Rational Grading in an Expressive Description Logic

Mitko Yanchev

Faculty of Mathematics and Informatics, Sofia University, Bulgaria
yanchev@fmi.uni-sofia.bg

Abstract. In this paper syntactic objects—concept constructors called *part restrictions* which realize rational grading are considered in Description Logics (DLs). Being able to convey statements about a rational part of a set of successors, part restrictions essentially enrich the expressive capabilities of DLs. We examine an extension of well-studied DL $\mathcal{ALCQIH}_{R^+}$ with part restrictions, and prove that the reasoning in the extended logic is still decidable. The proof uses tableaux technique augmented with *indices technique*, designed for dealing with part restrictions.

1 Introduction

Description Logics (DLs) are widely used in knowledge-based systems. The representation in the language of transitive relations, in different possible ways [11], is important for dealing with complex objects. *Transitive roles* permit such objects to be described by referring to their components, or ingredients without specifying a particular level of decomposition. The expressive power can be strengthened by allowing additionally *role hierarchies*. The DL $\mathcal{ALCH}_{R^+}$ [6], an extension of well-known DL $\mathcal{ALC}$ with both transitive roles and role hierarchies, is shown to be suitable for implementation. Though having the same EXPTIME-complete worst-case reasoning complexity as other DLs with comparable expressivity, it is more amendable to optimization [5].

Inverse roles enable the language to describe both the whole by means of its components and vice versa, for example has_part and is_part_of. This syntax extension is captured in DL $\mathcal{ALCIH}_{R^+}$ [7]. As a next step, in [7] the language is enriched with the counting (or *grading*—a term coming from the modal counterparts of DLs [4]) *qualifying number restrictions*, what results in DL $\mathcal{ALCQIH}_{R^+}$. It is given a sound and complete decision procedure for that logic.

We go further considering concept constructors which we call *part restrictions*, capable of distinguishing a rational part of a set of successors. These constructors are analogues of the modal operators for *rational grading* [12] which generalize the majority operators [10]. They are $MrR.C$ and (the dual) $WrR.C$, where r is a rational number in $(0, 1)$, R is a role, and C is a concept. The intended meaning of $MrR.C$ is '(strongly) More than r-part of R-successors (or R-*neighbours*, in the presence of inverse roles) of the current object possess the property C'. Part restrictions essentially enrich the expressive capabilities of DLs. From the 'object domain' point of view they seem to be more 'socially' than 'technically' oriented, but in any case they give new strength to the language. An example of the use of part restrictions is the concept $\mathsf{M}\frac{2}{3}\,\mathsf{voted.Yes}$ which expresses the notion of qualifying majority in a voting system.

On the other hand, *presburger constraints* in the language of extended modal logic EXML [3], a language with only independent relations, capture both integer and rational grading, and have rich expressiveness. The rational grading modalities are expressible by the presburger constraints, and the satisfiability of EXML is shown to be in PSPACE. Another constraints on role successors which subsume the part restrictions are introduced in [2] using the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic. The corresponding DL $\mathcal{ALCSCC}$ also captures both integer and rational grading, and it is shown that the complexity of reasoning in it is the same as in $\mathcal{ALCQ}$ (an extension of $\mathcal{ALC}$ with qualifying number restrictions), both without and with TBoxes.

A combinatorial approach to grading in modal logics [9] uses the so called *majority digraphs*. In this approach, in addition to integer and rational grading, also the grading with real coefficients can be expressed.

Nonetheless, the use of separate rational grading, having its place also in modal logics, proves markedly beneficial in DLs. Part restrictions can be combined in a DL with many other constructors. *Indices technique*, specially designed for exploring the part restrictions, allows following a common way for obtaining decidability and complexity results as in less, so in more expressive languages with rational grading. In particular, reasoning complexity results—polynomial, NP, and co-NP—concerning a range of description logics from the $\mathcal{AL}$-family with part restrictions added, are obtained ([14], [13], [15]), as well as PSPACE-results for modal and expressive description logics ([16], [17], [18]).

Now we consider the DL $\mathcal{ALCQPIH}_{R^+}$, in which the language of $\mathcal{ALCQIH}_{R^+}$ is augmented with part restrictions. We use the tableaux technique to prove that the reasoning in the extended logic is also decidable.

2 Syntax and Semantics of $\mathcal{ALCQPIH}_{R^+}$

The $\mathcal{ALCQPIH}_{R^+}$-syntax and semantics differ from those of $\mathcal{ALCQIH}_{R^+}$ only in the presence of part restrictions.

Definition 1. *Let* $\mathbf{C}_o \neq \emptyset$ *be a set of* concept names*,* $\mathbf{R}_o \neq \emptyset$ *be a set of* role names*, some of which transitive, and* $\mathbf{Q}_0$ *be a set of rational numbers in* $(0, 1)$*. We denote the set of transitive role names* $\mathbf{R}^+$*, so that* $\mathbf{R}^+ \subseteq \mathbf{R}_o$*. Then we define the set of* $\mathcal{ALCQPIH}_{R^+}$-roles *(we will refer to simply as 'roles') as* $\mathbf{R} = \mathbf{R}_o \cup \{R^- \mid R \in \mathbf{R}_o\}$*, where* R^- *is the inverse role of* R*.*

As the inverse relation on roles is symmetric, to avoid considering roles such as R^{--} *we define a function* Inv *which returns the inverse of a role. Formally,* $\mathsf{Inv}(R) = R^-$*, if* R *is a role name, and* $\mathsf{Inv}(R^-) = R$*. Thus,* $\mathsf{Inv}(\mathsf{Inv}(R)) = R$*.*

A role inclusion axiom *has the form* $R \sqsubseteq S$*, for two roles* R *and* S*, and the acyclic inclusion relation* $\sqsubseteq$*. For a set of role inclusion axioms* $\mathcal{R}$*, a* role hierarchy *is* $\mathcal{R}^+ := \big(\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\},\ \sqsubseteq^+\big)$, *where* $\sqsubseteq^+$ *is the reflexive and transitive closure of* $\sqsubseteq$ *over* $\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}$.

A role R *is* simple *with respect to* $\mathcal{R}^+$ *iff* $R \notin \mathbf{R}^+$ *and, for any* $S \sqsubseteq^+ R$, S *is also simple w.r.t.* $\mathcal{R}^+$.

The set of $\mathcal{ALCQPIH}_{R^+}$-concepts *(we will refer to simply as 'concepts') is the smallest set such that: 1. every concept name is a concept; 2. if* C *and* D *are concepts, and* R *is a role, then* $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, *and* $\exists R.C$ *are concepts; 3. if* C *is a concept,* R *is a simple role,* $n \geq 0$, *and* $r \in \mathbf{Q}_0$, *then* $\geqslant nR.C$, $\leqslant nR.C$, $MrR.C$, *and* $WrR.C$ *are concepts.*

The limitation roles in qualifying number restrictions, as well as in part restrictions to be simple is used essentially in the proofs. From the other side, the presence in the language of role hierarchies together with only number restrictions on transitive roles leads to undecidability [8]

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a nonempty set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$ which maps every concept to a subset of $\Delta^{\mathcal{I}}$ and every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, is defined in a standard way.[1] We only set the additional restriction for any object $x \in \Delta^{\mathcal{I}}$ and any role $R \in \mathbf{R}$ the set of objects, $R^{\mathcal{I}}$-related to ($R^{\mathcal{I}}$-*neighbours* of) x, denoted $R^{\mathcal{I}}(x)$, to be finite. $R^{\mathcal{I}}(x, C)$ denotes the set $\{y \mid \langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$ of $R^{\mathcal{I}}$-neighbours of x which are in $C^{\mathcal{I}}$, and $\sharp M$ denotes the cardinality of a set M. For part restrictions, for any concept C, simple role R, and $r \in \mathbf{Q}_0$ the definitions of mapping are:

$$(MrR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp R^{\mathcal{I}}(x, C) > r.\sharp R^{\mathcal{I}}(x)\}$$
$$(WrR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \sharp R^{\mathcal{I}}(x, \neg C) \leq r.\sharp R^{\mathcal{I}}(x)\} \ \Big(= (\neg MrR.\neg C)^{\mathcal{I}}\Big)$$

An interpretation $\mathcal{I}$ *satisfies a role hierarchy* $\mathcal{R}^+$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ for any $R \sqsubseteq^+ S \in \mathcal{R}^+$; we denote that by $\mathcal{I} \models \mathcal{R}^+$.

A concept C is *satisfiable* with respect to a role hierarchy $\mathcal{R}^+$ iff there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{R}^+$ and $C^{\mathcal{I}} \neq \emptyset$. Such an interpretation is called a *model of* C *with respect to* $\mathcal{R}^+$. For an object $x \in C^{\mathcal{I}}$ we say that x *satisfies* C, also that x is an *instance* of C, while $x \in \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$ *refuses* C.

Thus, for $x \in \Delta^{\mathcal{I}}$, x is in $(MrR.C)^{\mathcal{I}}$ iff strictly greater than r part of $R^{\mathcal{I}}$-neighbours of x satisfies C, and x is in $(WrR.C)^{\mathcal{I}}$ iff no greater than r part of $R^{\mathcal{I}}$-neighbours of x refuses C.

A concept D *subsumes* a concept C with respect to $\mathcal{R}^+$ (denoted $C \sqsubseteq_{\mathcal{R}^+} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for every interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{R}^+$.

Checking the subsumption between concepts is the most general reasoning task in DLs. From the other side, $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable. Thus, in

[1] All definitions and techniques from Section 5 of [7] concerning $\mathcal{ALCQIH}_{R^+}$ are applicable to the extended DL, eventually with only mild changes. So, in what follows we present explicitly, due to the restriction of space, only what is new, or changed, relying on and referring to [7] for the rest. The complete definition of the interpretation, the complete sets of tableaux properties and completion rules, and all proofs can be seen in [19].

the presence of negation of an arbitrary concept, checking the (un)satisfiability becomes as complex as checking the subsumption.

In what follows we consider concepts to be in the *negation normal form* (NNF). We denote the NNF of $\neg C$ by $\sim C$. The NNF of $\sim MrR.C$ is $WrR.\neg C$, and, dually, $\sim WrR.C = MrR.\neg C$. For any concept C in NNF we denote with $clos(C)$ the smallest set of concepts containing C and closed under sub-concepts and $\sim$. The size of $clos(C)$ is linear to the size of C. With $\mathbf{R}_C$ we denote the set of roles occurring in C and their inverses.

3 A Tableau for $\mathcal{ALCQPIH}_{R^+}$

We will use a tableaux algorithm to test the satisfiability of a concept. We extend the definition of $\mathcal{ALCQIH}_{R^+}$-tableau by modifying one property to reflect the presence of part restrictions, and adding two new ones. Thus we obtain a definition of a tableau for $\mathcal{ALCQPIH}_{R^+}$.

Definition 2. *A* tableau T *for a concept D in NNF with respect to a role hierarchy $\mathcal{R}^+$ is a triple* $(\mathbf{S}, \mathcal{L}, \mathcal{E})$*, where $\mathbf{S}$ is a set of individuals, $\mathcal{L} : \mathbf{S} \to 2^{clos(D)}$ is a function mapping each individual of $\mathbf{S}$ to a set of concepts which is a subset of $clos(D)$, $\mathcal{E} : \mathbf{R}_D \to 2^{\mathbf{S}\times\mathbf{S}}$ is a function mapping each role occurring in $\mathbf{R}_D$ to a set of pairs of individuals, and there is some individual $s \in \mathbf{S}$ such that $D \in \mathcal{L}(s)$. For all individuals $s, t \in \mathbf{S}$, concepts in $clos(D)$, and roles in $\mathbf{R}_D$, T must satisfy 13 properties.*

We denote with $R^T(s)$ the set of individuals, R-related to s, and $R^T(s, C) := \{t \in \mathbf{S} \mid \langle s, t\rangle \in \mathcal{E}(R) \text{ and } C \in \mathcal{L}(t)\}$. The new and the modified properties follow. In property 13 (modified property 11 from the definition of $\mathcal{ALCQIH}_{R^+}$-tableau), and in what follows, $\boxtimes$ is a placeholder, besides for $\geqslant n$ and $\leqslant n$, for arbitrary $n \geq 0$, also for $\exists$, and for Mr and Wr, for arbitrary $r \in \mathbf{Q}_0$.

11. If $MrR.C \in \mathcal{L}(s)$, then $\sharp R^T(s, C) > r.\sharp R^T(s)$.
12. If $WrR.C \in \mathcal{L}(s)$, then $\sharp R^T(s, \sim C) \leq r.\sharp R^T(s)$.
13. If $\boxtimes\, R.C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(R)$, then $C \in \mathcal{L}(t)$ or $\sim C \in \mathcal{L}(t)$.

Having the definition of $\mathcal{ALCQPIH}_{R^+}$-tableau, we can prove Lemma 1 following the standard way, also for the new and modified properties.

Lemma 1. *An $\mathcal{ALCQPIH}_{R^+}$-concept D is satisfiable with respect to a role hierarchy $\mathcal{R}^+$ iff there exists a tableau for D with respect to $\mathcal{R}^+$.*

4 Constructing an $\mathcal{ALCQPIH}_{R^+}$-Tableau

Lemma 1 guarantees that the algorithm constructing tableaux for $\mathcal{ALCQPIH}_{R^+}$-concepts can serve as a decision procedure for concept satisfiability (and hence, also for subsumption between concepts) with respect to a role hierarchy $\mathcal{R}^+$. We present such an algorithm.

As usual with the tableaux algorithms, $\mathcal{ALCQPIH}_{R^+}$-algorithm tries to prove the satisfiability of a concept D by constructing a *completion tree* (c.t. for short) $\mathbf{T}$, from which a tableau for D can be build. Each node x of the tree is labelled with a set of concepts $\mathcal{L}(x)$ which is a subset of $clos(D)$, and each edge $\langle x, y\rangle$ is labelled with a set of roles $\mathcal{L}(\langle x, y\rangle)$ which is a subset of $\mathbf{R}_D$. The algorithm starts with a single node (the c.t. root) x_0 with $\mathcal{L}(x_0) = \{D\}$, and the tree is then expanded by completion rules, which decompose the concepts in the nodes' labels, and add new nodes and edges, giving the relationships between nodes, and new labels to the nodes and edges.

A node y is an *R-successor* of a node x if y is a successor of x and $S \in \mathcal{L}(\langle x, y\rangle)$ for some S with $S \sqsubseteq^+ R$; y is an *R-neighbour* of x if it is an R-successor of x, or if x is an $\mathsf{Inv}(R)$-successor of y.

We denote with $R^{\mathbf{T}}(x)$ the set of R-neighbours of a node x in the c.t. $\mathbf{T}$, and with $R^{\mathbf{T}}(x, C)$—the set of R-neighbours of x in $\mathbf{T}$ which are labelled with C.

A c.t. $\mathbf{T}$ is said to contain a *clash* (i.e., the obvious contradiction) if, for some node x in $\mathbf{T}$, a concept C, a role R, some $n \geq 0$, and some $r \in \mathbf{Q}_0$ any of the following is the case. Otherwise it is *clash-free*.

$CL1$. $\{C, \neg C\} \subseteq \mathcal{L}(x)$

$CL2$. $\leqslant nR.C \in \mathcal{L}(x)$ and $\sharp R^{\mathbf{T}}(x, C) > n$

$CL3$. $MrR.C \in \mathcal{L}(x)$ and $\sharp R^{\mathbf{T}}(x, C) \leq r.\sharp R^{\mathbf{T}}(x)$

$CL4$. $WrR.C \in \mathcal{L}(x)$ and $\sharp R^{\mathbf{T}}(x, \neg C) > r.\sharp R^{\mathbf{T}}(x)$

A completion tree is *complete* if none of the completion rules is applicable, or if, for some node x, $\mathcal{L}(x)$ contains a clash of type $CL1$ or type $CL2$.[2]

If, for a concept D, the completion rules can be applied in a way to yield a complete and clash-free completion tree, then the algorithm returns '*D is satisfiable*'; otherwise, it returns '*D is unsatisfiable*'.

During the expansion the algorithm uses the *pair-wise blocking* technique as defined in [7], sections 4.1 and 5.3, to ensure only finite paths in the completion tree. It also uses *indices technique* which will be presented in details, to prevent from infinite branching of the tree (possibly) caused by part restrictions.

Figure 1 presents the completion rules which are new or modified in comparison with ones in the $\mathcal{ALCQIH}_{R^+}$-algorithm. *choose-rule* is augmented (via the placeholder ⊠) to add also labels, induced by ∃-concepts and part restrictions.

In the presence of part restrictions, *⩾-rule* which adds all the necessary successors at ones leads to incompleteness.[3] So, it is modified to add successors one by one, thus preventing the occurrence of redundant neighbours. This needs

[2] Part restrictions talk about no exact quantities, but ratios. So, instances of $CL3$ and $CL4$ (which are also conditions for applicability of M-rule and W-rule, see Figure 1) can appear and disappear dynamically during the c.t. generation. That is why we exclude them from the definition of the c.t. completeness.

[3] For example, the concept $A \sqcap \exists R^-.(\geqslant 4R.\top \sqcap \leqslant 5R.\top \sqcap M\frac{2}{5}R.(\neg A) \sqcap W\frac{1}{2}R.A)$, where $\top = A \sqcup \neg A$, and A is a concept name, has a unique tableau (modulo labelling of the individuals) with just four individuals. But no complete and clash-free c.t. can be built from that tableau using the 'all-at-once' *⩾-rule*.

some modification of $\leqslant$-*rule* also. $\leqslant$-rule transfers the label of an edge to just one other edge. So, the use of $\mathcal{L}(\langle x,y\rangle) \cap \mathcal{L}(\langle x,z\rangle) = \emptyset$ condition in the rule is justified as follows. If two edges are labelled with the same role, it has been labelling initially (even if some label transfer has happened meanwhile) two different edges connecting x with two of its neighbours. The possible cases are: 1) the labels of y and z are different and contradict each other for any labelling by *choose*-rule, so y and z cannot be merged; 2) the labels of y and z are different but there is a labelling by *choose*-rule which makes them not contradicting, then there is no need nodes to be merged, as when this labelling is made to the firstly generated node, the second one would not be generated at all; 3) the labels of y and z are identical, then the generation of both nodes is triggered by $\geqslant n$-concept with $n \geq 2$, M-, or W-concept, or, anyway, they are used for the satisfying in the c.t. of such a concept, so they *must not* be merged.

M-*rule* and W-*rule* (the *part rules*) are new *generating rules* (in addition to $\exists$-rule and $\geqslant$-rule) which deal with part restrictions. The rest of the rules—$\sqcap$-, $\sqcup$-, $\exists$-, $\forall$-, and $\forall_+$-rule—remain just as they are in [7], Section 5, Figure 5.

choose-rule:	If 1. $\boxtimes R.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. there is an R-neighbour y of x with $\{C, \sim C\} \cap \mathcal{L}(y) = \emptyset$ then $\mathcal{L}(y) \to \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \sim C\}$
$\geqslant$-rule:	If 1. $\geqslant nR.C \in \mathcal{L}(x)$, x is not blocked, and 2. $\sharp R^{\mathbf{T}}(x, C) < n$ then create a new successor y of x with $\mathcal{L}(\langle x,y\rangle) = \{R\}$ and $\mathcal{L}(y) = \{C\}$
$\leqslant$-rule:	If 1. $\leqslant nR.C \in \mathcal{L}(x)$, x is not indirectly blocked, and 2. $\sharp R^{\mathbf{T}}(x, C) > n$ and there are two R-neighbours y and z of x with $C \in \mathcal{L}(y)$, $C \in \mathcal{L}(z)$, $\mathcal{L}(\langle x,y\rangle) \cap \mathcal{L}(\langle x,z\rangle) = \emptyset$, and y is not a predecessor of x then 1. $\mathcal{L}(z) \to \mathcal{L}(z) \cup \mathcal{L}(y)$, 2. If z is a predecessor of x then $\mathcal{L}(\langle z,x\rangle) \to \mathcal{L}(\langle z,x\rangle) \cup \{\mathsf{Inv}(S) \mid S \in \mathcal{L}(\langle x,y\rangle)\}$ else $\mathcal{L}(\langle x,z\rangle) \to \mathcal{L}(\langle x,z\rangle) \cup \mathcal{L}(\langle x,y\rangle)$ 3. $\mathcal{L}(\langle x,y\rangle) \to \emptyset$
M-rule:	If 1. $MrR.C \in \mathcal{L}(x)$, x is not blocked, and 2. $\sharp R^{\mathbf{T}}(x, C) \leq r.\sharp R^{\mathbf{T}}(x)$ then calculate BAN_x, and if 3. $\sharp R^{\mathbf{T}}(x) < BAN_x$ then create a new successor y of x with $\mathcal{L}(\langle x,y\rangle) = \{R\}$ and $\mathcal{L}(y) = \{C\}$
W-rule:	If 1. $WrR.C \in \mathcal{L}(x)$, x is not blocked, and 2. $\sharp R^{\mathbf{T}}(x, \sim C) > r.\sharp R^{\mathbf{T}}(x)$ then calculate BAN_x, and if 3. $\sharp R^{\mathbf{T}}(x) < BAN_x$ then create a new successor y of x with $\mathcal{L}(\langle x,y\rangle) = \{R\}$ and $\mathcal{L}(y) = \{C\}$

Fig. 1. The new and the modified completion rules

We impose a *rule application strategy*: any generating rule can be applied only if all *non-generating rules* (i.e., $\sqcap$-, $\sqcup$-, $\forall$-, $\forall_+$-, *choose*- and $\leqslant$-rule) are inapplicable. Anyway, the generation process is non-deterministic in both which rule (from the group of non-generating or generating ones) to be applied, and which concept(s) to be chosen in the non-deterministic $\sqcup$-, *choose*-, and $\leqslant$-rule.

The rule application strategy is essential for the successful 'work' of $\leqslant$-rule, and for the part rules. It ensures that a) all concepts 'talking' about neighbours are already present in $\mathcal{L}(x)$, and b) all possible (re)labelling of neighbours of x is done before the application of a part rule. Both are necessary for applying the indices technique for the correct generation of successors, caused by part restrictions. The check-up in part rules (in 3.) for not reaching the *border amount of neighbours* for the current node x (BAN_x) is a kind of 'horizontal blocking' of the generation process, used to ensure inapplicability of part rules after a given moment. The notion is crucial for the termination of the algorithm, and its use is based on Lemma 6, which is the upshot of the indices technique.

5 Indices Technique

We develop a specific technique, which we call *indices technique*, to cope with the presence of part restrictions. This technique permits to extend appropriately the definition of a clash, to design completion rules, dealing with part restriction, and to give an adequate rule application strategy, as they are presented in the previous section, all to guarantee the correctness of the tableaux algorithm.

5.1 The clashes with part restrictions

$CL3$ and $CL4$, which are also conditions for applicability of part rules, are dynamic. Applied consecutively, part rules can 'repair' one clash, and, at the same time, provoke another. Thus, instances of $CL3$ and $CL4$ can appear and disappear, in some cases infinitely, during the c.t. generation, even if the initial concept is satisfiable. So, we have to take special care both to ensure the termination of part rules application, and not to leave avoidable 'part' clashes in the completion tree. That turns out to be the main difficulty in designing the algorithm. We overcome it by proving that if it is possible to unfold part restrictions at a given node avoiding *simultaneously* both kinds of clashes, it can be done within some number of neighbours. As clashes are always connected with a single node, talking about its label and its neighbours, that is enough to guarantee the termination. The following subsection presents the technique in details.

5.2 Counteracting part restrictions. Clusters

We start our analysis with the simplest case when, for a node x of the c.t. $\mathbf{T}$, there are in $\mathcal{L}(x)$ only part restrictions, and they all are with the same role R, and with sub-concepts which are either a fixed concept C, or its negation $\sim C$, and x is not an $\mathsf{Inv}(R)$-successor. All such part restrictions form the set:

$$\{Mr_1 R.C,\ Mr_2 R.{\sim}C,\ Wr_3 R.C,\ Wr_4 R.{\sim}C\} \quad (1)$$

We call the subset of (1) which is in $\mathcal{L}(x)$ a *cluster of R and C at x in* $\mathbf{T}$, and we denote it $Cl_x^{\mathbf{T}}(R,C)$. It is obvious, that during the generation of (R-) successors of x (if it is necessary) instances of $CL3$ and $CL4$ can appear only if two contradicting part restrictions are in that cluster.

Definition 3. *A part restriction which is in the label of a node x in a c.t.* $\mathbf{T}$ *is* $\mathbf{T}$-satisfied *(at x) if there is no clash with it at x.*

A cluster is $\mathbf{T}$-satisfied *if all part restrictions in it are* $\mathbf{T}$*-satisfied.*

A part restriction (a cluster) is c.t.-satisfiable *if it can be* $\mathbf{T}$*-satisfied, for some c.t.* $\mathbf{T}$.

In fact, in (1) there can be more than one part restriction of any of the four types. But note that, if $MrR.C$ is $\mathbf{T}$-satisfied, then that is the case with $Mr'R.C$ (being in the label of the same node), for any $r' < r$. So, we can take r_1 and r_2 to be the maximums, and, by analogues reasons, r_3 and r_4 to be the minimums of the r-s in part restrictions of the corresponding types. Thus we obtain the upper, representative for the c.t.-satisfiability of all part restrictions in the node's label, set with only four ones.

The idea behind the c.t.-satisfiability is that if a cluster, and more general, the set of all part restrictions labelling a node, is c.t.-satisfiable, then a c.t. without clashes with part restrictions at that node can be non-deterministically generated, while the part rules become inapplicable for this node (as the inequality in condition 2 or 3 in part rules becomes false). So, concerning part restrictions, c.t.-satisfiability is a sufficient condition for obtaining a clash-free complete c.t.

Our next observation is that both $Mr_1R.C$ and $Wr_3R.C$ act in the same direction concerning c.t. generation, as the former forces the addition of enough R-successors of x labelled with C, and the latter limits the number of R-successors of x labelled with $\sim C$. The same holds for $Mr_2R.{\sim}C$ and $Wr_4R.{\sim}C$ with respect to $\sim C$. At that time, as $Mr_1R.C$, so $Wr_3R.C$ counteract with any of $Mr_2R.{\sim}C$ and $Wr_4R.{\sim}C$. This leads to two main possibilities for $Cl_x^{\mathbf{T}}(R,C)$:

A. The cluster contains only part restrictions, acting in the same direction (or just a single one)—we call it *cluster of type* A, or A-cluster. In the absence of counteracting part restrictions these clusters are always c.t.-satisfiable.

B. The cluster contains at least two counteracting part restrictions—we call it *cluster of type* B, or B-cluster.

In order the c.t. generation process to be able to c.t.-satisfy a B-cluster, and to avoid $CL3$ and $CL4$ clashes, the next inequalities between the r-s in the cluster (or between the *indices*, from where we take the name of indices technique) must be fulfilled—follows directly from the semantics of part restrictions, the above remarks about counteractions, and the definition of c.t.-satisfiability:

1° $r_1 + r_2 < 1$	4° $r_3 + r_4 \geq 1$, what is
2° $r_1 < r_4$	(a) $r_3 + r_4 > 1$, or
3° $r_2 < r_3$	(b) $r_3 + r_4 = 1$

If any of the inequalities 1°–4° does not hold, any complete c.t. will contain a clash, as it is impossible to c.t.-satisfy simultaneously (at the same node) the part restrictions in which are the indices, taking part in the failed inequality.

We can combine that four inequalities in just one taking into account the kind of interaction between part restrictions. $Wr_3R.C$ means that $\sim C$ has to label not greater than r_3 part of all R-neighbours of x, i.e., that C has to label at least $(1 - r_3)$ part of them. We set $\check{r} = \max\bigl(\{r_1, 1 - r_3\}\bigr)$ (or, if the part restriction with r_1 or r_3 is not in the cluster, $\check{r}$ is just the expression with the other). Now, it is obvious that if C labels greater than $\check{r}$ part of all R-neighbours of x, then both $Mr_1R.C$ and $Wr_3R.C$ are (or the single one from the couple which is in the cluster, is) c.t.-satisfied. Analogues reasonings go with the other couple of part restrictions, acting in the same direction (the ones with Mr_2 and Wr_4), and (a part smaller than) $\hat{r} = \min(\{1 - r_2, r_4\})$.

We call *dominating* the part restrictions which determine $\check{r}$ and $\hat{r}$.

It is important to note that $r_3 + r_4 = 1$ does not spoil the c.t.-satisfiability (unlike $r_1 + r_2 = 1$). We exclude that case from the general examination, as a special sub-case, and discuss it separately. Thus, B divides into two sub-cases:
B(a). The cluster contains no counteracting W part restrictions, or $r_3 + r_4 \neq 1$.
B(b). The cluster contains counteracting W part restrictions and $r_3 + r_4 = 1$.

Clusters of type B(a). Our first claim is:

Lemma 2. *For a B(a)-cluster $Cl^{\mathbf{T}}_x(R, C)$, the inequalities* 1°, 2°, 3°, *and* 4°(a), *with the corresponding part restrictions being in the cluster, hold iff $\check{r} < \hat{r}$.*

Corollary 1. *$\check{r} < \hat{r}$ is a necessary condition for the c.t.-satisfiability of a B(a)-cluster $Cl^{\mathbf{T}}_x(R, C)$.*

The upper inequality is also a *sufficient condition* for a cluster's c.t.-satisfiability. Indeed, if $\check{r} < \hat{r}$, and the number of R-neighbours of x labelled with C—$|R^{\mathbf{T}}(x, C)|$—is strongly (due to the strong inequality for M) between $\check{r}.|R^{\mathbf{T}}(x)|$ and $\hat{r}.|R^{\mathbf{T}}(x)|$, then the dominating part restrictions are c.t.-satisfied, and so are the rest of the part restrictions in the cluster, if any. This shows that $\check{r} < \hat{r}$ guarantees the c.t.-satisfiability; practical c.t.-satisfaction of a cluster depends on the number of neighbours, and, of course, their appropriate labelling.

Note also that even though $\check{r} < \hat{r}$ holds, we can have *instable* c.t.-satisfaction, as it can be seen from the next example. Let the dominating part restrictions be $M\frac{2}{3}R.C$ and $M\frac{1}{4}R.{\sim}C$. They can be **T**-satisfied if $R^{\mathbf{T}}(x)$ has 10 nodes (with C labelling 7, and ${\sim}C$—3 of them), and also 11 nodes (with labelling $C : {\sim}C$—$8 : 3$), while if $R^{\mathbf{T}}(x)$ has 12 nodes, there is no way these part restrictions to be **T**-satisfied, as the first wants C to label at least 9, and the second—${\sim}C$ to label at least 4 R-neighbours of x. In case of 13 R-neighbours of x the part restrictions again can be simultaneously **T**-satisfied.

Definition 4. *A cluster $Cl^{\mathbf{T}}_x(R, C)$ is* n-satisfiable*, where $n \geq 0$, if it can be c.t.-satisfied when x has exactly n R-neighbours.*

A cluster is stably n-satisfiable*, if it is n-satisfiable, and for any natural number $n' > n$ it is also n'-satisfiable.*

A cluster is stably c.t.-satisfiable*, if it is stably n-satisfiable for some $n \geq 0$.*

Note that from the above definition it follows that if a cluster is stably n-satisfiable, it is also stably n'-satisfiable, for any natural number $n' > n$.

In the example above the cluster is 10-, and 11-satisfiable, it is not 12-satisfiable, and it is (in fact—stably) 13-satisfiable.

So, if we have a sufficient condition for stable n-satisfiability of B(a)-clusters, we will know exactly when, in the non-deterministic c.t. generation process, stable c.t.-satisfaction of such a cluster will be achieved in at least one non-deterministic generation (we call it a *successful* generation). Then we will be able to key at that moment the part rules with respect to the part restrictions of that cluster, thus avoiding infinite rules application in the unsuccessful generations.

Lemma 3. *Let, for a B(a)-cluster $Cl_x^{\mathbf{T}}(R,C)$, $\check{r} < \hat{r}$ hold. Then a sufficient condition for the non-deterministic $|R^{\mathbf{T}}(x)|$-satisfiability of the cluster is:*

$$|R^{\mathbf{T}}(x)| > \tfrac{1}{\hat{r}-\check{r}} \quad (\sharp)$$

Lemma 4. *Let, for a B(a)-cluster $Cl_x^{\mathbf{T}}(R,C)$, $\check{r} < \hat{r}$ and ($\sharp$) hold, and the dominating part restrictions in the cluster be $\mathbf{T}$-satisfied. Then any generating rule can always be applied in a way to yield $\mathbf{T}'$ such that the cluster to be $\mathbf{T}'$-satisfied.*

Lemma 4 shows that ($\sharp$) also guarantees the stability of the non-deterministic c.t.-satisfiability, namely stable $\left(\left\lfloor\frac{1}{\hat{r}-\check{r}}\right\rfloor+1\right)$-satisfiability. Being once fulfilled, ($\sharp$) holds for any greater number of R-neighbours of x, and so c.t.-satisfying of the dominating part restrictions can be preserved as $R^{\mathbf{T}}(x)$ grows.

Thus, Lemma 3 and Lemma 4 guarantee for a c.t.-satisfiable (with $\check{r} < \hat{r}$) B(a)-cluster $Cl_x^{\mathbf{T}}(R,C)$ that, having the number of R-neighbours of x equal to, or greater than $\left\lfloor\frac{1}{\hat{r}-\check{r}}\right\rfloor+1$ (what we will call the *border amount of neighbours* of x, BAN_x, for that cluster), the cluster can be non-deterministically c.t.-satisfied. Then, the termination of application of rules, triggered by (the part restrictions in) that cluster, is ensured by the check-up for $|R^{\mathbf{T}}(x)|$.

Shortly said, any c.t.-satisfiable B(a)-cluster can be non-deterministically stably c.t.-satisfied when the node has enough many neighbours on the role in the cluster. We will rate that in the general case for all (possibly counteracting) part restrictions, to preserve from infinite application of part rules.

Clusters of type B(b). B(b)-clusters are determined by the equality 4°(b) $r_3+r_4=1$ for the indices in W part restrictions. These clusters are c.t.-satisfiable if 2° $r_1 < r_4$ and 3° $r_2 < r_3$ hold (in case that the corresponding M part restrictions are in the cluster; in that case 1° obviously also holds). Thus, if 2° and 3° hold, or some M part restriction is missing, 4°(b) can be considered as a sufficient condition for the c.t.-satisfiability of a B(b)-cluster.

Lemma 5. *Let, for a B(b)-cluster $Cl_x^{\mathbf{T}}(R,C)$, $r_1 < r_4$ and $r_2 < r_3$ hold, in case the corresponding M part restrictions are in the cluster. Then the cluster is c.t.-satisfiable, and the sufficient condition it to be non-deterministically c.t.-satisfied is the number of R-neighbours of x to be devisable by the denominator of r_3 and r_4.*

The general case. Let us recall that the application of a part rule requires all possible applications of non-generating rules for the current node to be already done, what ensures all possible (at the moment) concepts, including part restrictions, to be already present in the node's label. Generalizing the considerations for counteracting in clusters, also taking into account the other concepts, triggering generating rules, and using the indices technique, we prove:

Lemma 6. *Let x be a node of a completion tree $\mathbf{T}$, and let all possible applications of non-generating rules for x be done. Then it can be calculated a natural number $BAN_x \geq 1$, depending on the concepts in $\mathcal{L}(x)$, and if x is a successor of u, possibly also depending on the concepts in $\mathcal{L}(u)$, and having the following property: all part restrictions in $\mathcal{L}(x)$ which are simultaneously $\mathbf{T}$-satisfiable can be non-deterministically simultaneously $\mathbf{T}$-satisfied when the number of neighbours of x on any role at the uppermost level in these part restrictions becomes equal to BAN_x.*

Lemma 6 both legitimates the use of BAN_x in the part rules applicability check-up, thus ensuring termination, and guarantees that all simultaneously c.t.-satisfiable part restrictions will be non-deterministically c.t.-satisfied, so that there would not be clashes with them in the complete c.t.

Note that the border amount of neighbours can change only if $\mathcal{L}(x)$ or $\mathcal{L}(u)$ be changed, for example by adding of some concept to any of them caused by an application of a rule for a successor. As the number of such possible changes is limited by the number of concepts in $clos(D)$, after finite number of recalculations we will obtain the final for the node x BAN_x.

6 Correctness of the Algorithm

As usual with tableaux algorithms we prove lemmas that the algorithm always terminates, and that it is sound and complete. The termination is ensured by pair-wise blocking, and by BAN-checkup which guarantees finite (at most exponential—in case of the usual binary coding of numbers) branching at a node. The build of a tableau from the completion tree and the reverse follows the constructions from [7], Section 5.4, Lemmas 16 and 17. Since the *internalization of terminologies* [1] is still possible in the presence of part restrictions, following the technique presented in [7], Section 3.1, we obtain finally:

Theorem 1. *The presented tableaux algorithm is a decision procedure for the satisfiability and subsumption of $\mathcal{ALCQPIH}_{R^+}$-concepts with respect to role hierarchies and terminologies.*

7 Conclusion

DL $\mathcal{ALCQPIH}_{R^+}$ augments $\mathcal{ALCQIH}_{R^+}$ with the ability to express rational grading. We showed that the decision procedure for the latter logic can be naturally extended to capture the new one. This indicates that the approach which realizes rational grading independently from integer grading is fruitful, and can be applied even to expressive description logics to give in a convenient way their rational grading extensions, still keeping the decidability.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable questions, remarks and suggestions.

References

1. F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. DFKI Research Report RR-90-13, DFKI, Kaiserslautern, 1990.
2. F. Baader. A new description logic with set constraints and cardinality constraints on role successors. In *FroCoS*, pages 43–59, 2017.
3. S. Demri and D. Lugiez. Presburger modal logic is PSPACE-complete. In *LNCS 4130*, Automated Reasoning (IJCAR 2006), London, UK, 2006. Springer-Verlag.
4. K. Fine. In so many possible worlds. *Notre Dame Journal of formal logic*, 13(4):516–520, 1972.
5. I. Horrocks. Optimisation techniques for expressive description logics. Technical report, 1997.
6. I. Horrocks and G. Gough. Description logics with transitive roles. In M.-C. Rousset, R. Brachman, F. Donini, E. Franconi, I. Horrocs, and A. Levy, editors, *Proc. of DL'97*, DL'97, pages 25–28, 1997.
7. I. Horrocks, U. Sattler, and S. Tobies. A description logic with transitive and converse roles, role hierarchies and qualifying number restrictions. LTCS-Report 99-08, LuFg Theoretical Computer Science, RWTH Aachen, Germany, 1999.
8. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings of the 6th International Conference on Logic Programming and Automated Reasoning*, LPAR '99, pages 161–180, London, UK, 1999. Springer-Verlag.
9. T. Lai, J. Endrullis, and L. S. Moss. Majority Digraphs. *ArXiv e-print 1509.07567*, September 2015.
10. E. Pacuit and S. Salame. Majority logic. In *Book of Abstracts*, Principles of Knowledge Representation and Reasoning (KR 2004), pages 598–605, 2004.
11. U. Sattler. A concept language extended with different kinds of transitive roles. In G. Görz and S. Hölldobler, editors, *20. Deutsche Jahrestagung für Künstliche Intelligenz*, number 1137 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1996.
12. T. Tinchev and M. Yanchev. Modal operators for rational grading. In *Book of Abstracts*, International Conference *Pioneers of Bulgarian Mathematics*, pages 123–124, Sofia, Bulgaria, 2006.
13. M. Yanchev. Part restrictions: adding new expressiveness in description logics. In *Abstracts of Informal Presentations*, CiE 2012, page 144, 2012. Available online at `http://www.mathcomp.leeds.ac.uk/turing2012/WScie12/Images/abstracts-booklet.pdf`.
14. M. Yanchev. Part restrictions in description logics: reasoning in polynomial time complexity. In *Contributed Talk Abstracts*, LC 2012, pages 38–39, 2012. Available online at `http://www.mims.manchester.ac.uk/events/workshops/LC2012/abs/contrib.pdf`.
15. M. Yanchev. Part restrictions in description logics with union and counting constructors. In *Collection of Abstracts*, CiE 2013, page 43, 2013. Available online at `https://cie2013.wordpress.com/2013/06/29/downloadable-material/`.
16. M. Yanchev. Complexity of generalized grading with inverse relations and intersection of relations, 2014. Presented at LC 2014. Abstract available online at `http://www.easychair.org/smart-program/VSL2014/LC-2014-07-15.html#talk:2053`.
17. M. Yanchev. A description logic with part restrictions: PSPACE-complete expressiveness. In A. Beckmann, E. Csuhaj-Varjú, and K. Meer, editors, *Collection of*

Unpublished Abstracts, CiE 2014, pages 261–270, Budapest, Hungary, 2014. Available online at https://www.fmi.uni-sofia.bg/about/lio/yanchev/ct_cie2014.
18. M. Yanchev. PSPACE reasoning in an expressive description logic with rational grading. In *Proceedings*, 11th Panhellenic Logic Symposium, pages 102–107, Delphi, Greece, 2017.
19. M. Yanchev. Decidability of an expressive description logic with rational grading, 2019. arXiv:1905.10338 [cs.LO].