Hybrid Reasoning for Description Logics with Nominals and Qualified Number Restrictions

Jocelyne Faddoul[1] , Volker Haarslev[1], Ralf Möller[2]
[1] Concordia University, Montreal, Canada
{j_faddou, haarslev}@encs.concordia.ca
[2] Hamburg University of Technology, Hamburg, Germany
r.f.moeller@tuhh.de

Contents

1 Introduction

Description Logics (DLs) are a family of knowledge representation formalisms used to represent and reason about an application's domain elements. They are interestingly applicable in the semantic web as they provide the basis for the Web Ontology Language (OWL) defined by the World Wide Web Consortium (W3C).

Practical DL reasoners such as FaCT++[1], RacerPro[2], and Pellet[3] implement tableau-based decision procedures. Such decision procedures typically decide a knowledge base consistency test by constructing a so-called pre-model for the knowledge base. Despite many optimization techniques [11] studied and implemented so far, they do not provide a generic practical DL reasoner. It is easy to find knowledge bases where one reasoner performs very well while the other is hopelessly inefficient. This is not only due to the high computational complexity of tableau calculi and inference services, but also to the fact that these algorithms create pre-models in an often blind way. Major inefficiency sources can be due to:

- (i) the high degree of non-determinism introduced by (a) the use of General Concept Inclusion axioms (GCIs) or (b) when merging domain elements is necessary,
- (ii) the construction of large models,
- (iii) the interaction between language constructors.

Recent work in [17, 18] shows how tableau algorithms can be combined with resolution algorithms to address the inefficiency of DL reasoning due to (i-a) and (ii).

In this report, we study an alternative reasoning algorithm which is by design more informed and can be used to address the inefficiency of DL reasoning due to (i-b), (ii) and (iii) where we address the interaction between nominals and qualified cardinality restrictions (QCRs). Our algorithm is hybrid because it combines tableau-based reasoning with arithmetic reasoning [21]. For ease of presentation we study this hybrid approach for the DL $\mathcal{ALCOQ}$ which is the basic description logic $\mathcal{ALC}$ extended with nominals and QCRs. The algorithm can be easily extended to handle role hierarchies, and transitive roles thus handling $\mathcal{SHOQ}$.

The algorithm is based on the assumption that domain elements consists of a set of individuals divided into subsets depending on their role filler membership and/or concept membership. Nominals can be seen as singleton sets and the at-least and at-most restrictions expressed in QCRs represent cardinality restrictions on the corresponding sets of role fillers. These restrictions on sets are encoded as linear inequations (as was first introducted in [19, 20]) solved by a linear programming (LP) algorithm (such as Simplex) with the objective of minimizing the sum of all cardinalities. If no solution for the inequations is possible, this means that the individuals cannot be distributed between sets without violating the cardinality restrictions. When a solution is returned, individuals are distributed among sets without violating any at-least or at-most restrictions and this number of individuals is minimal.

A preprocessing step which rewrites an $\mathcal{ALCOQ}$ knowledge base into an $\mathcal{ALCON}$ knowledge base with a weak form of role hierarchy allows the distinction between the numerical

[1]http://owl.man.ac.uk/factplusplus/

[2]http://www.racer-systems.com/

[3]http://pellet.owldl.org/

restrictions and the qualifications expressed by QCRs. The atomic decomposition technique allows the computation of all possible intersection between sets. It is applied on the sets of role fillers and nominals in contrast to the approaches presented in [20, 8, 6, 5] where the atomic decomposition is applied on sets of role fillers of a given individual. The decomposition of the sets of role fillers and nominals allows for the following.

- (a) The encoding of the nominals semantics into inequations.
- (b) The partitioning of the domain element into equivalence classes.
- (c) The handling of possible interactions between nominals and role fillers (based on QCRs, existential restrictions and number restrictions).

A standard tableau for $\mathcal{ALC}$ [3] is modified and extended to work with a linear inequation solver in such a way that it can:

- (1) Decide the satisfiability of concept descriptions that use propositional operators ($\sqcap, \sqcup, \neg$) and $\forall$, $\forall_\backslash$ operators. The $\forall_\backslash$ operator is a new operator introduced at pre-processing, see Algorithm 1.
- (2) Encode numerical restrictions on sets into a set of inequations processed by an inequation solver.
- (3) Make sure that a numerical solution satisfies logical restrictions by constructing a pre-model of the solution.

This hybrid algorithm comes with characteristics that render it not only novel but also well suited to optimize DL reasoning with nominals and QCRs due to the following.

1. Numerical restrictions are satisfied before creating domain elements, which means that domain elements are never merged and there is no need for a mechanism of merging or handling the so-called "yoyo" effect (a possibly infinite cycle of creating and merging domain elements which causes a termination problem). This also ensures a better handling of cycles while avoiding source (i-b) of inefficient reasoning.
2. Due to the partitioning of the domain element into equivalence classes, elements with the same restrictions fall into the same partition. We use one proxy element as a representative for each partition's elements. This means that there is no need to implement any blocking strategies to ensure termination since no two elements with the same restrictions will be created. The use of proxy elements by our tableau algorithm is inspired by [8] and is also used to address source (ii) of inefficient reasoning.
3. By setting the sum of all cardinalities as an objective function to be minimized by the linear inequation solver we can address source (ii) of inefficient reasoning by ensuring a minimum number of role fillers for each domain element.
4. The performance of an inequation solver is not affected by the values of the numbers used in the inequations. This means that relying on an inequation solver to reason about numerical restrictions, large values of numbers in QCRs are not problematic.

Finally, the hybrid calculus itself is not an optimization technique and a naïve implementation is by no means better then any other naïve implemenation of DL reasoning algorithms proposed so far. However, the calculus enjoys additional properties that help us set the ground for optimization techniques that address the sources of complexity of DL reasoning related to the interaction between nominals and number restrictions.[4] We therefore, limit the scope of this report to the illustration of the technical details and proofs for the correctness of the algorithm. Empirical evaluation and implementation optimizations will be reported in future work.

2 Preliminaries

In this section, we introduce the syntax, semantics and inference problems of $\mathcal{ALCOQ}$, i.e., the description logic $\mathcal{ALC}$ extended with nominals and qualified cardinality restrictions (QCRs).

2.1 Syntax and Semantics of $\mathcal{ALCOQ}$

Let N_{C}, N_{R} be non-empty and disjoint sets of concept names and role names respectively. Let $N_{\text{o}} \subseteq N_{\text{C}}$ be the set of nominals. The set of $\mathcal{ALCOQ}$ concepts is the smallest set such that: (i) every concept name $A \in N_{\text{C}}$ is a concept, and (ii) if C, D are concepts and R is a role name in N_{R} then $\neg C$, $(C \sqcup D)$, $(C \sqcap D)$, $(\exists R.C)$, $(\forall R.C)$, $(\geq nR.C)$, $(\leq nR.C)$ with $n \in \mathbb{N}$ are also concepts.

In the following we use $\top$ ($\bot$) as an abbreviation for $A \sqcup \neg A$ ($A \sqcap \neg A$) and $\geq nR$ ($\leq nR$) for $\geq nR.\top$ ($\leq nR.\top$).

Also since $\exists R.C$ can be converted to $\geq 1R.C$, we do not consider descriptions of the form $\exists R.C$.

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of $\Delta^{\mathcal{I}}$, a non-empty set of individuals, called the domain of the interpretation, and $\cdot^{\mathcal{I}}$, an interpretation function. The interpretation function $\cdot^{\mathcal{I}}$ maps each atomic concept $A \in N_{\text{C}}$ to a subset of $\Delta^{\mathcal{I}}$, and each atomic role $R \in N_{\text{R}}$ to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For all $\mathcal{ALCOQ}$ concepts, using $\#$ to denote the cardinality of a set, and given $\mathit{FIL}(R, s)$ defined as $\mathit{FIL}(R, s) = \{t \in \Delta^{\mathcal{I}} \mid \langle s, t\rangle \in R^{\mathcal{I}}\}$ such as the set of all *R-fillers* for a given role name R is defined as: $\mathit{FIL}(R) = \bigcup_{s \in \Delta^{\mathcal{I}}} \mathit{FIL}(R, s)$, the following must hold.

$$
\begin{array}{ll}
(C \sqcap D)^{\mathcal{I}} & = C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} & = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
\# o^{\mathcal{I}} & = 1 \text{ for all } o \in N_{\text{o}} \\
(\forall R.C)^{\mathcal{I}} & = \{s \in \Delta^{\mathcal{I}} \mid \langle s, t\rangle \in R^{\mathcal{I}} \Rightarrow t \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} & = \{s \in \Delta^{\mathcal{I}} \mid \exists t \colon \langle s, t\rangle \in R^{\mathcal{I}} \wedge t \in C^{\mathcal{I}}\} \\
(\geq nR.C)^{\mathcal{I}} & = \{s \in \Delta^{\mathcal{I}} \mid \#(\mathit{FIL}(R, s) \cap C^{\mathcal{I}}) \geq n\} \\
(\leq nR.C)^{\mathcal{I}} & = \{s \in \Delta^{\mathcal{I}} \mid \#(\mathit{FIL}(R, s) \cap C^{\mathcal{I}}) \leq n\}
\end{array}
$$

Definition 1 ($\mathcal{ALCOQ}$ TBoxes) An $\mathcal{ALCOQ}$ TBox $\mathcal{T}$ is a finite set of *general concept inclusion axioms* (GCIs) of the form $C \sqsubseteq D$, where C, D are concepts and $C \equiv D$ abbreviates

[4]Optimization technique for nominals were published only recently [22] with none addressing the interaction between nominals and QCRs.

$\{C \sqsubseteq D,\ D \sqsubseteq C\}$. A TBox $\mathcal{T}$ is said to be consistent if there exists an interpretation $\mathcal{I}$ satisfying $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each $C \sqsubseteq D \in \mathcal{T}$. $\mathcal{I}$ is called a model of $\mathcal{T}$. A concept C is said to be satisfiable w.r.t. a TBox $\mathcal{T}$ iff there exists a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$, i.e., there exists an individual $s \in C^{\mathcal{I}}$ as an instance of C. $\mathcal{I}$ is called a model of C w.r.t. $\mathcal{T}$.

Definition 2 ($\mathcal{ALCOQ}$ ABox) An $\mathcal{ALCOQ}$ ABox $\mathcal{A}$ is a finite set of concept membership assertions of the form $a{:}C$ or role membership assertions of the form $(a,b){:}R$ with a, b two individual names. An Abox $\mathcal{A}$ is said to be consistent w.r.t. $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ is satisfied for each $a{:}C$ in $\mathcal{A}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ is also satisfied for each $(a,b){:}R$ in $\mathcal{A}$.

In the following, we assume all $\mathcal{ALCOQ}$ concepts to be in their *negation normal form* (NNF). We use $\dot{\neg}C$ to denote the NNF of $\neg C$ and $\mathit{nnf}(C)$ to denote the NNF of C. Finally, we do not impose the *unique name assumption* (UNA) and therefore two nominals can refer to the same individual. However, the UNA can be easily applied by extending the initial TBox $\mathcal{T}$ with $\bigcup_{o_i, o_j \in N_o} \{o_i \sqsubseteq \neg o_j\}$ for $1 \leq i, j \leq \#N_o$ and $i \neq j$.

2.2 $\mathcal{ALCOQ}$ Inferences

Using nominals, concept satisfiability and ABox consistency can be reduced to TBox consistency; a concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ iff $(\mathcal{T} \cup \{o \sqsubseteq C\})$ is consistent and $o \in N_o$ new in $\mathcal{T}$, an ABox $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff $(\mathcal{T} \cup \bigcup_{(a:C)\in\mathcal{A}} \{a \sqsubseteq C\} \cup \bigcup_{((a,b):R)\in\mathcal{A}} \{a \sqsubseteq \exists R.b\})$ is consistent. Hence, in the following without loss of generality we restrict our attention to TBox consistency.

The concept axioms in $\mathcal{T}$ can be reduced to a single axiom $\top \sqsubseteq C_{\mathcal{T}}$ such that $C_{\mathcal{T}}$ abbreviates $\bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \mathit{nnf}(\neg C \sqcup D)$ [14]. TBox consistency can be checked by testing the consistency of $o \sqsubseteq C_{\mathcal{T}}$ with $o \in N_o$ new in $\mathcal{T}$, which means that at least $o^{\mathcal{I}} \in C_{\mathcal{T}}{}^{\mathcal{I}}$ and $C_{\mathcal{T}}{}^{\mathcal{I}} \neq \emptyset$. Moreover, since $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ then every domain element must also satisfy $C_{\mathcal{T}}$ ($C_{\mathcal{T}}$ holds for each domain element).

3 Motivation

In this section, we illustrate the challenges of DL reasoning with nominals and QCRs and we show how to address them using a hybrid reasoning approach which combines DL reasoning with arithmetic reasoning.

3.1 Nominals and QCRs

Nominals [23], known as named individuals, are studied in the areas of hybrid logic [4] as well as description logics. They play an important role in DL as they allow one to express the notion of uniqueness and identity; nominals must be interpreted as singleton sets. They are used to define concepts as an enumeration ($\mathit{Bool} \equiv (t \sqcup f)$ with t, f nominals), or as role fillers ($\mathit{Quebecor} \equiv \exists.\mathit{CitizenOf}.\mathit{quebec}$ with quebec a nominal). It is easy to find ontologies (for example, WINE[5] ontology) where nominals appear as names for specific individuals, countries, colors, etc...

[5]http://www.w3.org/TR/owl-guide/wine.rdf

QCRs are required in many applications [10, 1], they allow one to specify lower ($\geq nR.C$) and upper ($\leq nR.C$) bounds on the number of elements related via a certain role (R) with additionally specifying qualities on the related elements. They appear naturally in many domains. For example, in modeling the human skeletal system [16] one would need to say that a full grown adult has at least 14 bones in the face and 27 bones in the head ($\mathit{Adult} \sqsubseteq \geq 27 \mathit{hasBones}.\mathit{HeadBone} \sqcap \geq 14 \mathit{hasBones}.\mathit{FaceBone}$) and at most 306 bones ($\mathit{Adult} \sqsubseteq \leq 306 \mathit{hasBones}.\mathit{Bone}$).

3.2 Nominals and Numerical Restrictions

Nominals carry numerical restrictions; they not only name individuals but also allow us to count them. For example, defining the concept of an *EU* member state as *EU_MemberState* $\equiv$ *Austria* $\sqcup \ldots \sqcup$ *UK* where *Austria*,...,*UK* are all nominals means that instances of *EU_MemberState* can only be one of the 27 member states. This additional information carried with nominals interacts with concepts and roles in a way that can limit the number of instances of a certain concept or fillers for a certain role. Given an individual s, instance of a concept E ($s \in E^{\mathcal{I}}$), $C \in N_{\mathrm{C}}$, $R \in N_{\mathrm{R}}$, $o_1, \ldots, o_n \in N_{\mathrm{o}}$, and n, m non-negative integers, we distinguish between *global* and *local* numerical restrictions.

Local Restrictions

When E is of the form ($\geq nR$), or ($\leq mR$) it holds a numerical restriction on the cardinality of the set of *R-fillers* of s. For example, $s \in (\geq 2R)^{\mathcal{I}}$ imposes that at least 2 individuals s_1 and s_2 must be *R-filler*s of s, and therefore $\#\mathit{FIL}(R, s) \geq 2$. When E is of the form ($\forall R.C$) it also holds a numerical restriction due to $s \in (\forall R.C)^{\mathcal{I}} \Leftrightarrow s \in (\leq 0R.\neg C)^{\mathcal{I}}$, and $s \in (\forall R.(o_1 \sqcup \cdots \sqcup o_n))^{\mathcal{I}} \Rightarrow s \in (\leq nR)^{\mathcal{I}}$. These restrictions are local since they only affect the set of individuals that are *R-fillers* of s, $\mathit{FIL}(R, s)$.

Global Restrictions

Having $E \sqsubseteq o_1 \sqcup \cdots \sqcup o_n \in \mathcal{T}$ ($o_1 \sqcup \cdots \sqcup o_n \sqsubseteq E \in \mathcal{T}$) enforces a numerical restriction on the cardinality of the set of instances of E; there can be at-most (at-least) n instances of E corresponding to the interpretation of $o_1^{\mathcal{I}} \cup \ldots \cup o_n^{\mathcal{I}}$ assuming $o_1, \ldots, o_n$ are all disjoint. Such at-most (at-least) restrictions carried with nominals are global since they can affect the set of all individuals in the domain of interpretation ($\Delta^{\mathcal{I}}$). Nominals can specify concept cardinalities [2] as was shown in [24] and can interact with local restrictions. For example having $s \in (\forall R.E)^{\mathcal{I}}$ then the set of *R-fillers* of s is bounded by n (the size of E), $\mathit{FIL}(R, s) \leq n$ ($\mathit{FIL}(R, s) \geq n$).

3.3 Reasoning Challenges with Nominals

DL reasoning with nominals is challenging for existing DL reasoning algorithms due to the following:

1. The tree-model property is lost: The tree model property has been advantageous for DL tableau algorithms by allowing them to search for tree-like models.

2. Nominals must be treated as concepts: This is a challenging task for reasoning algorithms because in order to preserve the *nominal* semantics, each *nominal* must be interpreted as exactly one individual, whereas a concept is interpreted as a set of individuals.

3. Nominals interact with other constructors: The worst case is when *nominals* interact with *inverse roles* and *QCRs* [13]. Each of these constructs alone is challenging to reason with and needs special optimization techniques.

3.4 Reasoning Challenges with QCRs

DL reasoning with QCRs is challenging with existing DL reasoning algorithms due to the following:

1. Non-determinism introduced when choosing a distribution for each individual created to make sure that an at most restriction is not violated ($\leq nR.C$).

2. The use of large numbers with QCRs result in the creation of large models.

3. Non-determinism introduced when merging individuals to satisfy an at-most restriction

Clearly DLs that enable *nominals* and *QCRs* enjoy additional expressive power for instance, there is no other way to close a concept or domain with a finite number of elements using the DL $\mathcal{SHOIQ}$ except using *nominals.* However, the expressive power usually comes with challenging reasoning tasks.

Most semantic web reasoners supporting nominals and QCRs implement tableau-based decision procedures which usually need to be equipped with a set of optimization techniques [22] as their naïve implementations fail to be practical. Decision procedures were published in [14] with very weak implementations if any (no DL reasoner was able to classify the WINE[6] ontology until recent efforts [22]). In the presence of nominals, existing DL reasoning algorithms look for forest-like models which consist of trees rooted with arbitrarily interconnected nominals. They also need to make a clear distinction between a nominal and an individual. This distinction is crucial in maintaining the nominal semantics. In the case when two individuals need to be merged and one is a nominal; the nominal must survive. In the case when blocking is applicable; there cannot be a nominal node between an individual and a blocking individual otherwise by repeating the cycle, the nominal would also be repeated and the semantics is violated. The interaction between nominals, number restrictions and inverse roles is addressed using special tableau rules (*NN*-rule [13], *NI*-rule [17]).

Resolution-based reasoning procedures were proposed in [15] and were proven to be weak in dealing with large numbers in QCRs. Hypertableaux [18] which combines tableau and resolution-based reasoning were recently studied to minimize non-determinism in DL reasoning with no special treatment for QCRs. One might argue that there is no need to handle large numbers in ontologies. However, this seems to be a chicken and egg problem; we do not see a lot of ontologies using large numbers because no available reasoner handle large numbers yet. There exists a lot of cases where we need to use large numbers such as specifying that a person has 230 movable and semi movable joint ($Person \sqsubseteq \geq 230$

[6]http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine.rdf

$hasJoint.(MovableJoint \sqcup SemiMovableJoint))$ as part of the human skeletal system [16] representation.

4 Overview of the Hybrid Reasoning Algorithm for $\mathcal{ALCOQ}$

We present a novel reasoning algorithm that overcomes the challenge of reasoning with nominals by efficiently handling their interaction with QCRs. The calculus not only extends the one in [6, 5] to include nominals and GCIs but also relies on using proxy individuals and can be easily extended for the DL $\mathcal{SHOQ}$. It is based on three intuitions:

1. Nominals carry numerical restrictions.
2. One can think of a model as sets of individuals. Numerical restrictions on these sets can be handled using a linear inequation solver which, by considering a minimal solution ensures that a minimum number of nominals and role fillers satifies number restrictions.
3. To reduce the number of individuals in a pre-model one can use proxy individuals as representatives of individuals satisfying common restrictions.

Before describing our calculus we define some preprocessing of concept descriptions and we introduce the principles of the used non-tableau reasoning methods.

4.1 Preprocessing

We define a new concept operator $(\forall(R\backslash S).D)$ and a role implication operator $(R \sqsubseteq S)$ needed to rewrite $C_\mathcal{T}$ before applying the calculus. These operators are based on set semantics such that given an interpretation $\mathcal{I}$, then $(\forall(R\backslash S).D)^\mathcal{I} = \{s \in \Delta^\mathcal{I} \mid \langle s,t\rangle \in R^\mathcal{I} \wedge \langle s,t\rangle \notin S^\mathcal{I} \Rightarrow t \in D^\mathcal{I}\}$ is satisfied and $(R^\mathcal{I} \subseteq S^\mathcal{I})$ is satisfied for each role implication $R \sqsubseteq S \in \mathcal{R}$, with $\mathcal{R}$[7] a set of role implications.

Given $C_\mathcal{T}$, a set N_R of role names, and an empty set $\mathcal{R}$ of role implications, we re-write $C_\mathcal{T}$ using Algorithm 1. Please note that each rewriting step for a number restriction introduces a fresh role name R' new in $\mathcal{T}$ and $\mathcal{R}$ (See examples in Section 5.3).

Algorithm 1 *rw(E, N_R, $\mathcal{R}$)* for rewriting concept descriptions. Given $A \in N_\mathrm{C}$, C , D $\mathcal{ALCOQ}$ concepts, $R \in N_\mathrm{R}$ and $\mathcal{R}$ the set of role implications, the following rewriting holds:

```
$rw(A, N_\mathrm{R}, \mathcal{R}) \longrightarrow A, N_\mathrm{R}, \mathcal{R}$
$rw(\neg A, N_\mathrm{R}, \mathcal{R}) \longrightarrow \neg A, N_\mathrm{R}, \mathcal{R}$
$rw((C \sqcap D), N_\mathrm{R}, \mathcal{R}) \longrightarrow rw(C, N_\mathrm{R}, \mathcal{R}) \sqcap rw(D, N_\mathrm{R}, \mathcal{R}), N_\mathrm{R}, \mathcal{R}$
$rw((C \sqcup D), N_\mathrm{R}, \mathcal{R}) \longrightarrow rw(C, N_\mathrm{R}, \mathcal{R}) \sqcup rw(D, N_\mathrm{R}, \mathcal{R}), N_\mathrm{R}, \mathcal{R}$
$rw((\neg C), N_\mathrm{R}, \mathcal{R}) \longrightarrow rw(\dot{\neg} C, N_\mathrm{R}, \mathcal{R}), N_\mathrm{R}, \mathcal{R}$
$rw(\forall R.C, N_\mathrm{R}, \mathcal{R}) \longrightarrow \forall R.rw(C, N_\mathrm{R}, \mathcal{R}), N_\mathrm{R}, \mathcal{R}$
$rw((\geq nR.C), N_\mathrm{R}, \mathcal{R}) \longrightarrow (\geq nR' \sqcap \forall R'.rw(C, N_\mathrm{R}, \mathcal{R})), N_\mathrm{R} \cup \{R'\}, \mathcal{R} \cup \{R' \sqsubseteq R\}$
$rw((\leq nR.C), N_\mathrm{R}, \mathcal{R}) \longrightarrow (\leq nR' \sqcap \forall R'.rw(C, N_\mathrm{R}, \mathcal{R}) \sqcap \forall\ (R\backslash R').rw(\dot{\neg} C, N_\mathrm{R}, \mathcal{R})),$
    $N_\mathrm{R} \cup \{R'\}, \mathcal{R} \cup \{R' \sqsubseteq R\}$
```

[7] $\mathcal{R}$ is a weak form of role hierarchy $\mathcal{H}$ [14].

Lemma 1 (Preserving Satisfiability) *Rewriting concept expressions according to Algorithm 1 preserves satisfiability. Satisfying $C_\mathcal{T}$ consists of satisfying $rw(C_\mathcal{T}, N_\mathrm{R}, \mathcal{R})$ w.r.t. $\mathcal{R}$.*

Proof. It is easy to see that satisfiability is preserved for atomic concepts, negated concepts, conjunctions and disjunctions of concepts. Let C, D be $\mathcal{ALCOQ}$ concepts, n, m non-negative integer numbers and R a role name in N_R with $\mathcal{R}$ a set of role implications, we need to prove that $\geq nR.C$ is satisfiable iff $rw(\geq nR.C,\ N_\mathrm{R}, \mathcal{R})$, and $\leq mR.D$ is satisfiable iff $rw(\leq mR.D,\ N_\mathrm{R}, \mathcal{R})$. This means that we need to prove the following:

1. if $\geq nR.C$ is satisfiable then $\geq nR' \sqcap \forall R'.C$ is satisfiable w.r.t. $\mathcal{R}$.

   Proof. Let us assume that $\geq nR.C$ is satisfiable, this means that there exists a non-empty interpretation $\mathcal{I}$ with:

   (a) an individual $s \in \Delta^\mathcal{I}$ such that $s \in (\geq nR.C)^\mathcal{I}$.

   (b) n distinct individuals $t_1 \dots t_n \in \Delta^\mathcal{I}$ such that $t_i \in (FIL(R, s) \cap C^\mathcal{I})$ for $1 \leq i \leq n$.

   We show how we can construct the interpretation, $\mathcal{I}'$, of $\geq nR' \sqcap \forall R'.C$ from $\mathcal{I}$.

   We set $\mathcal{I}' = \mathcal{I}$ and we create a new role name R' in N_R such that $FIL(R',\mathrm{s}) = FIL(R,\mathrm{s}) \cap C^\mathcal{I}$. For $s \in \Delta^{\mathcal{I}'}$ the following holds:

   (a) $s \in (\geq nR')^{\mathcal{I}'}$ since $FIL(R',\mathrm{s}) \subseteq FIL(R,\mathrm{s})$ and there exists $t_1 \dots t_m \in FIL(R',\mathrm{s})$.

   (b) We can add $R' \sqsubseteq R \in \mathcal{R}$ and $\mathcal{I}'$ satisfies $\mathcal{R}$ because by definition of R' all the *R'-fillers* are also *R-fillers*.

   (c) $s \in (\forall R'.C)^{\mathcal{I}'}$ since $FIL(R',\mathrm{s}) \subseteq C^{\mathcal{I}'}$.

   (d) $s \in (\geq nR.C)^\mathcal{I}$ is not violated.

   Hence, if $\geq nR.C$ is satisfiable then $\geq nR' \sqcap \forall R'.C$ is also satisfiable w.r.t. $\mathcal{R}$.

2. if $\geq nR' \sqcap \forall R'.C$ is satisfiable w.r.t. $\mathcal{R}$ with $R' \sqsubseteq R \in \mathcal{R}$ then $\geq nR.C$ is satisfiable.

   Proof. Let us assume that $\geq nR' \sqcap \forall R'.C$ w.r.t. $\mathcal{R}$ is satisfiable, this means that there exists a non-empty interpretation $\mathcal{I}'$ with:

   (a) an individual $s \in \Delta^{\mathcal{I}'}$ such that $s \in (\geq nR' \sqcap \forall R'.C)^{\mathcal{I}'}$.

   (b) n distinct individuals $t_1 \dots t_n \in \Delta^{\mathcal{I}'}$ such that $t_i \in FIL(R', s)$ and $t_i \in C^{\mathcal{I}'}$ for $1 \leq i \leq n$.

   It is easy to construct the interpretation $\mathcal{I}$ of $\geq nR.C$ from $\mathcal{I}'$; if we set $\mathcal{I} = \mathcal{I}'$ then we have $s \in (\geq nR.C)^\mathcal{I}$ since there already exists n distinct individuals $t_1 \dots t_n \in \Delta^\mathcal{I}$ satisfying $t_i \in FIL(R, s) \cap C^\mathcal{I}$ for $1 \leq i \leq n$. Hence, if $\geq nR' \sqcap \forall R'.C$ is satisfiable w.r.t. $\mathcal{R}$ then $\geq nR.C$ is also satisfiable.

3. if $\leq mR.D$ is satisfiable then $\leq mR' \sqcap \forall R'.D \sqcap \forall(R \backslash R').\neg D$ is satisfiable w.r.t. $\mathcal{R}$.

Proof. Let us assume that $\leq mR.D$ is satisfiable, this means that there exists a non-empty interpretation $\mathcal{I}$ with:

(a) an individual $s \in \Delta^{\mathcal{I}}$ such that $s \in (\leq mR.D)^{\mathcal{I}}$.

(b) at most m individuals $t_1 \dots t_m \in \Delta^{\mathcal{I}}$ such that $t_i \in FIL(R, s)$ and $t_i \in D^{\mathcal{I}}$ for $1 \leq i \leq m$.

We show how we can construct the interpretation, $\mathcal{I}'$, of $\leq mR' \sqcap \forall R'.D \sqcap \forall(R\backslash R').\neg D$ from $\mathcal{I}$.

We set $\mathcal{I}' = \mathcal{I}$ and we create a new role name R' in N_{R} such that $FIL(R'$,s$) = FIL(R$,s$) \cap C^{\mathcal{I}}$. For $s \in \Delta^{\mathcal{I}'}$ the following holds:

(a) $s \in (\leq mR')^{\mathcal{I}'}$ since $FIL(R'$,s$) \subseteq FIL(R$,s$)$ and there exists $t_1 \dots t_m \in FIL(R'$,s$)$.

(b) We can add $R' \sqsubseteq R \in \mathcal{R}$ and $\mathcal{I}'$ satisfies $\mathcal{R}$ because by definition of R' all the *R'-fillers* are also *R-fillers*.

(c) $s \in (\forall R'.C)^{\mathcal{I}'}$ since $FIL(R'$,s$) \subseteq C^{\mathcal{I}'}$.

(d) $s \in (\forall R\backslash R'.\neg D)^{\mathcal{I}}$. Since we can have at most m individuals in $FIL(R$,s$) \cap D^{\mathcal{I}}$, this means that all intersections with $FIL(R$,s$)$ that do not also intersect with $FIL(R'$,s$)$ cannot intersect with $D^{\mathcal{I}}$; $FIL(R$,s$) \setminus FIL(R'$,s$) \subseteq \neg D^{\mathcal{I}}$.

(e) $s \in (\leq mR.D)^{\mathcal{I}}$ is not violated.

Hence if $\leq mR.D$ then $\leq mR' \sqcap \forall R'.D \sqcap \forall(R\backslash R').\neg D$ is satisfiable w.r.t. $\mathcal{R}$.

4. if $\leq mR' \sqcap \forall R'.A \sqcap \forall(R\backslash R').\neg A$ is satisfiable w.r.t. $\mathcal{R}$ then $\leq mR.A$ is satisfiable

Proof. Let us assume that $\leq mR' \sqcap \forall R'.A \sqcap \forall(R\backslash R').\neg A$ is satisfiable w.r.t. $\mathcal{R}$, this means that there exists a non-empty interpretation $\mathcal{I}'$ with:

(a) an individual $s \in \Delta^{\mathcal{I}'}$ such that $s \in (\leq mR' \sqcap \forall R'.D \sqcap \forall(R\backslash R').\neg D)^{\mathcal{I}'}$.

(b) at most m distinct individuals $t_1 \dots t_m \in \Delta^{\mathcal{I}'}$ such that $t_i \in FIL(R, s)$ and $t_i \in D^{\mathcal{I}'}$ for $1 \leq i \leq m$.

(c) $FIL(R$,s$) \setminus FIL(R'$,s$) \subseteq \neg D^{\mathcal{I}'}$.

It is easy to construct the interpretation $\mathcal{I}$ of $\leq mR.D$ from $\mathcal{I}'$; if we set $\mathcal{I} = \mathcal{I}'$ then we have $s \in (\leq mR.D)^{\mathcal{I}}$ since there already exists at most m distinct individuals $t_1 \dots t_m \in \Delta^{\mathcal{I}}$ satisfying $t_i \in FIL(R, s) \cap D^{\mathcal{I}}$ for $1 \leq i \leq m$. Hence, if $\leq mR' \sqcap \forall R'.A \sqcap \forall(R\backslash R').\neg A$ is satisfiable w.r.t. $\mathcal{R}$ then $\leq mR.D$ is also satisfiable.

We also define *clos(C)* to be the smallest set of concepts such that: (a) $C \in$ *clos(C)*, (b) if $D \in$ *clos(C)* then $\dot{\neg} D \in$ *clos(C)*, (c) if $(E \sqcap D)$ or $(E \sqcup D) \in$ *clos(C)* then $E, D \in$ *clos(C)*, (d) if $(\forall R.D)$ or $(\forall R\backslash S.D) \in$ *clos(C)* then $D \in$ *clos(C)* and (e) if $(\geq nR.E)$ or $(\leq mR.E)$ $\in$ *clos(C)* then $E \in$ *clos(C)*. The size of *clos(C)* is bounded by the size of C. The set of relevant sub-concepts of a TBox $\mathcal{T}$ is then defined as $clos(\mathcal{T}) = clos(rw(C_{\mathcal{T}}, N_{\mathrm{R}}, \mathcal{R}))$.

4.2 Using Inequation Solving to Reason with Nominals and QCRs

In [5, 6] the hybrid algorithm uses the atomic decomposition technique to reduce reasoning about QCRs to linear inequation solving. With nominals the technique needs to be extended so that additionally the interaction between local and global numerical restrictions is handled while still preserving the semantics of nominals.

In the next paragraph we give the intuition behind the atomic decomposition and illustrate how it can be extended in the presence of nominals to decide the satisfiability of the at-least and at-most restrictions while preserving the nominals semantics using an inequation solver.

4.2.1 The Atomic Decomposition

The atomic decomposition technique [21] has been used in [6, 5, 9] to divide the sets of role fillers into mutually disjoint atomic sets and these set cardinalities are used as a bridging function to encode local numerical restrictions as inequations. With nominals, the atomic decomposition technique must consider the decomposition of the whole domain into disjoint sets such that each domain element belongs to exactly one set. Thus, one has to consider all possible interactions between local and global restrictions which can also be encoded into inequations.

We illustrate how this works: for each role $R \in N_{\mathrm{R}}$ that is used in a number restriction or a QCR, a fresh sub-role R' introduced by the preprocessing enables a simple role hierarchy; R' can have only one super role R. Let $H(R)$ denote the set of role names for all sub-roles of R: $H(R) = \{R' \,|\, (R' \sqsubseteq R) \in \mathcal{R}\}$. We do not need to add R to $H(R)$ since it is always implied and does not occur in number restrictions anymore after preprocessing. For every role $R' \in H(R)$, the set of R'-*fillers* forms a subset of the set of R-*fillers* ($FIL(R') \subseteq FIL(R)$). We define $\overline{R'}$ to be the complement of R' w.r.t. $H(R)$, the set of $\overline{R'}$-*fillers* is then defined as $\overline{R'}$-*fillers* $=(FIL(R) \setminus FIL(R'))$.

Each subset P of $H(R)$ ($P \subseteq H(R)$) defines a unique set of role names that admits an interpretation $P^{\mathcal{I}}$ corresponding to the unique intersection of role fillers for the role names in P: $P^{\mathcal{I}} = \bigcap_{\mathrm{R}' \in P} FIL(R') \cap \bigcap_{\mathrm{R}'' \in (H(R) \setminus P)} FIL(\overline{R''})$. $P^{\mathcal{I}}$ cannot overlap with role fillers for role names that do not appear in P since it is assumed to overlap with their complement. This makes all $P^{\mathcal{I}}$ disjoint. For example, as shown in Fig. 1 if $P_1 = \{R_1, R_2\}$, $P_2 = \{R_2, R_3\}$ and $H(R) = \{R_1, R_2, R_3\}$ this means that P_1 is the partition name for $FIL(R_1) \cap FIL(R_2) \cap FIL(\overline{R_3})$ which is equal to $P_1^{\mathcal{I}}$ and P_2 is the partition name for $FIL(R_2) \cap FIL(R_3) \cap FIL(\overline{R_1})$, and therefore, although $P_1 \cap P_2 = \{R_2\}$ we have $P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}} = \emptyset$ (see [21]). Since $\mathcal{ALCOQ}$ does not allow concept expressions using role complements, no role complement will be explicity used. For ease of presentation, we do not list the role complements in a partition name.

For each nominal $o \in N_{\mathrm{o}}$, $o^{\mathcal{I}}$ can interact with R-*fillers* for some R in N_{R} and become a subset of the set of R-*fillers* ($o^{\mathcal{I}} \subseteq FIL(R)$ if having for example $\geq 1R.C \sqcap \forall R.o$) . In order to handle such interactions we define the set NR of all possible role names and nominals as: $NR = \bigcup_{R \in N_{\mathrm{R}}} H(R) \cup N_{\mathrm{o}}$. Let $\mathcal{P}$ be the set of partition names defined for the decomposition of NR: $\mathcal{P} = \{P \,|\, P \subseteq NR\}$. Then we have $\mathcal{P}^{\mathcal{I}} = \Delta^{\mathcal{I}}$ because it includes all possible domain elements which correspond to a nominal and/or a role filler ;$\mathcal{P}^{\mathcal{I}} = \bigcup_{P \subseteq NR} P^{\mathcal{I}}$.

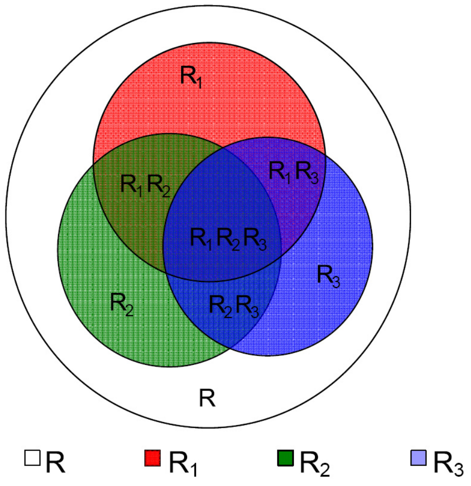

Figure 1: Atomic Decomposition of $H(R) = \{R_1, R_2, R_3\}$

4.2.2 Mapping Cardinalities to Variables

We assign a variable name v for each partition name P such that v can be mapped to a non-negative integer value n using $\sigma : \mathcal{V} \rightarrow \mathbb{N}$ such that $\sigma(v)$ denotes the cardinality of $P^{\mathcal{I}}$. Let $\mathcal{V}$ be the set of all variable names and α: $\mathcal{V} \rightarrow \mathcal{P}$ be a one-to-one mapping between each partition name $P \in \mathcal{P}$ and a variable $v \in \mathcal{V}$ such that $\alpha(v) = P$, and if a non-negative integer n is assigned to v using σ then $\sigma(v) = n = \#P^{\mathcal{I}}$. Let V_S denote the set of variable names mapped to partitions for a role ($S \in N_R$) or a nominal ($S \in N_o$) then V_S is defined as $V_S = \{v \in \mathcal{V} \,|\, S \in \alpha(v)\}$.

4.2.3 Reasoning with Nominals and QCRs

Given a partitioning $\mathcal{P}$ for all roles in N_{R} and nominals in N_{o} and a mapping α of variables, we can reduce a conjunction of $(\geq nR)$ and $(\leq mR)$ to a set of inequations and reason about the numerical restrictions using an inequation solver based on the following principles.

P1: Encoding Number Restrictions and Nominals Into Inequations. Since the partitions in $\mathcal{P}$ are mutually disjoint and the cardinality function is additive, a lower (upper) bound n (m) on the cardinality of the set of role fillers $FIL(S)$ for some role $S \in H(R)$ can be reduced to an inequation of the form $\sum_{v \in V_S} v \geq n$ ($\sum_{v \in V_S} v \leq m$). Thus, we can easily convert an expression of the form $(\geq nS)$ or $(\leq mS)$ into an inequation using ξ such that $\xi(S, \geq, n) = \sum_{v \in V_S} v \geq n$, and $\xi(S, \leq, m) = \sum_{v \in V_S} v \leq m$. The cardinality of a partition with a nominal can only be 1 based on the nominals semantics which is encoded into inequations using $\xi(o, \geq, 1)$ and $\xi(o, \leq, 1)$ for each nominal $o \in N_{\mathrm{o}}$. In this way we make sure that the nominal semantics is preserved; there is one and only one individual for each $o \in N_{\mathrm{o}}$: $\#o^{\mathcal{I}} = 1$.

P2: Getting a Solution. Given a set ξ_a of inequations, an integer solution defines the mapping σ for each variable v occurring in ξ_a to a non-negative integer n denoting the cardinality of the corresponding partition. For example, assuming $\sigma(v) = 4$ and $\alpha(v) = \{R', R''\}$, this means that the corresponding partition $(\alpha(v))^{\mathcal{I}}$ must have 4 fillers; $\#(FIL(R') \cap FIL(R'')) = 4$. Additionally, by setting the objective function to minimize the sum of all variables a minimum number of role fillers is ensured at each level. σ then defines a distribution of individuals that is consistent with the numerical restrictions encoded in ξ_a and the hierarchy expressed in $\mathcal{R}$.

4.3 Partitions are Signatures

A given model $\mathcal{I}$ of a TBox $\mathcal{T}$ consists of domain elements grouped into mutually disjoint partitions. Each partition represents a signature of concept descriptions that is common to all elements in the partition. A model $\mathcal{I}$ of $\mathcal{T}$ satisfies a signature $F \subseteq clos(\mathcal{T})$ iff $F^{\mathcal{I}} = \bigcap_{E \in F} E^{\mathcal{I}} \neq \emptyset$.

Lemma 2 *Let $\mathcal{I}$ be a model of $\mathcal{T}$, p a non-empty partition in P ($p^{\mathcal{I}} \subseteq \mathcal{P}^{\mathcal{I}}$), and i, j two domain elements such that $i, j \in p^{\mathcal{I}}$. If $i \in F^{\mathcal{I}}$ (F is the signature of $p^{\mathcal{I}}$) then:*

1. *$j \in F^{\mathcal{I}}$ and,*
2. *there exists no other domain element $i' \in \Delta^{\mathcal{I}}$ such that $i' \in p^{\mathcal{I}} \cap F^{\mathcal{I}} \cap p'^{\mathcal{I}}$ for some partition $p'^{\mathcal{I}} \subseteq \mathcal{P}^{\mathcal{I}}$ different from $p^{\mathcal{I}}$.*

Proof. It is easy to prove (2) since all partitions are disjoint by definition. For (1), given $R_1, \ldots R_n \in N_{\mathrm{R}}$, $o_1, \ldots, o_n \in N_{\mathrm{o}}$ and $i, j \in \Delta^{\mathcal{I}}$ consider the following cases.

- Case 1 - Nominals partition: $p^{\mathcal{I}}$ is a nominals partition, then it corresponds to some partition name $p \in \mathcal{P}$ of the form $p = \{o_1, \ldots, o_n\}$ and individuals in $p^{\mathcal{I}}$ satisfy the signature F such that $F^{\mathcal{I}} = p^{\mathcal{I}} = (o_1^{\mathcal{I}} \cap \ldots \cap o_n^{\mathcal{I}})$. Given the nominals semantics, $i \in F^{\mathcal{I}}$ and if there exists $j \in p^{\mathcal{I}}$ then $j \in F^{\mathcal{I}}$ since $i = j$; there is only one element in $p^{\mathcal{I}}$.

- Case 2 - Role filler partition: $p^{\mathcal{I}}$ is a role filler partition, then it corresponds to some partition name $p \in \mathcal{P}$ of the form $p = \{R_1, \ldots, R_n\}$ and individuals in $p^{\mathcal{I}}$ satisfy $p^{\mathcal{I}} = (FIL(R_1) \cap \ldots \cap FIL(R_n))$. If $i, j \in p^{\mathcal{I}}$ then $i, j \in (FIL(R_1) \cap \ldots \cap FIL(R_n))$; assume $i \notin (FIL(R_1) \cap \ldots \cap FIL(R_n))$ then i is a nominal or an R_x-filler for some $x > n$. However i cannot be a nominal since $p \cap N_{\mathrm{o}} = \emptyset$. Without loss of generality, assuming $i \in FIL(R_1)$ but $i \notin (FIL(R_2) \cap \ldots \cap FIL(R_n))$ this means that i belongs to a partition $p'^{\mathcal{I}}$ corresponding to some partition name $p' \in \mathcal{P}$ such that $R_1 \in p'$ and $\{R_2, \ldots, R_n\} \not\subseteq p'$. Now we have p' different from p with $i \in (p \cap p')$, this is a contradiction since partitions are disjoint. Therefore, $i \in (FIL(R_1) \cap \ldots \cap FIL(R_n))$, and by analogy we prove that $j \in (FIL(R_1) \cap \ldots \cap FIL(R_n))$. Therefore both i and j must satisfy the signature F such that $F^{\mathcal{I}} = \cap_{\forall R_1.C_1 \in \mathcal{T}} C_1^{\mathcal{I}} \ldots \cap_{\forall R_n.C_n \in \mathcal{T}} C_n^{\mathcal{I}}$.

- Case 3 - Nominals and role filler partition: $p^{\mathcal{I}}$ is a role filler partition of nominals, then it corresponds to some partition name $p \in \mathcal{P}$ of the form $p = \{o_1 \ldots o_k, R_1 \ldots R_l\}$ for some k, l, $1 \leq k, l \leq n$, and individuals in $p^{\mathcal{I}}$ satisfy $p^{\mathcal{I}} = (\bigcap_{1 \leq k \leq n} o_k^{\mathcal{I}} \cap \bigcap_{1 \leq l \leq n} FIL(R_l))$. Given the nominals semantics and similarly to case 1 if there exists $i, j \in p^{\mathcal{I}}$ then $i = j$. The signature F for $p^{\mathcal{I}}$ is such that it satisfies $F^{\mathcal{I}} = \bigcap_{1 \leq k \leq n} o_k^{\mathcal{I}} \cap \bigcap_{\forall R_1.C \in \mathcal{T}} C^{\mathcal{I}}$.

5 A Hybrid Tableau Algorithm for $\mathcal{ALCOQ}$

Before describing our hybrid algorithm, we define a tableau for $\mathcal{ALCOQ}$ TBox consistency.

5.1 The Tableau

Our tableau is different from other tableaux for $\mathcal{ALCOQ}$ by the way it ensures the semantics of the QCRs operator and the new operator ($\forall(R\backslash S)$) introduced after preprocessing an $\mathcal{ALCOQ}$ TBox.

Definition 3 (Tableau) Given an $\mathcal{ALCOQ}$ TBox $\mathcal{T}$ rewritten by applying Algorithm 1, we define a tableau $\mathrm{T} = (\mathbf{S}, \mathcal{L}, \mathcal{E})$ as an abstraction of a model for $\mathcal{T}$ with $\mathbf{S}$ a non-empty set of individuals, $\mathcal{L} : \mathbf{S} \rightarrow 2^{\mathrm{cl}(\mathcal{T})}$ a mapping between each individual and a set of concepts, and $\mathcal{E}$: $N_{\mathrm{R}} \rightarrow 2^{\mathrm{S}\times\mathrm{S}}$ a mapping between each role and a set of pairs of individuals in $\mathbf{S}$. For all $s,t \in \mathbf{S}$, $A \in N_{\mathrm{C}}$, C,$D \in$ *clos($\mathcal{T}$)*, $o \in N_{\mathrm{o}}$, R, $S \in N_{\mathrm{R}}$, and given the definition $R^{\mathcal{T}}(s) = \{t \in \mathbf{S} \mid \langle s,t\rangle \in \mathcal{E}(R)\}$, properties 1 - 10 must always hold:

1. $C_{\mathcal{T}} \in \mathcal{L}(s)$
2. If $A \in \mathcal{L}$(s) then $\neg A \notin \mathcal{L}$(s).
3. If $C \sqcap D \in \mathcal{L}$(s) then $C \in \mathcal{L}(s)$ and $D \in \mathcal{L}(s)$.
4. If $C \sqcup D \in \mathcal{L}$(s) then $C \in \mathcal{L}(s)$ or $D \in \mathcal{L}(s)$.
5. If $\forall S.C \in \mathcal{L}$(s) and $\langle s,t\rangle \in \mathcal{E}(S)$ then $C \in \mathcal{L}(t)$.
6. If $\forall(R\backslash S).C \in \mathcal{L}$(s) and $\langle s,t\rangle \in \mathcal{E}(R)$, and $\langle s,t\rangle \notin \mathcal{E}(S)$ then $C \in \mathcal{L}(t)$.
7. If $(\geq nR) \in \mathcal{L}$(s) then $\#R^{\mathcal{T}}(s) \geq n$.
8. If $(\leq mR) \in \mathcal{L}$(s) then $\#R^{\mathcal{T}}(s) \leq m$.
9. If $\langle s,t\rangle \in \mathcal{E}(R)$ and $R \sqsubseteq S \in \mathcal{R}$, then $\langle s,t\rangle \in \mathcal{E}(S)$.
10. For each $o \in N_{\mathrm{o}}$, $\#\{s \in \mathbf{S} \mid o \in \mathcal{L}(s)\} = 1$

Lemma 3 *An $\mathcal{ALCOQ}$ TBox $\mathcal{T}$ is consistent iff there exists a tableau for $\mathcal{T}$.*

Proof. The proof is similar to the one found in [14]. Property 6 of this tableau ensures that the semantics of the $\forall(R\backslash S).C$ operator is preserved. Property 9 ensures that the form of role hierarchy introduced at preprocessing is preserved, together with Properties 8 and 7 this property ensures that the semantics of the original (before rewriting) QCRs is preserved. Property 10 ensures that the semantics of nominals is preserved.

5.2 The Algorithm

In this section, we describe a hybrid algorithm which decides the existence of a tableau for an $\mathcal{ALCOQ}$ TBox $\mathcal{T}$. Our algorithm is hybrid because it relies on tableau expansion rules working together with an inequation solver, the corresponding model is represented using a *compressed completion graph.* The *compressed completion graph* is different from the "so-called" completion graphs used in standard tableau algorithms for $\mathcal{ALCOQ}$ [12] in four ways.

- First, it makes no distinction between a nominal node and a blockable node.

- Second, it allows the re-use of existing nodes instead of creating new similar ones.
- Third, it implements no blocking or merging of existing nodes.
- Finally, it uses a new labeling for nodes to collect and encode the number restrictions into inequations.

However, in the context of this report we will use "completion graph" to refer to a "compressed completion graph".

Definition 4 (Compressed Completion Graph) A *compressed completion graph* is a directed graph $G = (V, E, \mathcal{L}, \mathcal{L}_\mathrm{E})$. Each node $x \in V$ is labeled with two labels: $\mathcal{L}(x)$ and $\mathcal{L}_E(x)$, and each edge $\langle x, y\rangle \in E$ is labeled with a set, $\mathcal{L}(\langle x, y\rangle) \subseteq N_\mathrm{R}$, of role names.

$\mathcal{L}(x)$ denotes a set of concept expressions and role names such that $\mathcal{L}(x) \subseteq clos(\mathcal{T}) \cup \mathcal{P}$. By doing this, we not only label nodes based on the concept descriptions that they satisfy but also based on the partition they belong to. A partition name might include a nominal or a role name. We do not need to distinguish whether a nominal $o \in \mathcal{L}(x)$ is part of a partition name or a concept expression; in both cases x satisfies the nominal o. When a role name R appears in $\mathcal{L}(x)$ this means that x belongs to the partition for *R-fillers* and can therefore be used as an *R-filler.* This tagging is needed for the re-use of individual nodes.

$\mathcal{L}_\mathrm{E}(x)$ denotes a set ξ_x of inequations that must have a non-negative integer solution. The set ξ_x is the encoding of $(\geq nR)$ and $(\leq mR) \in \mathcal{L}(x)$. In order to make sure that numerical restrictions local for a node x are satisfied while the global restrictions carried with nominals are not violated, $\mathcal{L}_\mathrm{E}(x)$ is propagated from each node to all its successors. This makes sure that nominals are globally preserved while still satisfying the numerical restrictions at each level.

Let $\mathcal{T}$ be a preprocessed TBox rewritten into $\mathcal{T} = \{\top \sqsubseteq C_\mathcal{T}\}$ with $C_\mathcal{T} = rw(C_\mathcal{T}, N_\mathrm{R}, \mathcal{R})$ and $\mathcal{P}$ the corresponding atomic decomposition. To decide the consistency of $\mathcal{T}$ we need to test the consistency of $C_\mathcal{T}$ using $i \in N_\mathrm{o}$ new in $\mathcal{T}$ such that $i^\mathcal{I} \in {C_\mathcal{T}}^\mathcal{I}$ and every new individual satisfies $C_\mathcal{T}$.

The algorithm starts with the completion graph $G = (\{r_0\}, \emptyset\ , \mathcal{L}, \mathcal{L}_\mathrm{E})$. With $\mathcal{L}_\mathrm{E}(r_o) = \bigcup_{o \in N_\mathrm{o}}\{\xi(o, \leq, 1),\ \xi(o, \geq, 1)\}$ which is an encoding of the nominal semantics. The node r_0 is artificial and is not considered as part of the model, it is only used to process the numerical restrictions on nominals using the inequation solver which returns a distribution for them. The distribution of nominals (solution) is processed by the *fil*-Rule (see Fig. 2) which non-deterministically initializes the individual nodes for nominals. After at least one nominal is created, G is expanded by applying the expansion rules given in Fig. 2 until no more rules are applicable or when a clash occurs. No clash triggers or rules other then the *fil*-Rule apply to r_o.

Definition 5 (Clash) A node x in $(V \setminus \{r_0\})$ is said to contain a *clash* if:

- (i) $\{C, \neg C\} \subseteq \mathcal{L}(x)$, or
- (ii) a subset of inequations $\xi_x \subseteq \mathcal{L}_\mathrm{E}(x)$ does not admit a non-negative integer solution, this case is decided by the inequation solver, or
- (iii) for some $o \in N_\mathrm{o}$, $\#\{x \in (V \setminus \{r_0\}) \,|\, o \in \mathcal{L}(x)\} > 1$.

$\sqcap$-Rule	**If** $C \sqcap D \in \mathcal{L}(x)$, and $\{C, D\} \not\subseteq \mathcal{L}(x)$
	then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{C, D\}$.
$\sqcup$-Rule	**If** $C \sqcup D \in \mathcal{L}(x)$, and $\{C, D\} \cap \mathcal{L}(x) = \emptyset$
	then set $\mathcal{L}(x) = \mathcal{L}(x) \cup \{E\}$ with $E \in \{C, D\}$.
$\forall$-Rule	**If** $\forall R.C \in \mathcal{L}(x)$ and there exists y and $R' \in (\mathcal{L}(\langle x, y\rangle) \cap (H(R) \cup \{R\}))$,
	with $C \notin \mathcal{L}(y)$
	then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.
$\forall_{\backslash}$-Rule	**If** $\forall(R\backslash S).C \in \mathcal{L}(x)$, and there exists y and
	$R' \in (\mathcal{L}(\langle x, y\rangle) \cap ((H(R) \cup \{R\})\backslash((H(S) \cup \{S\})))$ with $C \notin \mathcal{L}(y)$
	then set $\mathcal{L}(y) = \mathcal{L}(y) \cup \{C\}$.
$\leq$-Rule	**If** $(\leq nR) \in \mathcal{L}(x)$ and $\xi(R, \leq, n) \notin \mathcal{L}_{\mathrm{E}}(x)$
	then set $\mathcal{L}_{\mathrm{E}}(x) = \mathcal{L}_{\mathrm{E}}(x) \cup \{\xi(R, \leq, n)\}$.
$\geq$-Rule	**If** $(\geq nR) \in \mathcal{L}(x)$ and $\xi(R, \geq, n) \notin \mathcal{L}_{\mathrm{E}}(x)$
	then set $\mathcal{L}_{\mathrm{E}}(x) = \mathcal{L}_{\mathrm{E}}(x) \cup \{\xi(R, \geq, n)\}$.
ch-Rule	**If** there exists v occurring in $\mathcal{L}_{\mathrm{E}}(x)$ with $\{v \geq 1,\ v \leq 0\} \cap \mathcal{L}_{\mathrm{E}}(x) = \emptyset$
	then set $\mathcal{L}_{\mathrm{E}}(x) = \mathcal{L}_{\mathrm{E}}(x) \cup \{V\}$, $V \in \{v \geq 1, v \leq 0\}$.
fil-Rule	**If** there exists v occurring in $\mathcal{L}_{\mathrm{E}}(x)$ with $\sigma(v) = m$ and $m > 0$, and
	there exists no y with $\alpha(v) \subseteq \mathcal{L}(y)$
	then 1. create a new node y,
	2. set $\mathcal{L}(y) = \alpha(v) \cup \{C_{\mathcal{T}}\}$,
	3. set $\mathcal{L}_{\mathrm{E}}(y) = \mathcal{L}_{\mathrm{E}}(x)$.
e-Rule	**If** $(\geq nR) \in \mathcal{L}(x)$ and there exists y with $R \in \mathcal{L}(y)$ and $R \notin \mathcal{L}(\langle x, y\rangle)$
	then set $\mathcal{L}(\langle x, y\rangle) = \mathcal{L}(\langle x, y\rangle) \cup \{R\}$, and
	if $\mathcal{L}_{\mathrm{E}}(x) \not\subseteq \mathcal{L}_{\mathrm{E}}(y)$ then set $\mathcal{L}_{\mathrm{E}}(y) = \mathcal{L}_{\mathrm{E}}(y) \cup \mathcal{L}_{\mathrm{E}}(x)$.

Figure 2: Expansion rules for $\mathcal{ALCOQ}$

When no rules are applicable or there is a clash, a *completion graph* is said to be *complete.*

Definition 6 (Proxy node) We use a proxy individual node as a representative for the elements of each partition to test the satisfiability of the given signature. The partition must be empty if the signature is not satisfiable. This can be done due to the following lemma.

Lemma 4 (Using a Proxy Individual) *Given a graph G as a representation of a model $\mathcal{I}$ for a TBox $\mathcal{T}$. Let P be a non-empty partition in $\mathcal{P}^{\mathcal{I}}$ and n a non-negative integer assigned by the inequation solver such that $n = \#P$. It is sufficient to create one proxy node in G as a representative of the n individuals in P.*

Proof. Lemma 4 is an easy consequence of Lemma 2. Creating one node in G for P does not violate any at-least or at-most restriction since these restrictions are numerically satisfied by the inequation solver. Additionally, if one node causes a clash because the signature of P cannot be satisfied, then P must be empty and we do not represent it in G.

When G is complete and there is no clash, this means that the numerical as well as the logical restrictions are satisfied ($C_{\mathcal{T}}{}^{\mathcal{I}} \neq \emptyset$) and there exists a model for $\mathcal{T}$: the algorithm returns that $\mathcal{T}$ is consistent. Otherwise the algorithm returns that $\mathcal{T}$ is inconsistent.

5.2.1 Strategy of Rule Application

Given a node x in the completion graph, the expansion rules in Figure 2 are triggered when applicable based on the following priorities:

- **Priority 1:** $\sqcap$-*Rule*, $\sqcup$-*Rule*, $\forall$-*Rule*, *ch-Rule*, $\leq$-*Rule*, $\geq$-*Rule*,*e-Rule*.
- **Priority 2:** *fil-Rule*.
- **Priority 3:** $\forall_{\backslash}$-*Rule*.

The rules with Priority 1 can be fired in arbitrary order.

The *fil-Rule* has Priority 2 to ensure that all at-least and at-most restrictions for a node x are encoded and satisfied by the inequation solver before creating any new nodes. This justifies why role fillers or nominals are never merged nor removed from the graph; a distribution of role fillers and nominals either survives into a complete model or fails due to a clash. Also, assigning the *fil-Rule* Priority 2 helps in early clash detection in the case when a numerical clash is detected by the inequation solver even before new nodes are created.

The $\forall_{\backslash}$-*Rule* has Priority 3. By giving this priority we ensure that the semantics of the $\forall(R\backslash S)$ operator are not violated. We allow the creation of all possible edges between a node and its successors before applying the $\forall(R\backslash S)$ operator semantics. This rule priority is needed to ensure the completeness of the algorithm.

5.2.2 Explaining the Rules

The $\sqcap$-Rule, $\sqcup$-Rule and the $\forall$-Rule rules are similar to the ones in the standard tableau rules for $\mathcal{ALC}$ [3].

$\forall_{\backslash}$**-Rule**. This rule is used to ensure the semantics of the new operator $\forall(R\backslash S).D$ (defined in preprocessing) by making sure that all *R-fillers* are labelled, and together with the *ch*-Rule (see explanation below) it has the same effect as the *choose*-rule in [3] needed to detect the unsatisfiability of concepts like $((\geq 3R.C) \sqcap (\leq 1R.D) \sqcap (\leq 1R.\neg D))$ (See Example 2 for details).

$\leq$**-Rule** and $\geq$**-Rule.** These rules encode the numerical restrictions in the label $\mathcal{L}$ of a node x into a set (ξ_x) of inequations maintained in $\mathcal{L}_{\mathrm{E}}(x)$ (P1). An inequation solver is always active and is responsible for finding a non-negative integer solution σ for ξ_x (P2) or triggering a clash if no solution is possible (see Def. 5). If the inequations added by these rules do not trigger a clash, encoded number restrictions can be satisfied by a possible distribution of role fillers.

*ch***-Rule.** This rule is used to check for empty partitions. Given a set of inequations in the label $(\mathcal{L}_{\mathrm{E}})$ of a node x and a variable v such that $\alpha(v) = P$ and $P \in \mathcal{P}$ we distinguish between two cases:

- (i) The case when $P^{\mathcal{I}}$ must be empty $(v \leq 0)$; this can happen when restrictions of elements assigned to this partition trigger a clash. For instance, if $\{\forall R_1.A,\ \forall R_2.\neg A\} \subseteq \mathcal{L}(x)$, $v \geq 1 \in \mathcal{L}_{\mathrm{E}}(x)$ and $P = \{R_1, R_2\}$ then a node y assigned to P with $\{R_1, R_2\} \subseteq \mathcal{L}(\langle x, y\rangle)$ triggers a clash $\{A, \neg A\} \subseteq \mathcal{L}(y)$ and $v \leq 0$ is enforced.

- (ii) The case when $P^{\mathcal{I}}$ must have at least one element ($1 \leq m \leq \sigma(v)$); if $P^{\mathcal{I}}$ can have at least one element without causing any logical clash, this means that we can have m elements also in $P^{\mathcal{I}}$ without a clash. Therefore, when creating nodes (using the *fil*-Rule) of corresponding partitions, it is enough to have one representative (proxy) node for each partition (see Lemma 4 for a proof).

Since the inequation solver is unaware of logical restrictions of filler domains we allow an explicit distinction between cases (i) and (ii). We do this by non-deterministically assigning ≤ 0 or ≥ 1 for each variable v occurring in $\mathcal{L}_{\mathrm{E}}(x)$.

***fil*-Rule**. This rule is used to generate individual nodes depending on the distribution (σ) returned by the inequation solver. The rule is fired for every non-empty partition P using $\sigma(v)$. It generates one proxy node as the representative for the m elements assigned to $P^{\mathcal{I}}$ by the inequation solver. The proxy individual is tagged with its partition name using $\alpha(v)$ in its label, $C_{\mathcal{T}}$ is also added to its label to make sure that every node created by the *fil*-Rule also satisfies $C_{\mathcal{T}}$.

***e*-Rule.** This rule connects a node x to a proxy individual y representing *R-fillers* of x by adding the edge for R between x and y. For instance, if $(\geq 2R) \in \mathcal{L}(x)$ and there exists a node y assigned to $P^{\mathcal{I}}$ such that $R \in P$ and $P \subseteq \mathcal{L}(y)$, this means that y can be used as an *R-filler* of x. Therefore, the e-Rule creates/updates the edge $\langle x, y\rangle$ with $R \in \mathcal{L}(\langle x, y\rangle)$. The node y can also be re-used to satisfy another $(\geq nR)$ restriction which gives this rule the ability to handle cycles without the need for blocking. For instance, if we have another node x_1 with $(\geq 2R) \in \mathcal{L}(x_1)$, y is re-used and R is added to the edge $\langle x_1, y\rangle$.

5.3 Examples

Example 1 To better illustrate the calculus, we demonstrate it by checking consistency of the following example TBox $\mathcal{T}$[8]

$$\mathcal{T} = \{ \begin{array}{lcl} A & \sqsubseteq & \geq 1R.(A \sqcap \geq 1R.A) \\ A & \sqsubseteq & o \\ o & \sqsubseteq & A \\ \} & & \end{array}$$

In this example, $N_{\mathrm{R}} = \{R\}$, $N_{\mathrm{o}} = \{o\}$ and $\mathcal{T}$ can be reduced to $\mathcal{T} = \{\top \sqsubseteq C_{\mathcal{T}}\}$ with $C_{\mathcal{T}} = (\neg A \sqcup \geq 1R.(A \sqcap \geq 1R.A)) \sqcap (\neg A \sqcup o) \sqcap (\neg o \sqcup A)$.

To test the consistency of $\mathcal{T}$, we need to check that at least one individual i is a member of $C_{\mathcal{T}}$ ($i \sqsubseteq C_{\mathcal{T}}$ with $i \in N_{\mathrm{o}}$ new in $\mathcal{T}$).

After rewriting $C_{\mathcal{T}}$ by applying Algorithm 1 we have:

$$\begin{array}{ll} C_{\mathcal{T}} & = (\neg A \sqcup ((\geq 1R_1) \sqcap \forall R_1.(A \sqcap \geq 1R_2 \sqcap \forall R_2.A))) \sqcap (\neg A \sqcup o) \sqcap (\neg o \sqcup A) \\ N_{\mathrm{R}} & = \{R, R_1, R_2\} \\ N_{\mathrm{o}} & = \{o, i\} \\ \mathcal{R} & = \{R_1 \sqsubseteq R,\ R_2 \sqsubseteq R\} \\ H(R) & = \{R_1, R_2\} \\ NR & = \{R_1, R_2, o, i\} \end{array}$$

The atomic decomposition of NR defines the set of disjoint partitions $\mathcal{P} = \{\{R_1\}, \{R_2\}, \{o\}, \{i\}, \{R_1, R_2\}, \{R_1, o\}, \{R_2, o\}, \{R_1, R_2, o\}, \{R_1, i\}, \{R_2, i\}, \{o, i\}, \{R_1, R_2, i\}, \{R_1, o, i\},$

[8]Adapted from [14].

$\{R_2, o, i\}$, $\{R_1, R_2, o, i\}\}$ and the set $\mathcal{V}$ of variables associated with each partition in $\mathcal{P}$: $\mathcal{V} = \{v_{\mathrm{R_1}}, v_{\mathrm{R_2}}, v_{\mathrm{o}}, v_{\mathrm{i}}, v_{\mathrm{R_1R_2}}, v_{\mathrm{R_1o}}, v_{\mathrm{R_2o}}, v_{\mathrm{R_1R_2o}}, v_{\mathrm{R_1i}}, v_{\mathrm{R_2i}}, v_{\mathrm{oi}}, v_{\mathrm{R_1R_2i}}, v_{\mathrm{R_1oi}}, v_{\mathrm{R_2oi}}, v_{\mathrm{R_1R_2oi}}\}$.

The calculus starts with the completion graph $G = (\{r_0\}, \emptyset, \mathcal{L}, \mathcal{L}_E)$ with

$$\mathcal{L}_{\mathrm{E}}(r_0) = \left\{ \begin{array}{l} v_{\mathrm{o}} + v_{\mathrm{R_1o}} + v_{\mathrm{R_2o}} + v_{\mathrm{R_1R_2o}} + v_{\mathrm{oi}} + v_{\mathrm{R_1oi}} + v_{\mathrm{R_2oi}} + v_{\mathrm{R_1R_2oi}} \geq 1 \\ v_{\mathrm{o}} + v_{\mathrm{R_1o}} + v_{\mathrm{R_2o}} + v_{\mathrm{R_1R_2o}} + v_{\mathrm{oi}} + v_{\mathrm{R_1oi}} + v_{\mathrm{R_2oi}} + v_{\mathrm{R_1R_2oi}} \leq 1 \\ v_{\mathrm{i}} + v_{\mathrm{R_1i}} + v_{\mathrm{R_2i}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{oi}} + v_{\mathrm{R_1oi}} + v_{\mathrm{R_2oi}} + v_{\mathrm{R_1R_2oi}} \geq 1 \\ v_{\mathrm{i}} + v_{\mathrm{R_1i}} + v_{\mathrm{R_2i}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{oi}} + v_{\mathrm{R_1oi}} + v_{\mathrm{R_2oi}} + v_{\mathrm{R_1R_2oi}} \leq 1 \end{array} \right\}$$

After applying the *ch*-Rule until it is not applicable anymore we might come up with the case where $v_{\mathrm{R_1R_2i}} \geq 1$ and all other variables are ≤ 0. A clash is detected since no arithmetic solution is possible because one variable indexed with o must be ≥ 1 to satisfy the inequations in $\mathcal{L}_{\mathrm{E}}(r_0)$.

Considering a completion graph with choices for the *ch*-Rule rule such as $v_{\mathrm{R_1R_2oi}} \geq 1$ and all other variables are $\leq$0, the inequation solver returns a solution σ with $\sigma(v_{\mathrm{R_1R_2oi}}) = 1$ and all other variables are zero.

The *fil*-Rule is applicable to r_0 and one new node y_1 is created such that $\mathcal{L}(y_1) = \alpha(v_{\mathrm{R_1R_2oi}}) \cup \{C_{\mathcal{T}}\} = \{R_1, R_2, o, i\} \cup \{(\neg A \sqcup ((\geq 1R_1) \sqcap \forall R_1.(A \sqcap \geq 1R_2 \sqcap \forall R_2.A))) \sqcap (\neg A \sqcup o) \sqcap (\neg o \sqcup A)\}$, and $\mathcal{L}_{\mathrm{E}}(y_1) = \mathcal{L}_{\mathrm{E}}(r_0)$.

After applying the $\sqcap$-Rule, and $\sqcup$-Rule to y_1 without having a clash, we get the label set to $\mathcal{L}(y_1) = \{R_1, R_2, o, i, A, (\geq 1R_1), \forall R_1.(A \sqcap (\geq 1R_2) \sqcap \forall R_2.A)\}$.

The $\geq$-Rule is applicable to y_1 and $(\geq 1R_1)$ is encoded into an inequation added to $\mathcal{L}_{\mathrm{E}}(y_1)$ such that

$$\mathcal{L}_{\mathrm{E}}(y_1)^{9} = \left\{ \begin{array}{l} v_{\mathrm{R_1R_2oi}} \geq 1 \\ v_{\mathrm{R_1R_2oi}} \leq 1 \\ v_{\mathrm{R_1}} + v_{\mathrm{R_1o}} + v_{\mathrm{R_1R_2}} + v_{\mathrm{R_1R_2o}} + v_{\mathrm{R_1i}} + v_{\mathrm{R_1oi}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{R_1R_2oi}} \geq 1 \end{array} \right\}$$

The *ch*-Rule can be applied several times (selecting the case ≤ 0) without having a clash and σ is still valid ($\sigma(v_{\mathrm{R_1R_2oi}}) = 1$).

The *e*-Rule is applicable to y_1 for $(\geq 1R_1)$ and $\mathcal{L}(\langle y_1, y_1\rangle)$ is set to $\{R_1\}$.

After applying the $\forall$-Rule and $\sqcap$-Rule once to y_1, we have $\mathcal{L}(y_1) = \{R_1, R_2, o, i, A, (\geq 1R_1), (\geq 1R_2), \forall R_2.A, \forall R_1.(A \sqcap (\geq 1R_2) \sqcap \forall R_2.A)\}$.

The $\geq$-Rule is applicable to y_1, it encodes $(\geq 1R_2)$ into the inequation $v_{\mathrm{R_2}} + v_{\mathrm{R_2o}} + v_{\mathrm{R_1R_2}} + v_{\mathrm{R_1R_2o}} + v_{\mathrm{R_2i}} + v_{\mathrm{R_2oi}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{R_1R_2oi}} \geq 1$ added to $\mathcal{L}_{\mathrm{E}}(y_1)$.

The *ch*-Rule can be applied several times (selecting the case $\leq$0) without having a clash and σ is still valid ($\sigma(v_{\mathrm{R_1R_2oi}}) = 1$).

The *e*-Rule is applicable for $(\geq 1R_2)$ and now $\mathcal{L}(\langle y_1, y_1\rangle) = \{R_1, R_2\}$.

No rules are applicable anymore and no clash has been detected: we have a completion graph consisting of a single node y_1[10] such that $\mathcal{L}(y_1) = \{R_1, R_2, o, i, A, (\geq 1R_1), (\geq 1R_2), \forall R_2.A, \forall R_1.(A \sqcap (\geq 1R_2) \sqcap \forall R_2.A)\}$, and $\mathcal{L}(\langle y_1, y_1\rangle) = \{R_1, R_2\}$.

Example 2 We illustrate how the calculus works when checking the consistency of the TBox

$$\mathcal{T} = \{\top \sqsubseteq \geq 3R.C \sqcap \leq 1R.D \sqcap \leq 1R.\neg D\}$$

[9]For sake of brevity, we do not list the variables that were mapped to zero by σ.

[10]The node r_0 is and will be ignored since it is not part of the model.

which is reduced to $\mathcal{T} = \{\top \sqsubseteq C_{\mathcal{T}}\}$ with $C_{\mathcal{T}} = ((\geq 3R_1 \sqcap \forall R_1.C) \sqcap (\leq 1R_2 \sqcap \forall R_2.D \sqcap \forall(R\backslash R_2).\neg D) \sqcap (\leq 1R_3 \sqcap \forall R_3.\neg D \sqcap \forall(R\backslash R_3).D))$. We need to check that at least one individual i is a member of $C_{\mathcal{T}} = rw(C_{\mathcal{T}}, N_{\mathrm{R}}, \mathcal{R})$ ($i \sqsubseteq C_{\mathcal{T}}$ with $i \in N_{\mathrm{o}}$ new in $\mathcal{T}$).

After rewriting $C_{\mathcal{T}}$ by applying Algorithm 1 we have:

$N_{\mathrm{R}} = \{R, R_1, R_2, R_3\}$
$N_{\mathrm{o}} = \{i\}$
$\mathcal{R} = \{R_1 \sqsubseteq R,\ R_2 \sqsubseteq R,\ R_3 \sqsubseteq R\}$
$H(R) = \{R_1, R_2, R_3\}$
$NR = \{R_1, R_2, R_3, i\}$

The atomic decomposition of NR defines the set $\mathcal{P}$ of disjoint partitions; $\mathcal{P} = \{\{R_1\}, \{R_2\}, \{R_3\}, \{i\}, \{R_1, R_2\}, \{R_1, R_3\}, \{R_2, R_3\}, \{R_1, R_2, R_3\}, \{R_1, i\}, \{R_2, i\}, \{R_3, i\}, \{R_1, R_2, i\}, \{R_1, R_3, i\}, \{R_2, R_3, i\}, \{R_1, R_2, R_3, i\}\}$ and the set $\mathcal{V}$ of variables associated with each partition in $\mathcal{P}$: $\mathcal{V} = \{v_{\mathrm{R_1}}, v_{\mathrm{R_2}}, v_{\mathrm{R_3}}, v_{\mathrm{i}}, v_{\mathrm{R_1R_2}}, v_{\mathrm{R_1R_3}}, v_{\mathrm{R_2R_3}}, v_{\mathrm{R_1R_2R_3}}, v_{\mathrm{R_1i}}, v_{\mathrm{R_2i}}, v_{\mathrm{R_3i}}, v_{\mathrm{R_1R_2i}}, v_{\mathrm{R_1R_3i}}, v_{\mathrm{R_2R_3i}}, v_{\mathrm{R_1R_2R_3i}}\}$.

The calculus starts with the completion graph $G = (\{r_0\}, \emptyset, \mathcal{L}, \mathcal{L}_E)$ with

$$\mathcal{L}_{\mathrm{E}}(r_0) = \left\{ \begin{array}{l} v_{\mathrm{i}} + v_{\mathrm{R_1i}} + v_{\mathrm{R_2i}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{R_3i}} + v_{\mathrm{R_1R_3i}} + v_{\mathrm{R_2R_3i}} + v_{\mathrm{R_1R_2R_3i}} \geq 1 \\ v_{\mathrm{i}} + v_{\mathrm{R_1i}} + v_{\mathrm{R_2i}} + v_{\mathrm{R_1R_2i}} + v_{\mathrm{R_3i}} + v_{\mathrm{R_1R_3i}} + v_{\mathrm{R_2R_3i}} + v_{\mathrm{R_1R_2R_3i}} \leq 1 \end{array} \right\}$$

Considering a completion graph with choices for the *ch*-Rule rule such as $v_{\mathrm{i}} \geq 1$ and all other variables are ≤ 0, the inequation solver returns a solution σ with $\sigma(v_i) = 1$ and all other variables are zero. The *fil*-Rule is applicable to r_0 and one new node y_1 is created such that $\mathcal{L}(y_1) = \alpha(v) \cup C_{\mathcal{T}} = \{i\} \cup \{(\geq 3R_1 \sqcap \forall R_1.C) \sqcap (\leq 1R_2 \sqcap \forall R_2.D \sqcap \forall(R\backslash R_2).\neg D) \sqcap (\leq 1R_3 \sqcap \forall R_3.\neg D \sqcap \forall(R\backslash R_3).D)\}$, and $\mathcal{L}_E(y_1) = \mathcal{L}_E(r_0)$. After several applications of the $\sqcap$-Rule, $\mathcal{L}(y_1) = \{i, (\geq 3R_1), \forall R_1.C, (\leq 1R_2), \forall R_2.D, \forall(R\backslash R_2).\neg D, (\leq 1R_3), \forall R_3.\neg D, \forall(R\backslash R_3).D)\}$. The $\leq$-Rule and $\geq$-Rule encode $\geq 3R_1$ and $\leq 1R_2$ and $\leq 1R_3$ into inequations added to $\mathcal{L}_E(y_1)$ such that if we do not consider the variables that are mapped to zero we have:

$$\mathcal{L}_{\mathrm{E}}(r_0) = \left\{ \begin{array}{l} v_{\mathrm{i}} \geq 1 \\ v_{\mathrm{i}} \leq 1 \\ v_{\mathrm{R_1}} + v_{\mathrm{R_1R_2}} + v_{\mathrm{R_1R_3}} + v_{\mathrm{R_1R_2R_3}} \geq 3 \\ v_{\mathrm{R_2}} + v_{\mathrm{R_1R_2}} + v_{\mathrm{R_2R_3}} + v_{\mathrm{R_1R_2R_3}} \leq 1 \\ v_{\mathrm{R_3}} + v_{\mathrm{R_1R_3}} + v_{\mathrm{R_2R_3}} + v_{\mathrm{R_1R_2R_3}} \leq 1 \end{array} \right\}$$

After applying the *ch*-Rule until it is not applicable anymore we might come up with the case where $v_{\mathrm{R_1}} \geq 1$ and all other variables are ≤ 0. A solution (σ) returned by the inequation solver maps $v_{\mathrm{R_1}}$ to 3 ($\sigma(v_{\mathrm{R_1}}) = 3$) and all other unassigned variables are zero. The *fil*-Rule is applicable to y_1 and one new node y_2 is created such that $\mathcal{L}(y_2) = \{R_1\} \cup \{C_{\mathcal{T}}\}$. Once the *e*-Rule is applied to y_0, $\mathcal{L}(\langle y_0, y_1\rangle)$ is set to $\{R_1\}$. The $\forall$-Rule becomes applicable to y_0 and $\mathcal{L}(y_1)$ is set to $\mathcal{L}(y_1) \cup \{C\}$. The $\forall_\backslash$-Rule is applicable twice to y_0 and $\mathcal{L}(y_1)$ is set to $\mathcal{L}(y_1) \cup \{D\} \cup \{\neg D\}$. A clash is detected since $\{D, \neg D\} \in \mathcal{L}(y_1)$ and the solution where $v_{\mathrm{R_1}} \geq 1$ is discarded. This means that the partition for $\alpha(v_{\mathrm{R_1}})$ must be empty and $v_{\mathrm{R_1}}$ must be ≤ 0. However with $v_{\mathrm{R_1}} \leq 0$ no possible solution for the set of inequations in $\mathcal{L}_E(y_0)$ is possible and a clash is detected. All possible distributions result in a clash and the TBox is not consistent.

5.3.1 Some Illustration

The TBoxes in Examples 1, 2 are simple TBoxes used to illustrate the features of the hybrid approach proposed in this paper. The TBox $\mathcal{T}$ in Example 1 contains cyclic descriptions, nominals and QCRs and is typical to highlight the strong features of the hybrid approach proposed in this paper. The TBox $\mathcal{T}$ in Example 2 shows the applicability and effect of the new constructor $\forall_{\backslash}$.

- In contrast to other tableau algorithms, a tree model property with cycle detection techniques is not crucial for termination.
- Nodes are never merged or pruned which means that we do not need to handle the so-called "yoyo" effect or manage all incoming and outgoing edges of nodes.
- When applying the algorithm with $\mathcal{T} = \{A \sqsubseteq (\geq nR.(A \sqcap (\geq nR.A)))$ for large values of n the behavior of the algorithm is not affected. This makes the extension of the algorithm to more expressive logics more promising.[11]
- By non-deterministically initializing the nominal nodes and by using a proxy individual as a representative of n individuals we can avoid a large number of completion rules to be unnecessarily triggered. A minimum number of role fillers is considered since the inequation solver returns a minimal solution.
- The inequation solver facilitates early clash (Definition 5 (ii)) detection.

It is easy to see that these features of the hybrid algorithm make it novel and we conjecture it is well suited for optimizing DL reasoning with nominals and QCRs.

6 Proof of Correctness and Termination

The soundness, completeness and termination of the algorithm presented in this report are consequences of Lemmas 3, 5, 6, 7, and Lemma 8.

Lemma 5 *Given a TBox $\mathcal{T}$ and its complete and clash-free completion graph G. Let x be a node in G, $C, D \in N_{\mathrm{C}}$, $R \in N_{\mathrm{R}}$, we define $Num(x) = \{E \in \mathcal{L}(x) \,|\, E$ is of the form $\geq nR, \leq mR\}$ as the set of at-least and at-most restrictions to be satisfied for x. A solution σ for the encoding ξ_x of $Num(x)$ is valid w.r.t. $\mathcal{T}$:*

- *(i) it does not violate $Num(y)$ for a node y in G,*
- *(ii) it does not violate a restriction implied by any operator used in $\mathcal{T}$,*
- *(iii) it does not violate the hierarchy $\mathcal{R}$ introduced during preprocessing.*

Proof. Our hybrid algorithm depends on a sound, complete, and terminating inequation solver to get a minimal solution σ for ξ_x. The expansion rules construct a model for role fillers and nominals based on the distribution reflected by σ. The algorithm needs to make sure that σ is consistent with $\mathcal{R}$ and $\mathcal{T}$.

[11] Large values of n are known to be problematic for most DL reasoners supporting at least $\mathcal{SHQ}$.

- (i) Since $\mathcal{L}_{\mathrm{E}}(x)$ is propagated through all nodes, all numerical restrictions including the ones encoded in $\mathcal{L}_{\mathrm{E}}(y)$ are always satisfied.
- (ii) If the distribution is not consistent with logical restrictions in $\mathcal{T}$ such as the ones implied by the $\forall$ operator, then for some $\forall R.D \in \mathcal{L}(x)$, we have a node y such that $R \in \mathcal{L}(y)$ and $\neg D \in \mathcal{L}(y)$. The $\forall$-Rule becomes applicable to x and D is added to $\mathcal{L}(y)$. Having $\{D, \neg D\} \subseteq \mathcal{L}(y)$ is not possible since G is clash-free. By analogy we prove that σ does not violate other restrictions implied by the $\forall_{\setminus}$, and ($\sqcap$, $\sqcup$, $\neg$) operators.
- (iii) If the distribution is not consistent with $\mathcal{R}$, then for some $(R' \sqsubseteq R) \in \mathcal{R}$, there exists an R'*-filler* y assigned to a partition P with $R' \in P$ and $P^{\mathcal{I}} \subseteq (FIL(R') \setminus FIL(R))$. This case is not possible due to the definition of H(R) which assumes that R is implied in P whenever $R' \in P$ and $R' \in H(R)$.

Lemma 6 (Termination) *When started with an $\mathcal{ALCOQ}$ TBox $\mathcal{T}$, the proposed hybrid algorithm terminates.*

Proof. Let $l = \#clos(\mathcal{T})$, termination of the hybrid algorithm is guaranteed due to the following.

- The rewriting in Algorithm 1 can be done in linear time and does not affect termination.
- The atomic decomposition computes the partitions in $\mathcal{P}$ w.r.t. $\mathcal{T}$, in the worst case $\#\{N_{\mathrm{R}} \cup N_{\mathrm{o}}\} = l$ and the size of $\mathcal{P}$ is $2^l - 1$ since we do not consider the empty partition. Although this computation is exponential, it is done only once.
- Getting a distribution of individuals (solution for the inequations) will not affect termination of the expansion rules since we assume a decidable arithmetic reasoner [21].
- The algorithm constructs a graph consisting of a set of arbitrarily interconnected nodes by applying expansion rules which do not remove nodes from the graph, nor remove concepts from node labels or edge labels. For each node x:
  - the number of times that the *fil*-Rule, or the *ch*-Rule can be applied is bounded by the size of $\mathcal{P}$. In the worst case we need to create one individual for each partition.
  - the number of times the *e*-Rule is applied for each at-least restriction is bounded by n (the largest number used in an at-least restriction). In the worst case individuals satisfying $\geq nR$ are distributed into n partitions. Therefore, the total number that this rule can be applied is bounded by $l * n$.
  - all other rules are applied at most l times.
- New nodes are created by the *fil*-Rule depending on a distribution of individuals into partitions in $\mathcal{P}$. For each partition we create at most one proxy node, and since partitions are disjoint, the total number of nodes in the completion graph is bounded by the size of $\mathcal{P}$. It is not possible to have more nodes in the graph since each node is either a nominal or a role filler and in both cases it must be in some partition in $\mathcal{P}$.

- Traditional termination problems such as cyclic TBoxes and "yo-yo" problems are not encountered:
  - cyclic definitions do not cause a termination problem since nodes having the same label (case when blocking is needed with other algorithms) will eventually be mapped to the same partition and only one proxy node is created. This also justifies why in these cases we have less individuals and why we do not need any blocking strategies.
  - the "yo-yo" problem of infinitely creating and merging nodes cannot occur since in a given model, nodes are never removed or merged.

Lemma 7 (Soundness) *If the expansion rules can be applied to $\mathcal{T}$ such that they yield a complete and clash-free completion graph, then $\mathcal{T}$ has a tableau.*

Proof. A tableau $\mathrm{T} = (\mathbf{S}, \mathcal{L}', \mathcal{E})$ can be obtained from a clash-free completion graph $\mathrm{G} = (V, E, \mathcal{L}, \mathcal{L}_{\mathrm{E}})$ by mapping nodes in G to individuals in T which can be defined from G as T such that: $\mathbf{S} = V \setminus \{r_0\}$, $\mathcal{L}'(x) = (\mathcal{L}(x) \setminus N_{\mathrm{R}})$, and $\mathcal{E}(R) = \{\langle x, y\rangle \in E \mid (H(R) \cup \{R\}) \cap \mathcal{L}(\langle x, y\rangle) \neq \emptyset\}$. We show that T is either a tableau or can be easily extended to a tableau for $\mathcal{T}$ since properties 1 - 10 of a tableau (see Def. 3) are either satisfied or can be easily satisfied.

- Properties 1, 2, 3, and 4 of a tableau are satisfied because G is clash-free and complete.
- Property 5: Assume $\forall S.C \in \mathcal{L}'(x)$ and $\langle x, y\rangle \in \mathcal{E}(S)$ then $C \in \mathcal{L}'(y)$, otherwise the $\forall$-Rule would be applicable. Property 6 is similarly satisfied.
- Properties 7: Assume $(\geq nS) \in \mathcal{L}'(x)$ then completeness of G implies that there exist j proxy individuals $y_1 \dots y_j$ each representing a partition of m_j individual such that $\sum_{i=1}^{j} m_i = n$ and $S \in \mathcal{L}(\langle x, y_i\rangle)$ $(1 \leq i \leq j)$. Due to Lemmas 4 and 5, we can replicate each y_i, $m_i - 1$ times and set $\mathbf{S} = \mathbf{S} \cup \{y_{i_k}\}$ and $\mathcal{L}(\langle x, y_{i_k}\rangle) = S$ with $1 \leq k \leq m_i - 1$, then we have $\#S^{\mathcal{T}}(x) \geq n$ and property 7 is satisfied. One might think that replicating individuals might result in violating the nominals semantics (Property 10) for example by replicating a nominal individual. However, this case can never happen since nominals are represented by proxy individuals y_i belonging to a partition with only one individual, $m_i = 1$ always holds for nominals partitions and is encoded by the inequations (see Property 10 below). Similarly, Property 8 cannot be violated due to replication of individuals; partition sizes (m_i) are assigned such that all at-least and at-most restrictions are satisfied (See Property 8 below).
- Property 8: Assume $(\leq mS) \in \mathcal{L}'(x)$ and $\#S^{\mathcal{T}}(x) \leq m$ is violated. This means that we have j proxy individuals $y_1 \dots y_j$ each representing a partition of m_j individual such that $\sum_{i=1}^{j} m_i > m$. This case cannot happen for two reasons:
  - (1) By having Priority 2 for the *fil-Rule*, nodes are created only after making sure that all at-least and at-most restrictions for a node x are satisfied by having a distribution of these fillers (a non-negative integer solution for the inequations in $\mathcal{L}_E(x)$). This means that no nodes will be created that violate an at-most restriction.

  - (2) G is clash free which means that for each $(\leq mS) \in \mathcal{L}(x)$ we have $\xi(S, \leq, m)$ in $\mathcal{L}_E(x)$ and there is no $\xi(S, \geq, n)$ in $\mathcal{L}_E(x)$ and $n > m$.

- Property 9: This property is always satisfied due to the definition of $H(R)$ which takes into account the role hierarchies introduced at re-writing.

- Property 10: This property is satisfied since G is initialized by setting $\xi(o, \geq 1)$ and $\xi(o, \leq 1)$ for each nominal $o \in N_{\text{o}}$ and every nominal o is assigned to a partition with only one individual. A node with o in its label is created by the *fil*-Rule. Since G is clash free then condition (iii) of a clash (see Definition 5) can never hold. In addition, no nodes that are created can be removed or merged. Therefore, the set of nodes with a nominal o in their label satisfies property 10. Also, since no nominal individual can be replicated to satisfy Property 7, Property 10 always holds.

Lemma 8 (Completeness) *If $\mathcal{T}$ has a tableau, then the expansion rules can be applied to $\mathcal{T}$ such that they yield a complete and clash-free completion graph.*

Proof. Let $\text{T} = (\mathbf{S}, \mathcal{L}', \mathcal{E})$ be a tableau for $\mathcal{T}$, T can be used to guide the application of the expansion rules. We define the mapping function π from nodes in the graph $G = (V, E, \mathcal{L}, \mathcal{L}_{\text{E}})$ to individuals in $\mathbf{S}$, inductively with the creation of new nodes, such that for each $x, y \in V$, a role $R, S \in N_{\text{R}}$ and a partition name $p \in \mathcal{P}$ we have:

1. $\mathcal{L}(x) \subseteq \mathcal{L}'(\pi(x)) \cup \mathcal{P}$
2. if $\langle x, y\rangle \in E$ and $S \in \mathcal{L}(\langle x, y\rangle)$, then $\langle \pi(x), \pi(y)\rangle \in \mathcal{E}(S)$
3. $\xi(R, \geq, n) \in \mathcal{L}_{\text{E}}(x)$ implies $\#R^{\mathcal{T}}(\pi(x)) \geq n$
4. $\xi(R, \leq, n) \in \mathcal{L}_{\text{E}}(x)$ implies $\#R^{\mathcal{T}}(\pi(x)) \leq n$

The claim is that having a completion graph G that satisfies the properties of π we can apply the expansion rules defined in Fig. 2, when applicable, to G without violating the properties of π. Initially G consists of the artificial node r_0 such that $\bigcup_{o \in N_{\text{o}}}\{\xi(o, \geq, 1), \xi(o, \leq, 1)\} \subseteq \mathcal{L}_E(r_0)$ and at least one node x_0 with some $o \in \mathcal{L}(x_0)$ is created. Given a tableau T for G, we can set $s_0 = \pi(x_0)$ for some $s_0 \in \mathbf{S}$.

We show that whenever we can apply an expansion rule to G, the properties of π are not violated: applying the $\sqcap$-Rule, $\sqcup$-Rule, $\forall$-Rule or $\forall_{(\backslash)}$-Rule strictly extends the label of a node x and this does not violate properties of π due to properties 1-6 of a tableau. Let us consider applying the other rules to a given node x:

- The ***ch*-Rule**: This rule strictly extends the system of inequations that is in the label $\mathcal{L}_E$ of a node x. It non-deterministically assigns a cardinality (0 or ≥ 1) for role filler partitions that are not assigned any value, and therefore no properties of π can be violated.

- The **$\geq$-Rule** and **$\leq$-Rule**: If $(\geq nR)$, $(\leq mR) \in \mathcal{L}(x)$, then $(\geq nR)$, $(\leq mR) \in \mathcal{L}'(\pi(x))$, this implies that $\#R^{\mathcal{T}}(\pi(x)) \geq n$, $\#R^{\mathcal{T}}(\pi(x)) \leq m$, (properties 6 and 7 of a tableau). Applying the $\geq$-Rule, $\leq$-Rule extends $\mathcal{L}_E(x)$ with $\xi(R, \geq, n)$, $\xi(R, \leq, m)$ which is still conform with the properties of π and those of a tableau.

- The ***fil*-Rule**: If every $(\geq nR)$, $(\leq mR) \in \mathcal{L}(x)$ is converted to $\xi(R, \geq, n)$, $\xi(R, \leq, m) \in \mathcal{L}_E(x)$ and due to the clash freeness of T this means that there exists a distribution of role fillers satisfying every $(\geq nR)$, $(\leq mR) \in \mathcal{L}(x)$. The distribution of fillers is encoded in a solution σ for $\mathcal{L}_E(x)$ and applying the *fil*-Rule which creates a proxy individual y as a representative for each corresponding partition based on the solution, and propagates the set of inequations in $\mathcal{L}_E(x)$ to y does not violate π.

- The ***e-Rule***: For each $(\geq nR) \in \mathcal{L}(x)$ we have $(\geq nR) \in \mathcal{L}'(\pi(x))$ which means that $\#R^{\mathcal{T}}(\pi(x)) \geq n$ must be satisfied. The e-Rule is applied to connect x to its *R-fillers* such that with each j^{th} $(1 \leq j \leq n)$ application of this rule an edge is created between x and some proxy individual y_j with y_j representing m_j (the number of elements assigned to a partition by the inequation solver) individuals of a partition p.

  Then, after all edges are created we have j proxy *R-fillers* each representing m_j individuals such as $\sum_{i=1}^{j} m_i \geq n$. Due to Lemmas 5 and 4 we can replicate each y_j, $m_i - 1$ times and by setting $\mathcal{L}(\langle x, y_{i_k}\rangle) = \{R\}$ with $1 \leq i \leq j$ and $1 \leq k \leq m_i - 1$ and by setting $\pi = \pi[y_{1_1} \rightarrow t_{1_1} \ldots y_{i_k} \rightarrow t_{i_k}]$ with $t_{1_1} \ldots t_{i_k}$ tableau elements in T satisfying $\#R^{\mathcal{T}}(\pi(x)) \geq n$. We can see that $\#R^{\mathcal{T}}(\pi(x)) \geq n$ is satisfied without violating π.

The resulting graph G is clash free due to the following:

1. G cannot contain a node x such that $\{A, \neg A\} \subseteq \mathcal{L}(x)$ since $\mathcal{L}(x) \subseteq \mathcal{L}'(\pi(x))$ and Property 2 of the definition of a tableau would be violated.

2. G cannot contain two nodes x and y such that $\{A, \neg A\} \subseteq \mathcal{L}(y)$ due to the following scenario

   - Initially $\{\forall R.A, \forall R\backslash S.\neg A\} \subseteq \mathcal{L}(x)$ and y a proxy node for a partion of *R-fillers* and *S-fillers* of x, and
   - after applying the *e-Rule* for some $\geq nR \in \mathcal{L}(x)$ and the $\forall$-*Rule* for $(\forall R.A) \in \mathcal{L}(x)$, y is an *R-filler* of x with $\{A\} \subseteq \mathcal{L}(y)$, and
   - after applying the $\forall_\backslash$-*Rule* for $(\forall R\backslash S.A) \in \mathcal{L}(x)$ we have $\{A, \neg A\} \subseteq \mathcal{L}(y)$ with y an *R-filler* of x.

   This case cannot happen. Due to the strategy of rule applications in section the 5.2.1, the $\forall_\backslash$-*Rule* cannot be applied if the $\geq mS$ is already applicable. The rule priorities make sure that the $\forall(R\backslash S)$ semantics is enforced only when no more nodes can become *S-fillers* of x and Properties 5 and 6 of the definition of a tableau are preserved.

3. G cannot contain a node x such that $\mathcal{L}_{\mathrm{E}}(x)$ is unsolvable. If $\mathcal{L}_{\mathrm{E}}(x)$ is unsolvable, this means that for some role $R \in N_{\mathrm{R}}$ we have:

   - $\{\xi(R, \geq, n)\} \subseteq \mathcal{L}_{\mathrm{E}}(x)$, and there is no possible distribution of *R-fillers* satisfying $\geq nR \subseteq \mathcal{L}(x)$, hence property 7 of a tableau would be violated due to the equivalence properties between $\xi(R, \geq, n) \in \mathcal{L}_{\mathrm{E}}(x)$ and $\#R^{\mathcal{T}}(\pi(x)) \geq n$ respectively, or
   - $\{\xi(R, \leq, n)\} \subseteq \mathcal{L}_{\mathrm{E}}(x)$, and there is no possible distribution of *R-fillers* satisfying $\leq mR$ hence property 8 of a tableau would be violated due to the equivalence properties between $\xi(R, \leq, m) \in \mathcal{L}_{\mathrm{E}}(x)$ and $\#R^{\mathcal{T}}(\pi(x)) \leq m$.

4. G cannot contain two nodes x and y such that for some nominal $o \in N_o$ we have $o \in \mathcal{L}(x) \cap \mathcal{L}(y)$, otherwise property 10 of a tableau would be violated.

Thus, the completeness of our hybrid algorithm is proved.

7 Conclusion

In this report we presented an alternative approach for designing DL reasoning algorithms. By studying the interaction between nominals and QCRs we designed a hybrid calculus that is more informed about the numerical features of nominals and that supports the DL $\mathcal{ALCOQ}$ with general TBoxes. The calculus can be easily extended to work with role hierarchies and transitive roles reaching the expressiveness of $\mathcal{SHOQ}$.

Finally, it is easy to see that a naïve implementation of this algorithm is probably as bad as a naïve implementation of standard tableau algorithms. One has to address the exponential blow up of variables used to represent cardinalities of sets and the high degree of non-determinism that they introduce if implemented naïvely. On the other hand, promising evidence to improve average case performance for the DL $\mathcal{SHQ}$ is reported in [7] where the use of variables has been optimized. It is part of ongoing work to implement and evaluate the performance of the hybrid algorithm as presented in this report.

References

[1] Alan Rector, G. S. Qualified cardinality restrictions QCRs: Constraining the number of values of a particular type for a property.

[2] Baader, F., Buchheit, M., and Hollunder, B. Cardinality restrictions on concepts. *Artificial Intelligence 88*, 1-2 (1996), 195–213.

[3] Baader, F., and Sattler, U. An overview of tableau algorithms for description logics. *Studia Logica 69* (2001), 5–40.

[4] Blackburn, P., and Marx, M. Third int. workshop on hybrid logic HyLo'01. *Logic Journal of the IGPL 9*, 5 (2001).

[5] Faddoul, J., Farsinia, N., Haarslev, V., and Möller, R. A hybrid tableau algorithm for $\mathcal{ALCQ}$. In *18th European Conference on Artificial Intelligence (ECAI 2008)* (2008), pp. 725–726.

[6] Faddoul, J., Farsinia, N., Haarslev, V., and Möller, R. A hybrid tableau algorithm for $\mathcal{ALCQ}$, (long version). In *Proc. of the 2008 Int. Workshop on Description Logics* (2008).

[7] Farsinia, N. thesis title. Master's thesis, Concorida University, 2008. Forthcoming.

[8] Haarslev, V., and Möller, R. Optimizing reasoning in description logics with qualified number restrictions. In *Description Logics* (2001).

[9] Haarslev, V., Timmann, M., and Möller, R. Combining tableaux and algebraic methods for reasoning with qualified number restrictions. In *Description Logics* (2001), pp. 152–161.

[10] Hollunder, B., and Baader, F. Qualifying number restrictions in concept languages. In *Proc. of the 2nd Int. Conference on Principles of Knowledge Representation and Reasoning,KR-91* (Boston (USA), 1991), pp. 335–346.

[11] Horrocks, I. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003, ch. Implementation and optimisation techniques, pp. 306–346.

[12] Horrocks, I., and Sattler, U. Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)* (2001), Morgan Kaufmann, Los Altos, pp. 199–204.

[13] Horrocks, I., and Sattler, U. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)* (2005), pp. 448–453.

[14] Horrocks, I., and Sattler, U. A tableaux decision procedure for $\mathcal{SHOIQ}$. *Journal of Automated Reasoning* (2007), 249–276.

[15] Kazakov, Y., and Motik, B. *A Resolution-Based Decision Procedure for $\mathcal{SHOIQ}$*, vol. 4130/2006. Springer Berlin / Heidelberg, 2006, pp. 662–677.

[16] Marieb, E. N., Hoehn, K., Wilhelm, P. B., and Zanetti, N. *Human Anatomy & Physiology: International Edition with Human Anatomy and Physiology Atlas, 7/E*, 7 ed. Pearson Higher Education, 2006.

[17] Motik, B., and Horrocks, I. Individual reuse in description logic reasoning. In *IJCAR* (2008), pp. 242–258.

[18] Motik, B., Shearer, R., and Horrocks, I. Optimized reasoning in description logics using hypertableaux. In *(CADE-21)* (2007), vol. 4603 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 67–83.

[19] Ohlbach, H. J., and Koehler, J. Reasoning about sets via atomic decomposition. Tech. Rep. TR-96-031, Berkeley, CA, 1996.

[20] Ohlbach, H. J., and Koehler, J. Role hierarchies and number restrictions. In *Proceedings of the 1997 International Workshop on Description Logics (DL'97)* (1997).

[21] Ohlbach, H. J., and Koehler, J. Modal logics description logics and arithmetic reasoning. *Artificial Intelligence 109*, 1-2 (1999), 1–31.

[22] Parsia, B., Cuenca Grau, B., and Sirin, E. From wine to water: Optimizing description logic reasoning for nominals. In *Proc. of the 10th Int. Conference on Principles of Knowledge Representation and Reasoning (KR2006)* (2006), pp. 90–99.

[23] Schaerf, A. Reasoning with individuals in concept languages. *Data and Knowledge Engineering 13*, 2 (1994), 141–176.

[24] Tobies, S. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research 12* (2000), 199–217.