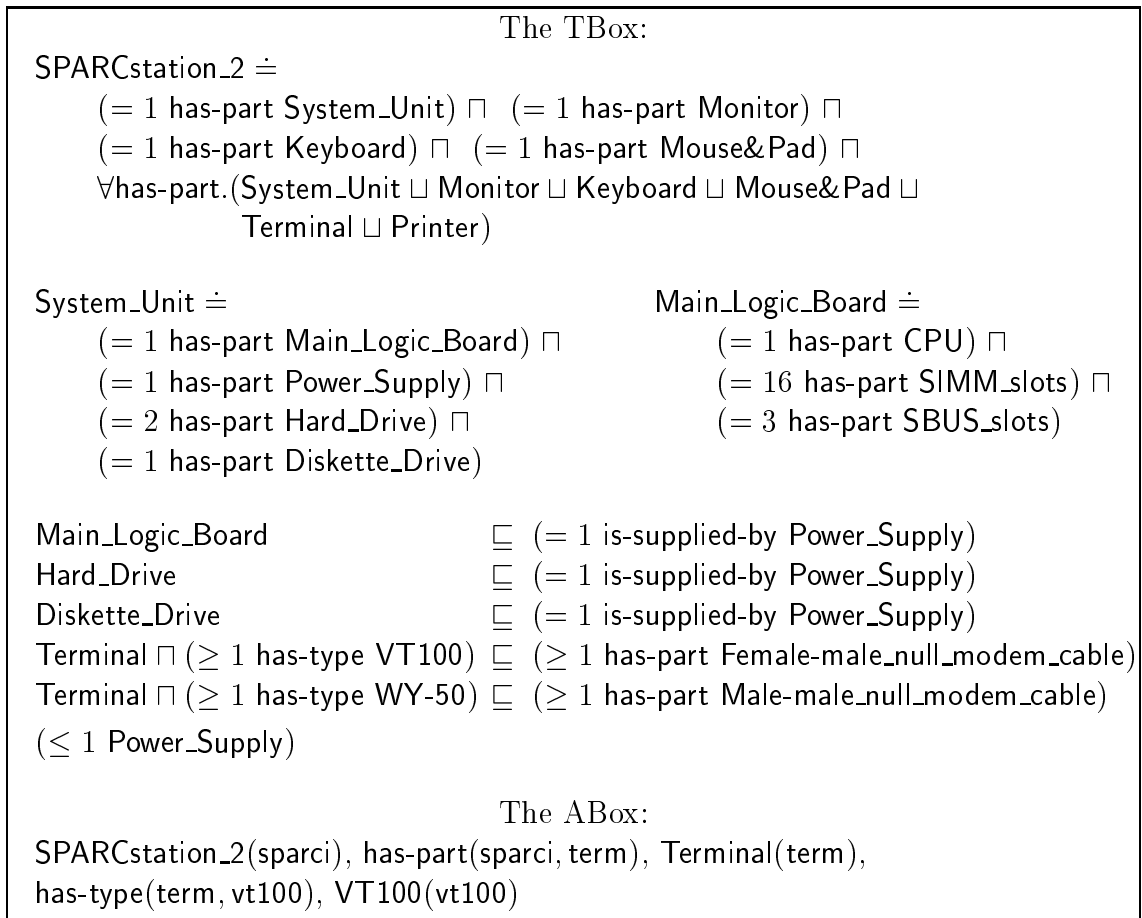


Figure 1: A SPARCstation 2

been used in an application of the DL-system CLASSIC [21].³ In the TBox, one defines a concept that describes admissible computer systems, and in the ABox one describes the actual configuration of a computer system. The instance test then checks whether the individual corresponding to the configuration is an instance of the concept “admissible computer system.” The description of the actual configuration can be done on different levels of abstraction. For example, we can describe a SPARCstation 2 by saying that it has four fillers of the *has-part* role that are respectively in the concepts *System_Unit*, *Monitor*, *Keyboard*, and *Mouse&Pad*. On a lower level of abstraction, the realization that the parts belong to these concepts is also left to the instance test.

In addition, one can also define concepts that describe the most frequent errors made when configuring such a system (e.g., forgetting some cables). When the instance test finds out that the configuration belongs to such an error concept

³One should note, however, that this application is not restricted to simple configuration checking, and that, beside the instance test, the firing of CLASSIC rules—which are not considered here—plays an important role.

then one knows the reason why the configuration was not admissible, and can take appropriate action.

Configuration generation The configuration domain is again modeled in the TBox, and the ABox contains a (high level) description of what should be configured. The consistency algorithm we shall describe below has the property that it not only answers with “consistent” or “inconsistent.” If the KB is consistent, it also yields a finite model (see the definition of the canonical model in Section 4), in which all the implicit information contained in the TBox and ABox is made explicit. In principle, this model describes an admissible configuration.

In Figure 1, the ABox describes that we want to have a SPARCstation 2 with an additional VT100 terminal. If we invoke the consistency algorithm of Section 4, it will generate the obligatory parts like system unit, etc. It also makes sure that the integrity constraints expressed by the inclusion axioms and the cardinality restriction are satisfied (more information on this idea of configuration by model generation can be found in [9, 10]).

4 The Consistency Algorithm

The method for deciding consistency of a KB presented below is rule-based in the sense that it starts with the original KB (consisting of an ABox \mathcal{A}_0 and a TBox \mathcal{T}_0), and applies certain consistency preserving transformation rules to the ABox until no more rules apply. If the “complete” KB thus obtained contains an obvious contradiction (called clash) then the original KB $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ was inconsistent. Otherwise, $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ was consistent since the complete KB can be used to construct a finite model.

The transformation rule that handles number restrictions of the form ($\geq n R C$) will generate n new ABox individuals x_1, \dots, x_n that stand for the role-fillers required by the restriction. Recall that unlike the individuals present in the original ABox (the old individuals), the new individual names should not be subjected to the unique name assumption. In fact, in a model they may well be interpreted identical to an old individual or a new individual introduced by another rule application. What must be ensured, however, is that x_1, \dots, x_n are interpreted by different objects. In order to express this we need a new type of assertion, called inequality assertion. Such an assertion is of the form $s \neq t$ for individuals s, t , and it has the obvious semantics, i.e., an interpretation \mathcal{I} satisfies $s \neq t$ iff $s^{\mathcal{I}} \neq t^{\mathcal{I}}$. These assertions are considered as being symmetric, i.e., saying that $s \neq t \in \mathcal{A}$ is the same as saying that $t \neq s \in \mathcal{A}$.

In the following, we assume that the set of individual names is partitioned into a set I_{old} of old individual names (subjected to the UNA) and a set I_{new} of new

individual names. The elements of I_{old} are just the individuals present in the original ABox,⁴ which means that I_{old} is finite. We assume that I_{new} is infinite to allow for an arbitrary number of rule applications. We denote individuals of I_{old} by the letters a, b , of I_{new} by x, y , and of $I = I_{\text{old}} \cup I_{\text{new}}$ by s, t (all possibly with subscript).

The transformation rule that handles disjunction (as well as the rules concerned with at-most restrictions) is *nondeterministic* in the sense that a given ABox is transformed into two (or finitely many) new ABoxes such that the original ABox is consistent with the TBox iff *one of* the new ABoxes is so. For this reason we will consider *generalized* KBs of the form $\langle \mathcal{M}, \mathcal{T} \rangle$, where $\mathcal{M} = \{\mathcal{A}_1, \dots, \mathcal{A}_l\}$ is a finite set of ABoxes. This generalized KB is called *consistent* iff there is some i , $1 \leq i \leq l$, such that $\langle \mathcal{A}_i, \mathcal{T} \rangle$ is consistent.

Treatment of cardinality restrictions So far, all that has been said also applies to rule-based consistency algorithms for less expressive languages (see, e.g., [4]). Cardinality restrictions give rise to two new problems.

To see the *first problem*, assume that the TBox contains the restriction ($\leq n C$), and that all individuals contained in the ABox are either asserted to be in C or in its complement. If the number m of individuals in C is larger than n then we know that we must take action, whereas $m \leq n$ shows that no action is required. In general, however, the ABox will also contain individuals for which no assertions relating them to C or $\neg C$ are present. For these individuals, we do not know a priori whether a model of the TBox and ABox will interpret them as elements of C or of $\neg C$. Thus we are not necessarily able to decide whether action is required or not.

To make sure that in the end all such indeterminate situations are resolved, we introduce a rule (called choose-rule below) that makes sure that at some stage of the transformation process each individual will either be asserted to be in C or its complement. (The choice is “don’t know” nondeterministic, i.e., both cases have to be considered.) In a slightly modified way the idea of such a choose-rule was already presented in [11], since qualified number-restrictions of the form ($\leq n R C$) cause a similar problem.

The *second problem* is that, due to the choose-rule, the transformation process need no longer terminate, unless one takes specific precautions to detect cyclic computations. In fact, if the concept C from above is of the form ($\geq m R D$) (for $m \geq 1$), then asserting C for an individual s_0 causes the introduction of a new individual s_1 . Because of the choose-rule, at some stage of the transformation we must consider an ABox where s_1 is asserted to be in C , which causes the introduction of a new individual s_2 , etc.

⁴Note that the individual o that might have been introduced when reducing the satisfiability or subsumption problem to the consistency problem is assumed to be a new individual (see Proposition 2.1).

In order to regain the termination property, we restrict the applicability of transformation rules that generate new individuals. The idea is that the application of such rules is blocked for a new individual x if there is another individual s in the ABox that has all concept assertions that x has. Termination is then due to the fact that there are only finitely many different concepts D that can occur in such assertions. To prevent cyclic blocking, which would destroy the correctness of the algorithm, we consider an enumeration t_0, t_1, t_2, \dots of I in which all elements of I_{old} come before all elements of I_{new} . We write $t < t'$ iff t comes before t' in this enumeration. Now blocking can formally be defined as follows:

Definition 4.1 *An individual $x \in I_{\text{new}}$ is blocked by an individual $s \in I$ in an ABox \mathcal{A} iff $\{D \mid D(x) \in \mathcal{A}\} \subseteq \{D' \mid D'(s) \in \mathcal{A}\}$ and $s < x$.*

Note that only new individuals can be blocked.

Similar termination problems are already caused by terminological axioms of the form $C \doteq D$. For this reason, the idea of blocking is already present in [8]. The main difference between the two notions of blocking is that in [8] equality of sets is required whereas we are satisfied with set inclusion. It turns out that our notion of blocking facilitates the termination proof. In addition, termination can be shown for arbitrary sequences of rule applications, and no longer depends on the use of a specific strategy (as required in [8]).

Preprocessing In order to facilitate the description of the transformation rules, we start with a preprocessing step that transforms the original KB into a simplified form.

As usual, all concepts occurring in the KB are transformed into negation normal form, where negation occurs only immediately in front of concept names. Negation normal forms can be computed in linear time by pushing negation signs into the descriptions (see, e.g., [4]). The expression $\sim C$ will denote the negation normal form of the concept $\neg C$.

In addition, we assume that the TBox contains only restrictions of the form $(\leq n C)$. In fact, a restriction $(\geq n C)$ can be expressed in the ABox by adding assertions $C(x_i)$ and $x_i \neq x_j$ (for $1 \leq i, j \leq n, i \neq j$), where the x_i are new individuals that did not occur in the original KB.

Finally, the UNA for old individuals is made explicit in the ABox by adding the assertions $a \neq b$ for each pair of distinct elements $a, b \in I_{\text{old}}$.

The transformation rules As a result of the preprocessing step, the input of the consistency algorithm is a generalized KB $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$ where \mathcal{A}_0 and \mathcal{T}_0 are in the simplified form described above. Starting with $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$, the algorithm applies the *transformation rules* of Figure 2 as long as possible.

The \rightarrow_{\sqcap} -rule

Condition: \mathcal{A} contains $(C_1 \sqcap C_2)(s)$, but it does not contain both $C_1(s)$ and $C_2(s)$.

Action: $\mathcal{A}' := \mathcal{A} \cup \{C_1(s), C_2(s)\}$

The \rightarrow_{\sqcup} -rule

Condition: \mathcal{A} contains $(C_1 \sqcup C_2)(s)$, but neither $C_1(s)$ nor $C_2(s)$.

Action: $\mathcal{A}' := \mathcal{A} \cup \{C_1(s)\}$, $\mathcal{A}'' := \mathcal{A} \cup \{C_2(s)\}$

The \rightarrow_{\geq} -rule

Condition: \mathcal{A} contains $(\geq n R C)(s)$, s is not blocked in \mathcal{A} , and there are no individual names s_1, \dots, s_n such that $R(s, s_i)$, $C(s_i)$, and $s_i \neq s_j$ ($1 \leq i, j \leq n, i \neq j$) are contained in \mathcal{A} .

Action: $\mathcal{A}' := \mathcal{A} \cup \{R(s, x_i), C(x_i) \mid 1 \leq i \leq n\} \cup \{x_i \neq x_j \mid 1 \leq i, j \leq n, i \neq j\}$, where $x_1, \dots, x_n \in I_{\text{new}}$ are distinct individuals such that $x_i > s'$ for all individual names s' occurring in \mathcal{A} .

The $\rightarrow_{\text{choose}}$ -rule

Condition: \mathcal{A} contains an individual t such that either

1. $(\leq n R C)(s)$ and $R(s, t)$ are in \mathcal{A} , or
2. $(\leq n C)$ is in \mathcal{T}_0 ,

and \mathcal{A} does not contain $(C \sqcup \sim C)(t)$.

Action: $\mathcal{A}' := \mathcal{A} \cup \{(C \sqcup \sim C)(t)\}$

The \rightarrow_{\leq} -rule

Condition: \mathcal{A} contains distinct individuals t_1, \dots, t_{n+1} such that either

1. $(\leq n R C)(s)$ and $R(s, t_1), \dots, R(s, t_{n+1})$ are in \mathcal{A} , or
2. $(\leq n C)$ is in \mathcal{T}_0 ,

and $C(t_1), \dots, C(t_{n+1})$ are in \mathcal{A} , and $t_i \neq t_j$ is not in \mathcal{A} for some $i \neq j$.

Action: For each pair t_i, t_j such that $t_j < t_i$ and $t_i \neq t_j$ is not in \mathcal{A} the ABox $\mathcal{A}_{i,j} := [t_i/t_j]\mathcal{A}$ is obtained from \mathcal{A} by replacing each occurrence of t_i by t_j .

Figure 2: Completion rules of the consistency algorithm.

The rules should be read as follows. They are applied to a generalized KB $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ (where \mathcal{M} is a set of ABoxes). The rules take an element \mathcal{A} of \mathcal{M} , and

replace it by one ABox \mathcal{A}' , by two ABoxes \mathcal{A}' and \mathcal{A}'' , or by finitely many ABoxes $\mathcal{A}_{i,j}$. The TBox \mathcal{T}_0 of the input is left unchanged.

The transformation rules are *sound* in the sense that if there is a model for the TBox \mathcal{T}_0 and the ABox \mathcal{A} , then there is also a model for \mathcal{T}_0 and one of the ABoxes \mathcal{A} is replaced by. Thus, if $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ is obtained from $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$ by a sequence of rule applications then $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ is consistent if $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ is consistent (see Section 5 for the proof).

The second important property of the set of transformation rules is that the transformation process always terminates, i.e., there cannot be an infinite sequence of rule application (see Section 5 for the proof). Thus, after finitely many transformation steps we obtain a generalized KB to which no more rules apply. We call such a generalized KB *complete*. Consistency of a complete (generalized) KB $\langle \{\mathcal{A}_1, \dots, \mathcal{A}_n\}, \mathcal{T}_0 \rangle$ can be decided by looking for obvious contradictions, so-called clashes, in the KBs $\langle \mathcal{A}_i, \mathcal{T}_0 \rangle$.

Definition 4.2 *A KB $\langle \mathcal{A}, \mathcal{T} \rangle$ contains a clash iff one of the following three situations occurs:*

1. $\{B(s), \neg B(s)\} \subseteq \mathcal{A}$ for some individual s and some concept name B .
2. $\{(\leq n R C)(s), R(s, t_i), C(t_i), t_i \neq t_j \mid 1 \leq i, j \leq n+1, i \neq j\} \subseteq \mathcal{A}$ for individuals s, t_1, \dots, t_{n+1} , a nonnegative integer n , a concept C , and a role name R .
3. $(\leq n C) \in \mathcal{T}$ and $\{C(s_i), s_i \neq s_j \mid 1 \leq i, j \leq n+1, i \neq j\} \subseteq \mathcal{A}$ for individuals s_1, \dots, s_{n+1} , a nonnegative integer n , and a concept C .

Obviously, a KB that contains a clash cannot be consistent. Consequently, if all KBs $\langle \mathcal{A}_i, \mathcal{T}_0 \rangle$ contain a clash, then $\langle \{\mathcal{A}_1, \dots, \mathcal{A}_n\}, \mathcal{T}_0 \rangle$ is inconsistent, which by soundness of the rules implies that the original KB $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ was inconsistent.

If, however, one of the KBs $\Sigma_i = \langle \mathcal{A}_i, \mathcal{T}_0 \rangle$ is clash-free then the corresponding canonical interpretation \mathcal{I}_{Σ_i} (as defined below) can be used to construct a model of the original KB $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ (see Section 5 for the proof).

Definition 4.3 *Let $\Sigma = \langle \mathcal{A}, \mathcal{T} \rangle$ be a KB. The canonical interpretation \mathcal{I}_Σ induced by Σ is defined as follows:*

- The domain $\Delta^{\mathcal{I}_\Sigma}$ of \mathcal{I}_Σ consists of all the individuals occurring in \mathcal{A} .
- For all concept names A we define $A^{\mathcal{I}_\Sigma} = \{s \mid A(s) \in \mathcal{A}\}$.
- For a role name R we define $R^{\mathcal{I}_\Sigma}$ inductively with respect to the total ordering $<$ on the individual names. If s_0 is the least element in $\Delta^{\mathcal{I}_\Sigma}$ then $(s_0, t) \in R^{\mathcal{I}_\Sigma}$ iff $R(s_0, t) \in \mathcal{A}$. Now let $s \in \Delta^{\mathcal{I}_\Sigma}$ be different from s_0 .

- If s is not blocked in \mathcal{A} then we define $(s, t) \in R^{\mathcal{I}_\Sigma}$ iff $R(s, t) \in \mathcal{A}$.
 - If s is blocked in \mathcal{A} then let s' be the least (with respect to the ordering $<$) individual name in $\Delta^{\mathcal{I}_\Sigma}$ that blocks s . By the definition of blocking, $s' < s$, and thus we can assume that the set $\{t \mid (s', t) \in R^{\mathcal{I}_\Sigma}\}$ is already defined, and we define $(s, t) \in R^{\mathcal{I}_\Sigma}$ iff $(s', t) \in R^{\mathcal{I}_\Sigma}$.
- For an individual s occurring in \mathcal{A} we set $s^{\mathcal{I}_\Sigma} := s$.

To sum up, we have seen that the transformation rules of Figure 2 reduce consistency of a KB $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ to consistency of a complete generalized KB $\langle \mathcal{M}, \mathcal{T}_0 \rangle$. In addition, consistency of this complete KB can be decided by looking for obvious contradictions (clashes). This shows the main result of the paper:

Theorem 4.4 *It is decidable whether or not a KB $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ is consistent.*

5 Proof of Correctness

To prove Theorem 4.4, we first show that the transformation rules are sound and terminating. Then we show that the canonical interpretation of a complete and clash-free KB can be used to construct a model of the original KB.

5.1 Soundness of the rules

Proposition 5.1 *Assume that the generalized KB $\langle \mathcal{M}', \mathcal{T}_0 \rangle$ is obtained from $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ by application of a transformation rule. If $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ is consistent then $\langle \mathcal{M}', \mathcal{T}_0 \rangle$ is consistent.*

Proof. In the following, we restrict our attention to the \rightarrow_{\geq} -rule and the \rightarrow_{\leq} -rule. The other rules can be treated similarly.

(1) Assume that the \rightarrow_{\geq} -rule is applied to the ABox \mathcal{A} in \mathcal{M} , and that \mathcal{M}' is obtained from \mathcal{M} by replacing \mathcal{A} by \mathcal{A}' . Thus \mathcal{A} contains an assertion $(\geq n R C)(s)$, and \mathcal{A}' is obtained from \mathcal{A} by adding $R(s, x_i)$, $C(x_i)$, and $x_i \neq x_j$ ($1 \leq i, j \leq n, i \neq j$), where $x_1, \dots, x_n \in I_{\text{new}}$ are such that $x_i > s'$ for all individual names s' occurring in \mathcal{A} . It is sufficient to show that $\langle \mathcal{A}', \mathcal{T}_0 \rangle$ is consistent if $\langle \mathcal{A}, \mathcal{T}_0 \rangle$ is consistent.

Thus, let \mathcal{I} be a model of \mathcal{A} and \mathcal{T}_0 . Since the new individual names x_1, \dots, x_n do not occur in \mathcal{A} , validity of assertions in \mathcal{A} does not depend on the interpretation of these names. Because \mathcal{A} contains $(\geq n R C)(s)$, we know that $s^{\mathcal{I}} \in (\geq n R C)^{\mathcal{I}}$. Thus there exist distinct elements d_1, \dots, d_n of $\Delta^{\mathcal{I}}$ with $(s^{\mathcal{I}}, d_i) \in R^{\mathcal{I}}$ and $d_i \in C^{\mathcal{I}}$ ($1 \leq i \leq n$). Obviously, if we modify \mathcal{I} to an interpretation \mathcal{I}' by interpreting the new individuals x_1, \dots, x_n as $x_i^{\mathcal{I}'} = d_i$, then \mathcal{I}' is a model of \mathcal{A}' and \mathcal{T}_0 .

(2) Assume that the \rightarrow_{\leq} -rule is applied to the ABox \mathcal{A} in \mathcal{M} . We restrict our attention to the case where \mathcal{A} contains assertions $C(t_1), \dots, C(t_{n+1})$ and \mathcal{T}_0 contains the cardinality restriction $(\leq n C)$ (the case of the qualified number restrictions can be treated analogously). \mathcal{M}' is obtained from \mathcal{M} by replacing \mathcal{A} by the finitely many ABoxes $\mathcal{A}_{i,j} := [t_i/t_j]\mathcal{A}$ (for $t_j < t_i$ and $t_i \neq t_j$ not in \mathcal{A}).

Now, let \mathcal{I} be a model of \mathcal{A} and \mathcal{T}_0 . Since $(\leq n C) \in \mathcal{T}_0$, we know that $C^{\mathcal{I}}$ contains at most n elements. Thus there exist indices i, j ($1 \leq i, j \leq n+1, i \neq j$) such that $t_i^{\mathcal{I}} = t_j^{\mathcal{I}}$. Without loss of generality we assume that $t_j < t_i$. Since \mathcal{I} is a model of \mathcal{A} , the assertion $t_i \neq t_j$ cannot be contained in \mathcal{A} , which implies that $[t_i/t_j]\mathcal{A}$ is an element of \mathcal{M}' . Obviously, \mathcal{I} is also a model of $[t_i/t_j]\mathcal{A}$ and \mathcal{T}_0 . \square

5.2 Termination

Proposition 5.2 *Let $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ be a finite KB. Then any sequence of rule applications starting with $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$ is finite.*

Before we can prove the proposition we have to introduce some notation. If a transformation rule replaces the ABox \mathcal{A} by $\mathcal{A}_1, \dots, \mathcal{A}_n$, we write $\mathcal{A} \rightarrow \mathcal{A}_i$ (for all i with $1 \leq i \leq n$). In order to express which rule has been applied, the arrow is equipped with the appropriate subscript; e.g., $\mathcal{A} \rightarrow_{\leq} \mathcal{A}_i$ means that the \rightarrow_{\leq} -rule has been applied.

For an ABox \mathcal{A} and an individual name s , we define

$$CA(s, \mathcal{A}) := \{C \mid C(s) \text{ is a concept assertion in } \mathcal{A}\}.$$

Obviously, the new individual x is blocked by s in \mathcal{A} iff $s < x$ and $CA(x, \mathcal{A}) \subseteq CA(s, \mathcal{A})$.

The following facts are an easy consequence of the way the transformation rules and CA are defined:

- Lemma 5.3**
1. *Let $\mathcal{A} \rightarrow \mathcal{A}'$, and let s be an individual in \mathcal{A} that is not replaced in \mathcal{A}' . Then we have $CA(s, \mathcal{A}) \subseteq CA(s, \mathcal{A}')$. If t_i is an individual in \mathcal{A} that is replaced by t_j in \mathcal{A}' then we have $t_j < t_i$, $CA(t_i, \mathcal{A}) \subseteq CA(t_j, \mathcal{A}')$ and $CA(t_j, \mathcal{A}) \subseteq CA(t_j, \mathcal{A}')$.*
 2. *Let $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ be a finite KB, let $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ be obtained from this KB by a finite number of applications of transformation rules, and let \mathcal{A} be an element of \mathcal{M} . For all concept assertions $C(s) \in \mathcal{A}$, the concept description C is a subdescription of a description $D \sqcup \sim D$, where D occurs in $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$.*
 3. *The second fact shows that there are only finitely many different sets $CA(s, \mathcal{A})$ for a given sequence of transformations starting from a finite KB.*

To prove Proposition 5.2 we assume to the contrary that there exists an infinite sequence of rule applications, which yields the KBs $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$, $\langle \mathcal{M}_1, \mathcal{T}_0 \rangle$, $\langle \mathcal{M}_2, \mathcal{T}_0 \rangle$, \dots . Since a transformation rule replaces one ABox by only finitely many new ABoxes, König's lemma implies that there is an infinite sequence of ABoxes $\mathcal{A}_1, \mathcal{A}_2, \dots$ such that $\mathcal{A}_0 \rightarrow \mathcal{A}_1 \rightarrow \mathcal{A}_2 \rightarrow \dots$.

For any individual s occurring in these ABoxes, there are only finitely many different concept assertions possible. Each rule application adds concept assertions on an individual or removes an individual. Thus, to have an infinite sequence of rule applications, infinitely many individuals must be generated, which means that the \rightarrow_{\geq} -rule must have been applied infinitely often. In addition, to a fixed individual s , the \rightarrow_{\geq} -rule cannot be applied infinitely many times. Indeed, s can occur in only finitely many different at-least assertions, and to each at-least assertion, the \rightarrow_{\geq} -rule is applied at most once. This shows that there must be infinitely many individual s_1, s_2, s_3, \dots to which the \rightarrow_{\geq} -rule was applied. Since, for any individual name s , there are only finitely many smaller individual names, we may without loss of generality assume that $s_1 < s_2 < s_3 < \dots$, and since I_{old} is finite we may assume that all these individuals are new individuals, i.e., elements of I_{new} .

For all i , let $\mathcal{A}_{j_i} \rightarrow_{\geq} \mathcal{A}_{j_{i+1}}$ be the transformation step at which the \rightarrow_{\geq} -rule is applied to s_i . Now consider the sets $CA(s_i, \mathcal{A}_{j_i})$. Since there are only finitely many different such sets, there must be indices $k < l$ such that $CA(s_k, \mathcal{A}_{j_k}) = CA(s_l, \mathcal{A}_{j_l})$. If s_k is still present in \mathcal{A}_{j_l} (i.e., it has not been replaced by an application of the \rightarrow_{\leq} -rule), then $CA(s_l, \mathcal{A}_{j_l}) = CA(s_k, \mathcal{A}_{j_k}) \subseteq CA(s_k, \mathcal{A}_{j_l})$. Since $s_k < s_l$ and s_l is a new individual, this means that s_l should be blocked in \mathcal{A}_{j_l} , which is a contradiction to our assumption that the \rightarrow_{\geq} is applied to s_l in \mathcal{A}_{j_l} .

If s_k is no longer present in \mathcal{A}_{j_l} then it has been replaced (possibly iteratively) by another individual, say t , and we know that $t < s_k$. Since in each replacement step the replacing individual inherits all the concept assertions of the replaced individual, we know that $CA(s_k, \mathcal{A}_{j_k}) \subseteq CA(t, \mathcal{A}_{j_l})$. Again, we can conclude that s_l is blocked in \mathcal{A}_{j_l} . This completes the proof of termination. \square

5.3 Completeness

Let $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$ be a generalized KB obtained as the result of our preprocessing step. This means that \mathcal{T}_0 contains only at-most restrictions. The ABox \mathcal{A}_0 may contain concept assertions and inequality assertions both for old and new individuals, but all role assertions are of the form $R(a, b)$ for old individuals a, b .

Assume that $\langle \mathcal{M}, \mathcal{T}_0 \rangle$ is a complete generalized KB that was obtained by starting with $\langle \{\mathcal{A}_0\}, \mathcal{T}_0 \rangle$ and applying the transformation rules of Figure 2 until no more rules apply. Let $\mathcal{A} \in \mathcal{M}$ be such that $\Sigma = \langle \mathcal{A}, \mathcal{T}_0 \rangle$ does not contain a clash, and let \mathcal{I}_{Σ} be the corresponding canonical interpretation. In the following, we show that \mathcal{I}_{Σ} can

be used to construct a model of $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$.

First, note that \mathcal{I}_Σ need not be a model of $\Sigma = \langle \mathcal{A}, \mathcal{T}_0 \rangle$. The problem is that an individual s that is blocked in \mathcal{A} need not have been blocked at an earlier stage of the transformation process. Thus, at such an earlier stage, the \rightarrow_{\geq} -rule may have been applied to s , generating an individual t with $R(s, t) \in \mathcal{A}$. This role assertion need not be satisfied by the canonical interpretation (see the definition of role-fillers for blocked individuals in the definition of the canonical interpretation).

However, \mathcal{I}_Σ is a model of a certain subset of \mathcal{A} , and this will be sufficient to show the desired result.

Definition 5.4 *The set of relevant assertions of an ABox \mathcal{B} are defined as*

$$\begin{aligned} \text{rel}(\mathcal{B}) \quad := \quad & \{C(s) \mid C(s) \text{ is a concept assertion in } \mathcal{B}\} \cup \\ & \{R(a, b) \mid R(a, b) \in \mathcal{B} \text{ and } a, b \in I_{\text{old}}\} \cup \\ & \{s \neq t \mid s \neq t \text{ is an inequality assertion in } \mathcal{B}\}. \end{aligned}$$

Thus, $\text{rel}(\mathcal{B})$ is obtained from \mathcal{B} by removing all role assertions involving new individuals. Since the ABox \mathcal{A}_0 obtained by preprocessing does not contain role assertion for new individuals, we know that $\text{rel}(\mathcal{A}_0) = \mathcal{A}_0$.

Lemma 5.5 *Let $\Sigma = \langle \mathcal{A}, \mathcal{T}_0 \rangle$ be a complete and clash-free KB. Then the canonical interpretation \mathcal{I}_Σ is a model of $\langle \text{rel}(\mathcal{A}), \mathcal{T}_0 \rangle$.*

Proof. First, consider a *role assertion* $R(a, b) \in \text{rel}(\mathcal{A})$. We know that $a, b \in I_{\text{old}}$, and thus a cannot be blocked. By the definition of the canonical interpretation, $R(a, b) \in \mathcal{A}$ thus yields $(a, b) \in R^{\mathcal{I}_\Sigma}$.

Second, consider an *inequality assertion* $s_1 \neq s_2 \in \text{rel}(\mathcal{A})$. Since individual names interpret themselves in \mathcal{I}_Σ , it is sufficient to show that s_1 and s_2 cannot be identical names. Obviously, the ABox \mathcal{A}_0 obtained after the preprocessing step does not contain an inequality assertion of the form $s \neq s$, and it is easy to see that this property is invariant under rule application. In fact, the only rule that “identifies” different individual names is the \rightarrow_{\leq} -rule. But this rule is applied for individuals t_i and t_j only if $t_i \neq t_j$ is not contained in the ABox.

Third, consider a *concept assertion* $C(s) \in \text{rel}(\mathcal{A})$. We show by induction on the structure of C that $s \in C^{\mathcal{I}_\Sigma}$. Note that $C(s) \in \text{rel}(\mathcal{A})$ iff $C(s) \in \mathcal{A}$.

(1) Assume that C is a concept name B . Then $B(s) \in \mathcal{A}$ implies $s \in B^{\mathcal{I}_\Sigma}$ by definition of the canonical interpretation.

(2) Assume that C is of the form $\neg B$ for a concept name B . Since \mathcal{A} was assumed to be clash-free, we know that $B(s)$ is not contained in \mathcal{A} , and thus $s \notin B^{\mathcal{I}_\Sigma}$ by definition of the canonical interpretation.

(3) Assume that C is of the form $C_1 \sqcap C_2$ for concept descriptions C_1 and C_2 . Since the \rightarrow_{\sqcap} -rule is not applicable to \mathcal{A} , we know that \mathcal{A} contains both $C_1(s)$ and $C_2(s)$. By induction, we can deduce that $s \in C_1^{\mathcal{I}_\Sigma}$ and $s \in C_2^{\mathcal{I}_\Sigma}$, which obviously implies $s \in (C_1 \sqcap C_2)^{\mathcal{I}_\Sigma}$.

(4) The case where C is of the form $C_1 \sqcup C_2$ can be treated analogously.

(5) Assume that C is of the form $(\geq n R D)$. We have to distinguish two cases: s can be blocked or not.

(5.1) Assume that s is not blocked. Thus, since the \rightarrow_{\geq} -rule is not applicable, we know that \mathcal{A} contains individuals s_1, \dots, s_n such that the assertions $R(s, s_i)$, $D(s_i)$, and $s_i \neq s_j$ ($1 \leq i, j \leq n, i \neq j$) are in \mathcal{A} . Because s is not blocked, $R(s, s_i) \in \mathcal{A}$ implies $(s, s_i) \in R^{\mathcal{I}_\Sigma}$ (by definition of \mathcal{I}_Σ). In addition, induction yields $s_i \in D^{\mathcal{I}_\Sigma}$. Finally, since we already know that \mathcal{I}_Σ satisfies all inequality assertions in \mathcal{A} , the s_i are all different from each other. This shows that $s \in (\geq n R D)^{\mathcal{I}_\Sigma}$.

(5.2) Now, assume that s is blocked in \mathcal{A} . Let s' be the least (with respect to the ordering $<$) individual name in $\Delta^{\mathcal{I}_\Sigma}$ that blocks s .

First, we show that s' is not blocked. Otherwise, the individual s'' that blocks s' satisfies $s'' < s' < s$ and $CA(s, \mathcal{A}) \subseteq CA(s', \mathcal{A}) \subseteq CA(s'', \mathcal{A})$. Thus s'' blocks s and is smaller than s' , which is a contradiction.

Because $CA(s, \mathcal{A}) \subseteq CA(s', \mathcal{A})$ we have $(\geq n R D)(s') \in \mathcal{A}$. As shown in (5.1) this implies that there are distinct individual names s_1, \dots, s_n such that $(s', s_i) \in R^{\mathcal{I}_\Sigma}$ and $s_i \in D^{\mathcal{I}_\Sigma}$. By definition of the canonical interpretation, we also have $(s, s_i) \in R^{\mathcal{I}_\Sigma}$, which yields $s \in (\geq n R D)^{\mathcal{I}_\Sigma}$.

(6) Assume that C is of the form $(\leq n R D)$. In order to show that $s \in (\leq n R D)^{\mathcal{I}_\Sigma}$ we assume to the contrary that there exist distinct individuals s_1, \dots, s_{n+1} such that $(s, s_i) \in R^{\mathcal{I}_\Sigma}$ and $s_i \in D^{\mathcal{I}_\Sigma}$ (for $i = 1, \dots, n+1$). Again, we have to distinguish two cases, depending on whether s is blocked or not.

(6.1) Assume that s is not blocked. Then $(s, s_i) \in R^{\mathcal{I}_\Sigma}$ implies $R(s, s_i) \in \mathcal{A}$. Since the $\rightarrow_{\text{choose}}$ -rule is not applicable, we know that $(D \sqcup \sim D)(s_i)$ is in \mathcal{A} for all i , and since the \rightarrow_{\sqcup} -rule is not applicable, we have for all i that either $D(s_i)$ or $\sim D(s_i)$ is in \mathcal{A} . By induction, $\sim D(s_i) \in \mathcal{A}$ would yield $s_i \in (\sim D)^{\mathcal{I}_\Sigma} = (\neg D)^{\mathcal{I}_\Sigma}$.⁵ Thus, $s_i \in D^{\mathcal{I}_\Sigma}$ yields $D(s_i) \in \mathcal{A}$ for $i = 1, \dots, n+1$. This is a contradiction, since now either the \rightarrow_{\leq} -rule must be applicable, or \mathcal{A} must contain a clash.

(6.2) Now, assume that s is blocked in \mathcal{A} . Let s' be the least (with respect to the ordering $<$) individual name in $\Delta^{\mathcal{I}_\Sigma}$ that blocks s .

As in (5.2), we can deduce that s' is not blocked, and that \mathcal{A} contains the assertion $(\leq n R D)(s')$. In addition, $(s, s_i) \in R^{\mathcal{I}_\Sigma}$ implies $(s', s_i) \in R^{\mathcal{I}_\Sigma}$ (by definition of \mathcal{I}_Σ). Thus we can proceed as in (6.1), with s' in place of s . This

⁵We assume that the negation sign does not contribute to the size of a concept term. Thus $\sim D$ has the same size as D (cf. [11]).

completes the proof that \mathcal{I}_Σ satisfies all the concept assertions in \mathcal{A} .

Finally, consider an *element* ($\leq n$ C) of the *TBox* \mathcal{T}_0 . Assume that there are $n+1$ different individuals $s_1, \dots, s_{n+1} \in \Delta^{\mathcal{I}_\Sigma}$ such that $s_i \in C^{\mathcal{I}_\Sigma}$ for $i = 1, \dots, n+1$. As in (6.1) above, the fact that the $\rightarrow_{\text{choose}}$ -rule is not applicable can be used to show that $C(s_i) \in \mathcal{A}$. Again, this is a contradiction, since now either the \rightarrow_{\leq} -rule must be applicable, or \mathcal{A} must contain a clash. This completes the proof that \mathcal{I}_Σ is a model of $\langle \text{rel}(\mathcal{A}), \mathcal{T}_0 \rangle$. \square

Lemma 5.6 *Assume that $\mathcal{B} \rightarrow \mathcal{B}'$ and that \mathcal{I}' is a model of $\text{rel}(\mathcal{B}')$ and \mathcal{T}_0 . Then there exists an interpretation \mathcal{I} that is a model of $\text{rel}(\mathcal{B})$ and \mathcal{T}_0 .*

Proof. If \mathcal{B}' is obtained from \mathcal{B} by an application of the \rightarrow_{\sqcap} -, \rightarrow_{\sqcup} -, \rightarrow_{\geq} -, or $\rightarrow_{\text{choose}}$ -rule then \mathcal{B} is a subset of \mathcal{B}' . Thus, $\text{rel}(\mathcal{B}) \subseteq \text{rel}(\mathcal{B}')$, which shows that we can simply use $\mathcal{I} := \mathcal{I}'$.

Thus, the only interesting case is the \rightarrow_{\leq} -rule. This means that $\mathcal{B}' = [t_i/t_j]\mathcal{B}$ is obtained from \mathcal{B} by replacing each occurrence of t_i by t_j (for some individuals t_i and t_j in \mathcal{B}). We know that $t_j < t_i$ and that $t_i \neq t_j$ is not in \mathcal{B} . Since t_i has been replaced by t_j , we also know that t_i does not occur in \mathcal{B}' . Hence, if we define \mathcal{I} such that \mathcal{I} is identical to \mathcal{I}' , with the exception that $t_i^{\mathcal{I}} := t_j^{\mathcal{I}'}$, then we know that \mathcal{I} is a model of $\text{rel}(\mathcal{B}')$. It remains to be shown that \mathcal{I} is also a model of $\text{rel}(\mathcal{B})$.

Let σ be the substitution that replaces t_i by t_j , and leaves all the other individuals unchanged. Since $\mathcal{B}' = [t_i/t_j]\mathcal{B}$ we know for all individuals s in \mathcal{B} that $C(s) \in \text{rel}(\mathcal{B})$ (respectively $s \neq t \in \text{rel}(\mathcal{B})$) implies $C(\sigma(s)) \in \text{rel}(\mathcal{B}')$ (respectively $\sigma(s) \neq \sigma(t) \in \text{rel}(\mathcal{B}')$). In addition, by our definition of \mathcal{I} , we have $\sigma(s)^{\mathcal{I}} = s^{\mathcal{I}'}$. This shows that all the concept and inequality assertions in $\text{rel}(\mathcal{B})$ are satisfied by \mathcal{I} .

Finally, let $R(a, b)$ be a role assertion in $\text{rel}(\mathcal{B})$. Thus we know that a and b are old individuals. Neither of these two individuals can be equal to t_i . To show this, assume (without loss of generality) that a is equal to t_i . Since $t_j < t_i = a$, this implies that t_j is also an old individual. But then we have $t_i \neq t_j$ in \mathcal{B} (because in the preprocessing step the unique name assumption for old individuals has been made explicit). This is a contradiction since the \rightarrow_{\leq} -rule generates the ABox $[t_i/t_j]\mathcal{B}$ only if $t_i \neq t_j$ is not in \mathcal{B} .

Since a and b are different from t_i we know that $R(a, b)$ is also contained in $\text{rel}(\mathcal{B}')$, and we are done. \square

To sum up, Lemma 5.5 shows that \mathcal{I}_Σ is a model of the clash-free and complete KB $\langle \text{rel}(\mathcal{A}), \mathcal{T}_0 \rangle$. By applying Lemma 5.6 iteratively, we can deduce that $\langle \text{rel}(\mathcal{A}_0), \mathcal{T}_0 \rangle$ has a model. Since the ABox \mathcal{A}_0 obtained after preprocessing satisfies $\text{rel}(\mathcal{A}_0) = \mathcal{A}_0$, we thus know that $\langle \mathcal{A}_0, \mathcal{T}_0 \rangle$ has a model. This completes the proof of Theorem 4.4.

6 Conclusion

We have shown how to extend a terminological KR formalism by a construct that can express global restrictions on the cardinality of concepts. The usefulness of these cardinality restrictions on concepts was demonstrated by an example from a configuration application.

Because of the possibility of global restrictions, knowledge bases in our new formalism have significantly different properties from knowledge bases in more traditional terminological formalisms. We have seen that in the presence of cardinality restrictions, subsumption between concepts depends not only on the TBox, but also on the ABox. In addition, it is no longer true that unconnected objects (i.e., objects not connected by role relationships) in an ABox or a model cannot influence each other. This complicates the algorithmic treatment of the new construct.

Unlike role-value maps, however, (which could be used to model similar situations), our new construct leaves all the important inference problems decidable. As a lower complexity bound, we know that KB-consistency is EXP-TIME-hard, since cardinality restrictions on concepts can express general terminological axioms, for which EXP-TIME-hardness is known [17]. Our algorithm yields as upper bound NEXP-TIME, but it is not clear whether a deterministic exponential time algorithm can be derived.

The consistency algorithm combines and simplifies the ideas developed for the treatment of qualified number restrictions and of terminological axioms. In particular, our new definition of blocked objects allows for a termination proof that does not require a specific strategy for rule application.

References

- [1] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.
- [2] F. Baader, H.-J. Bürckert, B. Hollunder, W. Nutt, and J. H. Siekmann. Concept logics. In *Proceedings of the Symposium on Computational Logics*, Brüssel, Belgium, 1990.
- [3] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.
- [4] F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithms. In M. Richter and H. Boley, editors, *Inter-*

- national Workshop on Processing Declarative Knowledge*, volume 567. Springer, 1991.
- [5] C. Bagnasco, P. Petrin, and L. Spampinato. Taxonomic reasoning in configuration tasks. Technical Report QR-91-1, Quinary SpA, 1991.
 - [6] J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219, 1981.
 - [7] R. J. Brachman, R. J. Bobrow, P. R. Cohen, J. W. Klovstad, B. L. Webber, and W. A. Woods. *Research in natural language understanding, annual report*. Technical Report No. 4274. Bolt, Beranek and Newman, Cambridge, Mass., 1979.
 - [8] M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.
 - [9] M. Buchheit, R. Klein, and W. Nutt. Configuration as model construction: The constructive problem solving approach. In *Proceedings of the Third International Conference on Artificial Intelligence in Design, AID'94*, Lausanne, Switzerland, 1994.
 - [10] M. Buchheit, R. Klein, and W. Nutt. Constructive problem solving: A model construction approach towards configuration. Technical Memo TM-95-01, DFKI Saarbrücken, 1995.
 - [11] 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.
 - [12] R. Klein. Model representation and taxonomic reasoning in configuration problem solving. In *Proceedings of the German Workshop on Artificial Intelligence, GWAI-91*, 1991.
 - [13] D. L. McGuinness and L. A. Resnick. Description logic-based configuration for consumers. In *Proceedings of the International Workshop on Description Logic, DL'95, Internal Report RAP. 07.95*, Università degli Studi di Roma “La Sapienza”, Rome, Italy, 1995.
 - [14] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*. LNAI 422. Springer-Verlag, Berlin, Germany, 1990.
 - [15] B. Owsnicki-Klewe. Configuration as a consistency maintenance task. In *Proceedings of the German Workshop on Artificial Intelligence, GWAI-88*, 1988.

- [16] J. Quantz. How to fit generalized quantifiers into terminological logics. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 543–547, Vienna, Austria, 1992.
- [17] Klaus Schild. Terminological cycles and the propositional μ -calculus. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany, 1994.
- [18] W. van der Hoek and M. De Rijke. Generalized quantifiers and modal logic. *Journal of Logic, Language and Information*, 2:19–58, 1993.
- [19] R. Weida. Closing the terminology. In *Proceedings of the International Workshop on Description Logic, DL'95, Internal Report RAP. 07.95*, Universita degli Studi di Roma “La Sapienza”, Rome, Italy, 1995.
- [20] J. R. Wright, D. L. McGuinness, C. Foster, and G. T. Vesonder. Conceptual modeling using knowledge representation: Configurator applications. In *Proceedings of the Workshop on AI in Distributed Information Networks, IJCAI-95*, Montreal, Canada, 1995.
- [21] J. R. Wright, E. S. Weixelbaum, K. Brown, G. T. Vesonder, S. R. Palmer, J. I. Berman, and H. H. Moore. A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems. *AI Magazine*, 14(3):69–80, 1993.