A Tableau Algorithm for Description Logics with Nominal Schema*

Adila Krisnadhi** and Pascal Hitzler

Kno.e.sis Center, Wright State University, Dayton OH
adila@knoesis.org, pascal.hitzler@knoesis.org

Abstract. We present a tableau algorithm for the description logic $\mathcal{ALCOV}$. This description logic is obtained by extending the description logic $\mathcal{ALCO}$ with the expressive nominal schema construct that enables DL-safe datalog with predicates of arbitrary arity to be covered within the description logic framework. The tableau algorithm provides a basis to implement a delayed grounding strategy which was not facilitated by earlier versions of decision procedures for satisfiability in expressive description logics with nominal schemas.

1 Introduction

The quest for suitable ontology languages for the Semantic Web [1] has produced a plethora of proposals drawing from all corners of KR research. Most notable among the expressive languages is the W3C[1] standard Web Ontology Language (OWL) [2] whose major variant, OWL 2 DL, is based on the description logic (DL) $\mathcal{SROIQ}$ [3]. At the same time, logic programming based approaches such as F-Logic [4] have also been investigated prominently, and have eventually led to the W3C standard Rule Interchange Format (RIF), which in its core variant, called RIF Core [5], is essentially Datalog, i.e., function-free Horn logic.

This divergence in underlying paradigms, i.e., DLs on the one hand and logic programming on the other, has naturally led to substantial efforts to reconcile them in a satisfactory manner (see the related work section of [6] for a recent survey). However, satisfactory integrations are not easy to obtain, which is mainly due to the fact that description logics are generally supposed to be decidable, and because straightforward integrations with rules lead to undecidability.

The most prominent approach to date to overcome this decidability issue is to alter the semantics of rule bases in such combined languages, and to do this in such a way that variables occurring in rules can bind only to constants which are explicitly present in the knowledge base—in DL lingua, these variables can

* This work is supported by the National Science Foundation (NSF) project "TROn – Tractable Reasoning with Ontologies" under the award 1017225 III: Small.

** The first author acknowledges the support of Fulbright Indonesia Presidential Scholarship PhD Grant. 2010–2013

[1] World Wide Web Consortium, http://www.w3.org/

bind only to *known individuals*. This reading of (Datalog) rules is only a mild violation of the usual semantics, because such rules are usually interpreted under a Herbrand (minimal model) semantics anyway. Datalog rules with semantics modified such as just mentioned are commonly called *DL-safe Rules*, and decidability is retained if such rules are added to DL knowledge bases [7].

More recently, however, it was realized that this notion of DL-safety can be relaxed such that only some of the variables in the rules have to be affected [8]. And very recently [9], the description logic syntax construct *nominal schemas* was introduced, which not only further generalizes the notion of DL-safety, but also allows one to express the DL-safe rules (and their generalizations) with predicates of arbitrary arity within the description logic syntax, using the new construct (see Appendix of [10]). It was also shown that extending standard DLs with nominal schemas does not affect the decidability. In fact, in case of $\mathcal{SROIQ}$, adding nominal schemas does not increase the complexity. This decidability result, however, hinged on a rather straightforward but inefficient algorithmization based on *full grounding upfront* which induces an exponential blow up already at the beginning of reasoning (see Section 3.1). It is thus necessary to have a smarter alternative which can be potentially more practical to implement.

In this paper, we explore one such alternative by adapting standard tableau algorithm for DLs to work with nominal schemas while allowing for a *delayed grounding* strategy. The rationale is that by delaying the grounding until it is absolutely necessary to do so can minimize the blow up compared to performing it fully at the beginning of reasoning. In order to make the presentation easier to follow, we focus on the DL $\mathcal{ALCOV}$ — obtained by extending $\mathcal{ALCO}$ with nominal schemas — which is less expressive than $\mathcal{SROIQV}$. The employed approach should not be too difficult to generalize for other DLs more expressive than $\mathcal{ALCOV}$, including $\mathcal{SROIQV}$.

The plan of the paper is as follows. We first define the syntax and semantics of $\mathcal{ALCOV}$. Then we present the tableau algorithm and prove its correctness. Afterward, we proceed with discussion on how to extend the approach to more expressive DLs. Finally, we conclude.

2 The DL $\mathcal{ALCOV}$

Let N_C, N_R, N_I, and N_V be pairwise disjoint sets of *concept names*, *role names*, *individual names*, and *variables*.

Definition 1. *The set* $\mathbf{C}$ *of* ($\mathcal{ALCOV}$) concepts *is defined by the grammar:*

$$\mathbf{C} ::= \top \mid \bot \mid \mathsf{N}_C \mid \{\mathsf{N}_I\} \mid \{\mathsf{N}_V\} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \exists \mathsf{N}_R.\mathbf{C} \mid \forall \mathsf{N}_R.\mathbf{C}$$

A general concept inclusion (GCI) *is an expression* $C \sqsubseteq D$ *where* $C, D \in \mathbf{C}$. *A* TBox *is a finite set of GCIs. An* ABox assertion *is either (i) a* concept assertion *which is an expression of the form* $C(a)$*; (ii) a* role assertion *which is an expression of the form* $R(a, b)$*; (iii) a* negative role assertion *which is an expression of the form* $\neg R(a, b)$*; or (iv) an* inequality assertion *which is an*

expression of the form $a \not\doteq b$; where $C \in \mathbf{C}$, $a, b \in \mathsf{N}_I$, and $R \in \mathsf{N}_R$. An ABox *is a finite set of ABox assertions. An* axiom *is either a GCI or an ABox assertion. A* knowledge base (KB) *is a finite set of axioms.*

For every concept C, $\mathsf{Var}(C)$ is the set of all variables that occurs, possibly more than once, in C. Likewise, $\mathsf{Ind}(C)$ is the set of all individual names occurring in C. Both sets are naturally extended to sets of concepts, axioms and knowledge bases.

The syntax of $\mathcal{ALCOV}$ only differs from that of $\mathcal{ALCO}$ by the presence of the set N_V of variables and the corresponding concept expression of the form $\{x\}$ for $x \in \mathsf{N}_V$ which is called *nominal schemas* which generalize the notion of DL-safety. In this regards, nominal schema is a kind of variable nominal, that is, an $x \in \mathsf{N}_V$ should be read as a variable which can bind only to *known* individuals, i.e., elements of N_I that appear explicitly in the KB. Since we only care about variables and individual names appearing in the KB, we simply assume henceforth that $\mathsf{N}_I = \mathsf{Ind}(\mathcal{K})$ and $\mathsf{N}_V = \mathsf{Var}(\mathcal{K})$ where $\mathcal{K}$ is the KB in consideration, and any complexity analysis based on $\mathcal{K}$ will take the signature into account.

Definition 2 (Variable Assignment). *A* variable assignment *is a (total or partial) function θ from N_V to N_I. The domain of θ is denoted $\mathsf{dom}(\theta)$. If $\mathsf{dom}(\theta)$ only consists of one variable v, i.e., it is a singleton set, then we sometimes write θ as $[v/\theta(v)]$. If $\mathsf{dom}(\theta) = \mathsf{N}_V$, then θ is a total function and we say that θ is a* total variable assignment*. For a variable assignment θ (not necessarily total) and a set of variables $M \subseteq \mathsf{dom}(\theta)$, the* restriction of θ w.r.t. M *is the variable assignment $\theta|_M$ defined as $\{x \mapsto \theta(x) \mid x \in M\}$.*

Definition 3 (Grounding). *Let C be a concept. We say that C is* ground *whenever $\mathsf{Var}(C) = \emptyset$. Further, given a variable assignment θ, the concept $C\theta$ is obtained from C by syntactically substituting all occurrences of every variable $x \in \mathsf{Var}(C) \cap \mathsf{dom}(\theta)$ with $\theta(x)$. In particular, $C\theta = C$ whenever $\mathsf{Var}(C) \cap \mathsf{dom}(\theta) = \emptyset$. Also, note that $C\theta$ is ground when $\mathsf{Var}(C) \subseteq \mathsf{dom}(\theta)$, and we say that $C\theta$ is a* ground form *of C (w.r.t. θ).*

The application of a variable assignment as defined above is called a grounding*. This definition is extended naturally to sets of concepts, axioms, or knowledge bases.*

Note that there are $|\mathsf{N}_I|^{|\mathsf{N}_V|} = |\mathsf{Ind}(\mathcal{K})|^{|\mathsf{Var}(\mathcal{K})|}$, i.e., exponentially many total variable assignment for a given knowledge base $\mathcal{K}$. Thus, grounding $\mathcal{K}$ will yield a knowledge base that is exponentially larger than $\mathcal{K}$.

The semantics of $\mathcal{ALCOV}$ concepts is defined as in standard DLs, while also taking into account variable assignments for nominal schemas. Essentially, the semantics of a concept with nominal schemas is based on its ground form, hence at least one total variable assignment is required whenever a nominal schema appear anywhere in the knowledge base. To this end, we assume that if $\mathsf{N}_V \neq \emptyset$, then also $\mathsf{N}_I \neq \emptyset$. This is somewhat similar to the way rules are interpreted in Herbrand semantics. As discussed in Section 1, nominal schemas are a generalization of DL-safety which is used to ensure the decidability of

integration between rules and description logics. Hence, the above assumption is also in accordance with the aim of rule-DL integration because Herbrand semantics also assumes that at least one constant symbol exists.

Definition 4. *An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$ called the* domain *and a function $\cdot^{\mathcal{I}}$ that maps each $a \in \mathsf{N}_I$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each $A \in \mathsf{N}_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each $R \in \mathsf{N}_R$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Given a total variable assignment θ and an interpretation $\mathcal{I}$, the semantics of $\mathcal{ALCOV}$ concepts, roles and individual names is defined w.r.t. the function $\cdot^{\mathcal{I},\theta}$ in the following where $a \in \mathsf{N}_I$, $A \in \mathsf{N}_C$, $R \in \mathsf{N}_R$ $v \in \mathsf{N}_V$, $t \in \mathsf{N}_I \cup \mathsf{N}_V$ and $C, D \in \mathbf{C}$.*

$$
\begin{aligned}
A^{\mathcal{I},\theta} &= A^{\mathcal{I}} & R^{\mathcal{I},\theta} &= R^{\mathcal{I}} \\
a^{\mathcal{I},\theta} &= a^{\mathcal{I}} & v^{\mathcal{I},\theta} &= (\theta(v))^{\mathcal{I},\theta} \\
\top^{\mathcal{I},\theta} &= \top^{\mathcal{I}} = \Delta^{\mathcal{I}} & \bot^{\mathcal{I},\theta} &= \bot^{\mathcal{I}} = \emptyset \\
\{t\}^{\mathcal{I},\theta} &= \{t^{\mathcal{I},\theta}\} & (\neg C)^{\mathcal{I},\theta} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\theta} \\
(C \sqcap D)^{\mathcal{I},\theta} &= C^{\mathcal{I},\theta} \cap D^{\mathcal{I},\theta} & (C \sqcup D)^{\mathcal{I},\theta} &= C^{\mathcal{I},\theta} \cup D^{\mathcal{I},\theta} \\
(\exists R.C)^{\mathcal{I},\theta} &= \{\delta \mid \exists \epsilon\colon \langle \delta, \epsilon \rangle \in R^{\mathcal{I},\theta} \text{ and } \epsilon \in C^{\mathcal{I},\theta}\} \\
(\forall R.C)^{\mathcal{I},\theta} &= \{\delta \mid \forall \epsilon\colon \text{ if } \langle \delta, \epsilon \rangle \in R^{\mathcal{I},\theta} \text{ then } \epsilon \in C^{\mathcal{I},\theta}\}
\end{aligned}
$$

*Given an interpretation $\mathcal{I}$ and a total variable assignment θ, we say that $\mathcal{I}$ θ-*satisfies*:*

(i) a GCI $C \sqsubseteq D$, written $\mathcal{I} \models_\theta C \sqsubseteq D$, iff $C^{\mathcal{I},\theta} \subseteq D^{\mathcal{I},\theta}$;
(ii) a concept assertion $C(a)$, written $\mathcal{I} \models_\theta C(a)$, iff $a^{\mathcal{I},\theta} \in C^{\mathcal{I},\theta}$;
(iii) an assertion $R(a,b)$, written $\mathcal{I} \models_\theta R(a,b)$, iff $\langle a^{\mathcal{I},\theta}, b^{\mathcal{I},\theta} \rangle \in R^{\mathcal{I},\theta}$;
(iv) an assertion $\neg R(a,b)$, written $\mathcal{I} \models_\theta \neg R(a,b)$, iff $\langle a, b \rangle \notin R^{\mathcal{I},\theta}$; and
(v) an assertion $a \not\doteq b$, written $\mathcal{I} \models_\theta a \not\doteq b$, iff $a^{\mathcal{I},\theta} \neq b^{\mathcal{I},\theta}$.

*For a KB $\mathcal{K}$ (i.e., a TBox, an ABox or a union of them), $\mathcal{I}$ θ-*satisfies *$\mathcal{K}$, written $\mathcal{I} \models_\theta \mathcal{K}$, iff $\mathcal{I} \models_\theta \alpha$ for every $\alpha \in \mathcal{K}$. Further, $\mathcal{I}$* satisfies *$\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I} \models_\theta \mathcal{K}$ for every total variable assignment θ. If such an $\mathcal{I}$ exists, we say that $\mathcal{K}$ is* satisfiable *and $\mathcal{I}$ is a* model *of $\mathcal{K}$. Finally, the* KB-satisfiability problem *is the following reasoning problem: "Given a KB $\mathcal{K}$, is $\mathcal{K}$ satisfiable?".*

Herein, axioms are assumed to be *standardized apart*: for every pair of different axioms α_1 and α_2 in a KB $\mathcal{K}$, $\mathsf{Var}(\alpha_1) \cap \mathsf{Var}(\alpha_2) = \emptyset$. This is analogous to the way variables in rules are interpreted: the scope of a variable is within a rule, i.e., if the same variable occurs in two different rules, they are considered to be different (although syntactically the same). Standardizing apart simplifies the treatment of nominal schemas as we do not need to specify to which variable in which axiom is a grounding intended.

Note that the *unique name assumption* (UNA) is not made since it can be easily enforced by simply asserting inequalities in the ABox between every pair of individual names. Clearly, there is only at most a quadratic blow up of the size of the KB if the UNA is enforced this way.

Finally, because nominals are present in $\mathcal{ALCOV}$, KB satisfiability can be easily reduced to TBox satisfiability as the ABox part can be internalized as TBox axioms: $C(a)$ can be expressed by $\{a\} \sqsubseteq C$; $R(a,b)$ by $\{a\} \sqsubseteq \exists R.\{b\}$; $\neg R(a,b)$ by $\{a\} \sqsubseteq \forall R \neg \{b\}$; and $a \neq b$ by $\{a\} \sqsubseteq \neg\{b\}$.

Lemma 1. *KB satisfiability in $\mathcal{ALCOV}$ can be polynomially reduced to satisfiability of an $\mathcal{ALCOV}$ TBox.*

Thus, from now on, for KB satisfiability in $\mathcal{ALCOV}$, we assume that the input KB only consists of TBox.

3 Reasoning Algorithm for $\mathcal{ALCOV}$

3.1 Grounding Upfront

As noted in Section 2, the semantics of a KB is defined in terms of its ground form. Given an $\mathcal{ALCOV}$ KB $\mathcal{K}$ that contains some occurrences of nominal schemas, it is easy to see that its ground form is an exponentially larger KB $\mathcal{K}'$ that contains no nominal schema. Obviously, $\mathcal{K}'$ is actually an $\mathcal{ALCO}$ KB that carries the classical DL semantics. In fact, this is the most obvious approach for reasoning with any DL extended with nominal schemas. Let $\mathcal{L}$ be the extension of the DL $\mathcal{L}'$ with nominal schemas and $\mathcal{K}$ a KB in $\mathcal{L}$. To decide satisfiability of $\mathcal{K}$, we perform all possible grounding to all of the axioms in $\mathcal{K}$ yielding an exponentially larger $\mathcal{K}'$ which is expressible in $\mathcal{L}'$, then we perform any existing decision procedure of KB satisfiability for $\mathcal{L}'$ on $\mathcal{K}'$.

Such a combinatorial explosion is of course not desirable. We would prefer an approach in which grounding is performed *selectively* only when *needed*. In this section, we will describe a modification of standard tableau algorithms that incorporates *delayed* grounding.

3.2 A Tableau Algorithm with Delayed Grounding for $\mathcal{ALCOV}$

Standard tableau algorithms for DLs work on a graph structure whose nodes and edges are respectively labeled with sets of concepts and roles. Those algorithms are realized through a set of *tableau rules*, each of which is applied to a concept of certain form occurring in a node's label when certain conditions are satisfied. Tableau algorithms proceed by exhaustively applying the rules so long as any of them can be applied and no situation that indicate unsatisfiability — called *clash* — occurs. If they terminate without generating a clash, the resulting graph structure describes a model of the input KB or concept.

Our modification here follows the same setting, but also incorporate a grounding rule which can be applied to concepts (containing nominal schemas) occurring in a node label. The main idea is that if it is safe to apply a standard tableau rule without first grounding the applicable concept, then the algorithm should permit such a rule application to occur. The criteria as to which delaying grounding is safe is intuitively related to how nominal schemas occur within

the applicable concept. Furthermore, in case grounding is really needed prior to applying a tableau rule, we prefer to perform grounding *partially*, i.e., just on sufficient number of nominal schemas so that some safe application of tableau rules can proceed. At this point, the reader should keep in mind that the grounding rule that we will specify later can still be applied, although safe application of standard tableau rules is still possible. Thus, the grounding-upfront approach illustrated in Section 3.1 can be seen as a special case of the tableau algorithm with delayed grounding, although on the other hand, an efficient implementation would most likely defer grounding as much as possible.

The following two lemmas ensure the correctness of partial grounding. The proof of Lemma 2 is obvious from the semantics, while Lemma 3 follows from Lemma 2.

Lemma 2 (Partial Grounding). *Let C be a concept, $\mathcal{I}$ an interpretation, $s \in \Delta^{\mathcal{I}}$ an element of domain of interpretation, θ a total variable assignment, M_1, M_2 sets of variables with $M_1 \subseteq M_2 \in \mathsf{N}_V$. Then, $s \in (C\theta|_{M_1})^{\mathcal{I},\theta}$ iff $s \in (C\theta|_{M_2})^{\mathcal{I},\theta}$.*

Lemma 3. *Let C be a concept, Υ a set of variables, $\mathcal{I}$ an interpretation, and θ a total variable assignment. Then $C^{\mathcal{I},\theta} = (C\theta|_{\Upsilon})^{\mathcal{I},\theta} = (C\theta)^{\mathcal{I},\theta}$.*

For a concept C, $\mathsf{NNF}(C)$ is the concept this equivalent to C in negation normal form, i.e., in which negation signs only appear directly in front of concept names, nominals or nominal schemas. It is well-known that $\mathsf{NNF}(C)$ can be computed from C in linear time.

We now describe the main part of the tableau algorithm for deciding $\mathcal{ALCOV}$ KB satisfiability. Let $\mathcal{T}$ be a KB of which satisfiability is to be decided. By Lemma 1, $\mathcal{T}$ consists only of GCIs and no other types of axioms. Define Φ as the set of all total variable assignments. Also, the *closure* $\mathsf{clos}(C)$ of a concept C is a set of concepts containing C and is closed under sub-concepts. We then define the set $\mathsf{fclos}(\mathcal{T})$ as follows:

$$\mathsf{fclos}(\mathcal{T}) \coloneqq \bigcup_{\substack{C \sqsubseteq D \in \mathcal{T} \\ \theta \in \Phi, \Upsilon \subseteq \mathsf{N}_V}} \mathsf{clos}(\mathsf{NNF}(\neg C \sqcup D)\theta|_{\Upsilon})$$

Clearly, $\mathsf{fclos}(\mathcal{T})$ is finite because $\mathcal{T}$, Φ and N_V are finite. In addition, $\mathsf{NNF}(\neg C \sqcup D) \in \mathsf{fclos}(\mathcal{T})$ for every GCI $C \sqsubseteq D \in \mathcal{T}$ because Υ above can be empty. The algorithm starts with an *initial completion graph* for $\mathcal{T}$ as defined below.

Definition 5 (Completion Graph). *Let $\mathcal{T}$ be an $\mathcal{ALCOV}$ TBox and Φ is the set of total variable assignments from N_V to N_I. Assume that $|\mathsf{N}_I| = n$ and a_i's, $1 \leqslant i \leqslant n$, are all of its elements. A* completion graph *for $\mathcal{T}$ is a directed graph $\mathbf{G} = (V, E, \mathsf{L}, \mathsf{M})$ where V is a set of* nodes*, $E \subseteq V \times V$ is a set of* edges*, L is a labeling function that labels each node $r \in V$ with a set $\mathsf{L}(r) \subseteq \mathsf{fclos}(\mathcal{T})$ and each edge $\langle r, r' \rangle \in E$ with a set[2] $\mathsf{L}(r, r') \subseteq \mathsf{N}_R$, and M is a function that*

[2] the extra brackets are dropped for readability

maps each node to a set of non-empty variable assignment, i.e., a subset of $\{\theta|_\Upsilon \mid \theta \in \Phi, \Upsilon \subseteq \mathsf{N}_V, \Upsilon \neq \emptyset\}$.

To simplify the notation, henceforth, by $R \in \mathsf{L}(r, r')$*, we mean that both* $R \in \mathsf{L}(r, r')$ *and* $\langle r, r' \rangle \in E$.

An initial completion graph *for* $\mathcal{T}$ *with* $\mathsf{N}_I = \mathsf{Ind}(\mathcal{T}) = \{a_1, \ldots, a_n\}$ *is then the completion graph* $\mathbf{G}_0 = (V_0, \emptyset, \mathsf{L}_0, \emptyset)$ *for* $\mathcal{T}$ *where* $V_0 = \{r_1, \ldots, r_n\}$*, and* $\mathsf{L}_0(r_i) = \{\top, \{a_i\}\}$ *for* $1 \leqslant i \leqslant n$*. In addition, we specifically call the nodes* $r_1, \ldots, r_n$ *in* V_0 *as* initial nodes.

If $R \in \mathsf{L}(r, r')$*, then we say that* r' *is an* R-successor *of* r *and* r *is an* R-predecessor *of* r'*. In this case, we also say that* r' *is an* (R)-neighbor *of* r*. The transitive closure of the predecessor relation is called the* descendant *relation and the transitive closure of the successor relation is called the* ancestor *relation.*

A node $r \in V$ *is a* nominal node *if* $\mathsf{L}(r)$ *contains a nominal. Otherwise,* r *is a* blockable node*. Note that initial nodes are always nominal nodes.*

Definition 6 (Clash). *Let* $\mathbf{G} = (V, E, \mathsf{L}, \mathsf{M})$ *be a completion graph for a TBox* $\mathcal{T}$*. A node* $r \in V$ *contains a* clash *if at least one of the following holds:*

(i) $\bot \in \mathsf{L}(x)$*;*
(ii) $\{A, \neg A\} \subseteq \mathsf{L}(r)$ *for some* $A \in \mathsf{N}_C$*;*
(iii) $\{\{a\}, \neg\{a\}\} \subseteq \mathsf{L}(r)$ *for some* $a \in \mathsf{N}_I$*;*

We then say that $\mathbf{G}$ *contains a* clash *if some of its nodes contains a clash.*

Definition 7 (Blocking). *A node* r *is* blocked *if* r *is blockable and one of the following holds: (i) either* r *has a blocked ancestor; or (ii)* r *has a blockable ancestor* r' *such that* $\mathsf{L}(r) = \mathsf{L}(r')$ *and the path between* r' *and* r *consists only of blockable nodes; in this case we say that* r' (directly) blocks r.

The $\mathcal{ALCOV}$ tableau algorithm is thus as follows. Given a TBox $\mathcal{T}$, we exhaustively apply the tableau expansion rules from Figure 1 to the completion graph for $\mathcal{T}$, which is initialized to its initial completion graph, until none of the rules is applicable or the completion graph contains a clash. In the algorithm, $TBox$-rule, $\sqcap$-rule, $\forall$-rule, $\sqcup$-rule, $\exists^{\sqcup}$-rule, and $\exists$-rule are called *generating rules*, the O-rule is called the *shrinking rule*, while gr-rule, grv-rule, $gr\sqcup$-rule, $gr\exists^{\sqcup}$-rule, and $gr\exists$-rule are called *grounding rules*. Used in some of the tableau expansion rules, the set $\mathsf{Tvar}(C)$ is defined as: $\mathsf{Tvar}(\{v\}) = \{\{v\}\}$ for any $v \in \mathsf{N}_V$; $\mathsf{Tvar}(\neg C) = \mathsf{Tvar}(C)$; $\mathsf{Tvar}(C \sqcap D) = \mathsf{Tvar}(C \sqcup D) = \mathsf{Tvar}(C) \cup \mathsf{Tvar}(D)$; $\mathsf{Tvar}(\forall R.C) = \mathsf{Tvar}(\exists R.C) = \emptyset$; and $\mathsf{Tvar}(C) = \emptyset$ if $\mathsf{Var}(C) = \emptyset$. The completion graph is *complete* when none of the rules is applicable to it or it contains a clash. If the expansion rules can be applied such that a complete, clash-free completion graph for $\mathcal{T}$ is generated, then the algorithm returns "$\mathcal{T}$ is satisfiable". Otherwise, it returns "$\mathcal{T}$ is unsatisfiable".

In general, this tableau algorithm is similar to the tableau algorithm for the DLs $\mathcal{ALCO}$ [11] (see also the tableau algorithms for DLs that cover $\mathcal{ALCO}$: $\mathcal{SHOQ}$ [12], and $\mathcal{SHIO}$ [13]). In fact, the action performed by the O-rule is a merging of nodes and pruning parts of the completion graph due to nominals which is also described in the more expressive DLs DLs $\mathcal{SHOIQ}$ [14] and $\mathcal{SROIQ}$ [3].

$TBox$-rule: if: for some $C \sqsubseteq D \in \mathcal{T}$, $\mathsf{NNF}(\neg C \sqcup D) \notin \mathsf{L}(r)$ and r is not blocked,
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{\mathsf{NNF}(\neg C \sqcup D)\}$

O-rule: if: $\{a\} \in \mathsf{L}(r)$ for some $a \in \mathsf{N}_I$, r is not blocked
then: for some initial node s with $\{a\} \in \mathsf{L}(s)$, do the following four steps:
first, for each edge $\langle r', r\rangle$ do: if $\langle r', s\rangle \in E$, set $\mathsf{L}(r', s) \coloneqq \mathsf{L}(r', s) \cup \mathsf{L}(r, s)$; else set $E \coloneqq E \cup \{\langle r', s\rangle\}$ and $\mathsf{L}(r', s) \coloneqq \mathsf{L}(r, s)$
second, for each edge $\langle r, r'\rangle$ where r' is not blockable, do : if $\langle s, r'\rangle \in E$, set $\mathsf{L}(s, r') \coloneqq \mathsf{L}(s, r') \cup \mathsf{L}(r, r')$; else set $E \coloneqq \{\langle s, r'\rangle\}$ and $\mathsf{L}(s, r') \coloneqq \mathsf{L}(r, r')$;
third, set $\mathsf{L}(s) \coloneqq \mathsf{L}(s) \cup \mathsf{L}(r)$; and
fourth, perform $\textsc{Prune}(r)$ which is the following recursive procedure: (1) for every successor r' of r, remove $\langle r, r'\rangle$ from E; and, if r' is blockable, then $\textsc{Prune}(r')$; (2) remove r from V.

$\sqcap$-rule: if: $C \sqcap D \in \mathsf{L}(r)$, r is not blocked, $\{C, D\} \not\subseteq \mathsf{L}(r)$
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{C, D\}$

$\sqcup$-rule: if: $C \sqcup D \in \mathsf{L}(r)$, $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, r is not blocked, and $\{C, D\} \cap \mathsf{L}(r) = \emptyset$
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{C'\}$ for some $C' \in \{C, D\}$

$\forall$-rule: if: $\forall R.C \in \mathsf{L}(r)$, r is not blocked, $R \in \mathsf{L}\langle r, s\rangle$ and $C \notin \mathsf{L}(s)$ for some node s
then: $\mathsf{L}(s) \coloneqq \mathsf{L}(s) \cup \{C\}$

$\exists^{\sqcup}$-rule: if: $\exists R.(C \sqcup D) \in \mathsf{L}(r)$, $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, r is not blocked,
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{\exists R.C'\}$ for some $C' \in \{C, D\}$

$\exists$-rule: if: $\exists R.C \in \mathsf{L}(r)$, C is not a disjunction $C' \sqcup D'$, $\mathsf{Tvar}(C) = \emptyset$, r is not blocked, r has no R-successor s with $C \in \mathsf{L}(s)$
then: create a node s with $\mathsf{L}(r, s) \coloneqq \{R\}$ and $\mathsf{L}(s) \coloneqq \{C\}$

grv-rule: if: (i) for some $v \in \mathsf{N}_V$ and some $a \in \mathsf{N}_I$, either $\{v\} \in \mathsf{L}(r)$ and $\{a\} \notin \mathsf{L}(r)$, or $\neg\{v\} \in \mathsf{L}(r)$ and $\neg\{a\} \notin \mathsf{L}(r)$; (ii) $v/a \notin \mathsf{M}(r)$; and (iii) r is not blocked
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{D[v/a] \mid D \in \mathsf{L}(r)\}$ and $\mathsf{M}(r) \coloneqq \mathsf{M}(r) \cup \{v/a\}$

$gr\sqcup$-rule: if: (i) $C \sqcup D \in \mathsf{L}(r)$; (ii) $(C \sqcup D)[v/a] \notin \mathsf{L}(r)$ for some $v \in \mathsf{Var}(C) \cap \mathsf{Var}(D)$ and $a \in \mathsf{N}_I$; (iii) $v/a \notin \mathsf{M}(r)$; and (iv) r is not blocked
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{D[v/a] \mid D \in \mathsf{L}(r)\}$ and $\mathsf{M}(r) \coloneqq \mathsf{M}(r) \cup \{v/a\}$

$gr\exists^{\sqcup}$-rule: if: (i) $\exists R.(C \sqcup D) \in \mathsf{L}(r)$; (ii) $\exists R.(C \sqcup D)[v/a] \notin \mathsf{L}(r)$ for some $v \in \mathsf{Var}(C) \cap \mathsf{Var}(D)$ and $a \in \mathsf{N}_I$; (iii) $v/a \notin \mathsf{M}(r)$; and (iv) r is not blocked
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{D[v/a] \mid D \in \mathsf{L}(r)\}$ and $\mathsf{M}(r) \coloneqq \mathsf{M}(r) \cup \{v/a\}$

$gr\exists$-rule: if: (i) $\exists R.C \in \mathsf{L}(r)$ and C is not a disjunction; (ii) $\exists R.C[v/a] \notin \mathsf{L}(r)$ for some $v \in \mathsf{Tvar}(C)$ and $a \in \mathsf{N}_I$; (iii) $v/a \notin \mathsf{M}(r)$; and (iv) r is not blocked
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{D[v/a] \mid D \in \mathsf{L}(r)\}$ and $\mathsf{M}(r) \coloneqq \mathsf{M}(r) \cup \{v/a\}$

gr-rule: if: (i) $C \in \mathsf{L}(r)$; (ii) $C[v/a] \notin \mathsf{L}(r)$, but $v/a \in \mathsf{M}(r)$ for some $v \in \mathsf{Var}(C)$ and some $a \in \mathsf{N}_I$; and (iii) r is not blocked
then: $\mathsf{L}(r) \coloneqq \mathsf{L}(r) \cup \{D[v/a] \mid D \in \mathsf{L}(r)\}$

Fig. 1. Tableau expansion rules for $\mathcal{ALCOV}$.

The new contribution of this paper is the modification of those tableau algorithms to incorporate delayed grounding to deal with nominal schemas. This is realized through grounding rules and some additional precondition for the standard tableau expansion rules. All grounding rules non-deterministically choose a variable assignment that can be used to partially ground the corresponding concept. This provides some flexibility for incorporating some heuristics that

allow an implementation to ground variables in such a way that a clash can be found as early as possible.

The intuition behind the grounding rules comes from the way concept expressions with nominal schemas are read as FOL formulas with equality. For example, the concept $\exists R.(\{x\} \sqcap \exists S.\{y\})$ can be read as the formula

$$\forall x.\forall y.\exists z'.(R(s,z') \wedge x = z' \wedge \exists z''.(S(z',z'') \wedge y = z'')) \tag{1}$$

where s is the meta-variable that represents a domain element for which the formula holds. The universal quantifiers that correspond to (FO-)variables x, y and z associated with nominal schemas are interpreted in a special way: they range over the set of named individuals, i.e., not the whole domain. Thus, grounding the nominal schemas to every possible individual names is equivalent to dropping those universal quantifiers.

Now, suppose that during the execution of the algorithm, a node s of the completion graph has the above concept in its label and the knowledge base contains a and b as the named individuals (hence, there are two initial nodes s_a whose label contains $\{a\}$ and s_b whose label contains $\{b\}$). The $gr\exists$-rule will ground the concept to generate new concepts: $\exists R.(\{a\} \sqcap \exists S.\{y\})$ and $\exists R.(\{b\} \sqcap \exists S.\{y\})$ which can be read as the formulas

$$\forall y.\exists z'.(R(s,z') \wedge a = z' \wedge \exists z''.(S(z',z'') \wedge y = z'')) \tag{2}$$

$$\forall y.\exists z'.(R(s,z') \wedge b = z' \wedge \exists z''.(S(z',z'') \wedge y = z'')) \tag{3}$$

The algorithm will process the above two concepts in the labeling of s by generating two individuals, each of which are then merged by the O-rule to s_a and s_b. This makes both s_a's and s_b's labels contains $\exists S.\{y\}$. This situation is the same as having the following two formulas:

$$R(s,a) \wedge \forall y.\exists z''.(S(a,z'') \wedge y = z'') \tag{4}$$

$$R(s,b) \wedge \forall y.\exists z''.(S(b,z'') \wedge y = z'') \tag{5}$$

Clearly, the conjunction of (4) and (5) is equivalent to (1) if we interpret the (first-ordervariable x to range only over named individuals. More importantly, notice that the universal quantifier that binds y now moves "inside". This is done correctly because the existential quantifier that binds z' has been instantiated beforehand and all possible instantiations of z' must be equal to some named individuals due to x which generate, in this case, two different formulas. Note that this cannot be done for the universal quantifier that binds x on (1) because even if z' is instantiated beforehand, there is only one formula which does not capture all possible instantiations of z'.

The above example also illustrates that grounding y can be done after $\exists$-rule is applied. This epitomises the delayed grounding strategy. A full grounding upfront would have fully grounded y (in addition to x) in the labeling of s. This would result in four new labels, instead of two as in the above example. If this is followed by applications of $\exists$-rule, there will be four new individuals

$$
\begin{aligned}
\exists \text{hasReviewAssignment}.((\{x\} \sqcap \exists \text{hasAuthor}.\{y\}) \sqcap (\{x\} \sqcap \exists \text{atVenue}.\{z\})) \\
\sqcap \exists \text{hasSubmittedPaper}.(\exists \text{hasAuthor}.\{y\} \sqcap \exists \text{atVenue}.\{z\}) \\
\sqsubseteq \exists \text{hasConflictingAssignedPaper}.\{x\} \\
\{p_0\} \sqsubseteq \exists \text{hasAuthor}.\{a_{1000}\} \sqcap \exists \text{hasAuthor}.\{a_1\} \\
\{p_i\} \sqsubseteq \exists \text{hasAuthor}.\{a_i\} \sqcap \exists \text{hasAuthor}.\{a_{i+1}\} \\
\{a_i\} \sqsubseteq \exists \text{hasSubmittedPaper}.\{p_{i-1}\} \sqcap \exists \text{hasSubmittedPaper}.\{p_i\} \\
\{a_{1000}\} \sqsubseteq \exists \text{hasSubmittedPaper}.\{p_{999}\} \sqcap \exists \text{hasSubmittedPaper}.\{p_0\} \\
\{p_j\} \sqsubseteq \exists \text{AtVenue}.\{\text{ISWC}\} \\
\{a_k\} \sqsubseteq \exists \text{hasReviewAssignment}.\{p_{k-4}\} \sqcap \exists \text{hasReviewAssignment}.\{p_{k-3}\} \\
\{a_1\} \sqsubseteq \exists \text{hasReviewAssignment}.\{p_{999}\} \sqcap \exists \text{hasReviewAssignment}.\{p_{998}\}
\end{aligned}
$$

Fig. 2. Example for delayed grounding. $i = 1, \ldots, 999$, $j = 0, \ldots, 999$, $k = 4, \ldots, 1000$.

generated, although they will eventually be merged into two due to O-rule. This unnecessary creation of new individuals is avoided if grounding y can be delayed.

In general, the precondition concerning variables specified in $\sqcup$-rule, $\exists$-rule and $\exists^{\sqcup}$-rule pinpoints which variables must be grounded before the concept can be "split". This is closely related to those situations in which universal quantifiers can be moved inside a FOL formula as exemplified above. For $\sqcup$-rule, any variable occurring in both disjuncts must be grounded before the disjunction can be split. Intuitively, this is due to the fact that universal quantifiers cannot distribute over disjunction, i.e., $\forall x.(A(x) \vee B(x))$ is not equivalent to $\forall x.A(x) \vee \forall x.B(x)$. This similarly holds for $\exists^{\sqcup}$-rule which also pushes the disjunction outside. For $\exists$-rule that is applied to a concept $\exists R.C$ with C not a disjunction, any variable occurring in the set $\mathsf{Tvar}(C)$ must be grounded before the tableau algorithm can process the concept by creating a new individual. Meanwhile, $\sqcap$-rule and $\forall$-rule can be safely applied without prior grounding of nominal schemas. This should be clear as occurrences of nominal schemas induce universal quantifiers and those quantifiers can be freely moved inside conjunction or other universal quantifiers.

The above explanation should make the idea of delayed grounding clear to the reader. To describe how the algorithm would potentially perform in comparison to grounding-upfront approach described in 3.1, we use the knowledge base KB in Figure 2. Given KB, we ask whether it entails the existence of a conflicting review assignment, i.e., whether the concept $D \coloneqq \forall \text{hasConflictingAssignedPaper}.\bot$ is unsatisfiable w.r.t. KB.

The answer must be yes since a_1 and a_{1000} co-author p_0, but a_1 has review assignment on p_{999} whose authors include a_{1000}. To run the algorithm, we first add a GCI $\{r_0\} \sqsubseteq D$ to KB where r_0 is a new individual. This knowledge base has 2002 individual names and three nominal schema $\{x\}$, $\{y\}$ and $\{z\}$. Note that if we solve this entailment by grounding the first axiom up front, we would have more than 8×10^9 new axioms.

On the other hand, our algorithm can discover the clash without necessarily grounding everything up front. Initially, an initial completion graph is constructed with nodes, say, r_0, $a_0, \ldots, a_{999}$, and $p_0, \ldots, p_{999}$, and $iswc$. We can quickly add GCIs into r_0, p_{999}, and a_{1000}. Because r_0 contains only one nominal $\{r_0\}$ in its label, r_0 will immediately receive the RHSs of all GCIs in KB, except the first GCI. Next, after a few applications of the $\sqcap$-rule, we apply the $\exists$-rule and the o-rule to r_0, a_{1000} and p_{999}. We obtain, without applying the grounding rule:

- p_0 is a successor of both r_0 and a_{1000} through the hasSubmittedPaper role.
- p_{999} is a successor of r_0 through the hasReviewAssignment role.
- p_{999} is a successor of a_{1000} through the hasSubmittedPaper role.
- p_0 and p_{999} has $iswc$ as their successor through the atVenue role.
- a_{1000} is a successor of p_{999} through the hasAuthor role.

Next, we apply the $\sqcup$-rule on r_0 to choose one of the disjuncts from $\mathsf{NNF}(\neg C_1 \sqcup D_1)$ where $C_1 \sqsubseteq D_1$ is the first axiom in KB (the only one containing occurrences of nominal schemas). Suppose the first disjunct is chosen on r_0. Next, apply $\forall$-rules and the GR3-rule to eventually obtain a clash when $\neg\{x\}$ is grounded to $\neg\{p_{999}\}$ on p_{999}. If we choose to propagate $\neg\{y\}$ through the hasAuthor role, again we get a clash when grounding $\neg\{y\}$ to $\neg\{a_{1000}\}$ on a_{1000}. Similarly, a clash also occurs when we propagate $\neg\{z\}$ through the atVenue role.

Next, suppose the second disjunct is chosen on r_0. Propagating through the hasSubmittedRole means that p_0 must contain the label $\forall$hasAuthor.$\neg\{y\} \sqcup \forall$atVenue.$\neg\{z\}$. Similar to p_{999}, any choice of disjunct here will lead to a clash. So we can only now select the third disjunct on r_0: $\exists$hasConflictingAssignedPaper.$\{x\}$. Here, grounding x to p_{999} and applying the $\exists$-rule followed by the O-rule will connect r_0 to p_{999} through the hasConflictingAssignedPaper role. But then, r_0 also contains $\forall$hasConflictingAssignedPaper.$\bot$ in its label. Hence, applying the $\forall$-rules will propagate $\bot$ to p_{999} leading to a clash.

So, by only grounding x to p_{999}, y to a_{1000} and z to ISWC, we can discover the clashes quickly and derive the unsatisfiability of $\forall$hasConflictingAssignedPaper.$\bot$. By delaying grounding and then performing it selectively, we can avoid the combinatorial explosion which is unavoidable when grounding is done upfront. Obviously, this does not mean that combinatorial explosion can always be avoided. On the other hand, this idea can lead to more practical heuristics that enable efficient implementations for reasoning with nominal schemas.

4 Correctness of $\mathcal{ALCOV}$ Tableau Algorithm

We first show that the tableau algorithm for $\mathcal{ALCOV}$ as described in Section 3.2 terminates. We then proceed with soundness and completeness afterward. Due to space restriction, we refer the reader to [15] for full proofs.

Lemma 4 (Termination). *Given a $\mathcal{ALCOV}$ TBox $\mathcal{T}$ as input, the tableau algorithm for $\mathcal{ALCOV}$ terminates.*

Proof. Each application of the grounding rules add a new partially grounded concepts to o con- cept labeling of a node in the completion graph. Thus, the number of those rule applications is bounded by the number of partially grounded SROIQV concept that can be generated from $\mathsf{fclos}(\mathcal{T})$ which is finite due to the fact the num- ber of all possible variable assignments is finite. The termination would then follow from this, together with the termination of $\mathcal{ALCO}$ tableau algorithm which is implied by termination of $\mathcal{SHOQ}$ tableau algorithm [12].

In order to show the soundness and completeness of the algorithm, we employ the standard unraveling technique that is commonly used in many correctness proofs of tableau algorithms in DLs. This technique is realized through the so-called tableau structure that corresponds to a model of a TBox.

Definition 8 (Tableau for $\mathcal{ALCOV}$). *Let $\mathcal{T}$ be an $\mathcal{ALCOV}$ TBox and Φ the set of all total variable assignments. A* tableau for $\mathcal{T}$ *is a triple $T = (S, \mathcal{L}, \mathcal{E})$ such that S is a non-empty set, $\mathcal{L}\colon S \to 2^{\mathsf{fclos}(\mathcal{T})}$ maps every element of S to a subset of $\mathsf{fclos}(\mathcal{T})$, and $\mathcal{E}\colon \mathsf{N}_R \to 2^{S\times S}$ maps each role to a set of pairs of elements from S. Furthermore, T satisfies the following conditions:*

- ***(T0)*** *for each $s \in S$, $\mathsf{NNF}(\neg C \sqcup D) \in \mathcal{L}(s)$ for every $C \sqsubseteq D \in \mathcal{T}$;*
- ***(T1)*** *for each $a \in \mathsf{N}_I$, $\{a\} \in \mathcal{L}(s)$ for some $s \in S$;*
- ***(T2)*** *for each $a \in \mathsf{N}_I$ and $s, t \in S$, $\{a\} \in \mathcal{L}(s) \cap \mathcal{L}(t)$ implies $s = t$;*
- ***(T3)*** *for each $v \in \mathsf{N}_V$ and $s \in S$, $\{v\} \in \mathcal{L}(s)$ implies $\{a\} \in \mathcal{L}(s)$ for every $a \in \mathsf{N}_I$;*
- ***(T4)*** *for each $v \in \mathsf{N}_V$ and $s \in S$, $\neg\{v\} \in \mathcal{L}(s)$ implies $\{a\} \notin \mathcal{L}(s)$ for every $a \in \mathsf{N}_I$;*
- ***(T5)*** *for each $s \in S$, $\top \in \mathcal{L}(s)$ and $\bot \notin \mathcal{L}(s)$;*
- ***(T6)*** *for each $s \in S$, $A \in \mathcal{L}(s)$ implies $\neg A \notin \mathcal{L}(s)$ where A atomic or nominal;*
- ***(T7)*** *for each $s \in S$, if $C \sqcap D \in \mathcal{L}(s)$, then $\{C, D\} \subseteq \mathcal{L}(s)$.*
- ***(T8)*** *for each $s \in S$, if $\forall R.C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(R)$ for some $t \in S$, then $C \in \mathcal{L}(t)$.*
- ***(T9)*** *for each $s \in S$, if $C \sqcup D \in \mathcal{L}(s)$, then*
  - *(T9a) if $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, then $\{C, D\} \cap \mathcal{L}(s) \neq \emptyset$;*
  - *(T9b) else, for every $\theta \in \Phi$, $\Upsilon \subseteq \mathsf{Var}(C) \cap \mathsf{Var}(D)$, $(C \sqcup D)\theta|_\Upsilon \in \mathcal{L}(s)$.*
- ***(T10)*** *for each $s \in S$, if $\exists R.C \in \mathcal{L}(s)$, then*
  - *(T10a) if C is of the form $D \sqcup E$ and $\mathsf{Var}(D) \cap \mathsf{Var}(E) = \emptyset$, then $\{\exists R.D, \exists R.E\} \cap \mathcal{L}(s) \neq \emptyset$;*
  - *(T10b) if C is of the form $D \sqcup E$ and $\mathsf{Var}(D) \cap \mathsf{Var}(E) \neq \emptyset$, then for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Var}(D) \cap \mathsf{Var}(E)$, $(\exists R.C)\theta|_\Upsilon \in \mathcal{L}(s)$;*
  - *(T10c) if C is **not** of the form $D \sqcup E$ and $\mathsf{Tvar}(C) = \emptyset$, then there is some $t \in S$ such that $\langle s, t\rangle \in \mathcal{E}(R)$ and $C \in \mathcal{L}(t)$;*
  - *(T10d) if C is not of the form $D \sqcup E$ and $\mathsf{Tvar}(C) \neq \emptyset$, then for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Tvar}(C)$, $(\exists R.C)\theta|_\Upsilon \in \mathcal{L}(s)$.*
- ***(T11)*** *for each $s \in S$, if $C \in \mathcal{L}(s)$, then:*
  - *(i) if there exists $\{v\} \in \mathcal{L}(s)$ (resp. $\neg\{v\} \in \mathcal{L}(s)$) for some $v \in \mathsf{N}_V$, and for every $a \in \mathsf{N}_I$, $\{a\} \in \mathcal{L}(s)$ (resp. $\neg\{a\} \in \mathcal{L}(s)$), then for each such a, $C[v/a] \in \mathcal{L}(s)$*

(ii) *if there exists* $D = D_1 \sqcup D_2 \in \mathcal{L}(s)$ *and for every* $\theta \in \Phi$ *and* $\Upsilon \subseteq \mathsf{Var}(D_1) \cap \mathsf{Var}(D_2)$, $D\theta|_\Upsilon \in \mathcal{L}(s)$, *then for each such* θ *and* Υ, $C\theta|_\Upsilon \in \mathcal{L}(s)$;

(iii) *if there exists* $D = \exists R.(D_1 \sqcup D_2) \in \mathcal{L}(s)$, *and for every* $\theta \in \Phi$ *and* $\Upsilon \subseteq \mathsf{Var}(D_1) \cap \mathsf{Var}(D_2)$, $D\theta|_\Upsilon \in \mathcal{L}(s)$, *then for each such* θ *and* Υ, $C\theta|_\Upsilon \in \mathcal{L}(s)$;

(iv) *if there exists* $D = \exists R.D_1$, D_1 *is not a disjunction, and for every* $\theta \in \Phi$ *and* $\Upsilon \subseteq \mathsf{Tvar}(D_1)$, $D\theta|_\Upsilon \in \mathcal{L}(s)$, *then for each such* θ *and* Υ, $C\theta|_\Upsilon \in \mathcal{L}(s)$.

Lemma 5. *An* $\mathcal{ALCOV}$ *TBox* $\mathcal{T}$ *is satisfiable iff there is a tableau for* $\mathcal{T}$.

Proof (of Lemma 5).

"If direction": Let $\mathcal{T}$ be an $\mathcal{ALCOV}$ TBox and $T = (S, \mathcal{L}, \mathcal{E})$ be a tableau for $\mathcal{T}$. Define an interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ such that

- $\Delta^\mathcal{I} \coloneqq S$ which is not empty;
- for every $A \in \mathsf{N}_C$, $A^\mathcal{I} \coloneqq \{s \in \Delta^\mathcal{I} \mid A \in \mathcal{L}(s)\}$;
- for every $a \in \mathsf{N}_I$, $a^\mathcal{I} \coloneqq s$ for some $s \in \Delta^\mathcal{I}$ with $\{a\} \in \mathcal{L}(s)$; and
- for every $R \in \mathsf{N}_R$, $R^\mathcal{I} \coloneqq \mathcal{E}(R)$.

We prove that $\mathcal{I}$ is a model of $\mathcal{T}$ and conforms the semantics from Definition 4 by showing the following claim: "For every concept $C \in \mathsf{fclos}(\mathcal{T})$ and every $s \in \Delta^\mathcal{I}$, if $C \in \mathcal{L}(s)$, then $s \in C^{\mathcal{I},\theta}$ for every total variable assignment θ."

The claim is proved by induction on the length of C (which is in NNF). The base cases where C is a concept name, top concept or bottom concept is immediate from the definition of $\mathcal{I}$ and (T5). If C is a nominal, then the claim is established by definition of $\mathcal{I}$. Note also that $\mathcal{I}$ is well-defined due to (T1) and (T2). If C is a nominal schema, then the claim holds because of (T3). Now, let Φ be the set of every total variable assignment. We consider the other induction cases as follows for every $s \in \Delta^\mathcal{I} = S$:

- $C = \neg\{v\}$ for some $v \in \mathsf{N}_V$: If $\neg\{v\} \in \mathcal{L}(s)$, then by (T4), there is no $a \in \mathsf{N}_I$ such that $\{a\} \in \mathcal{L}(s)$. By definition of $\mathcal{I}$, $s \neq a^\mathcal{I}$ for every $a \in \mathsf{N}_I$. Thus, for every $\theta \in \Phi$, $s \neq (v\theta)^{\mathcal{I},\theta}$ which implies that $s \notin (\{v\}\theta)^{\mathcal{I},\theta} = (\{v\})^{\mathcal{I},\theta}$, i.e., $s \in (\neg\{v\})^{\mathcal{I},\theta}$.
- $C = \neg D$ where D is an atomic concept or a nominal: Suppose that $\neg D \in \mathcal{L}(s)$. (T6) implies that $D \notin \mathcal{L}(s)$. By definition of $\mathcal{I}$, $s \notin D^\mathcal{I} = D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Thus, $s \in (\neg D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
- $C = D \sqcap E$: Suppose $D \sqcap E \in \mathcal{L}(s)$. By (T7), $\{D, E\} \subseteq \mathcal{L}(s)$. By induction, $s \in D^{\mathcal{I},\theta}$ and $s \in E^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. By the semantics, $s \in (D \sqcap E)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
- $C = \forall R.D$: Suppose $\forall R.D \in \mathcal{L}(s)$ and $\langle s, t\rangle \in R^\mathcal{I} = \mathcal{E}(R)$ for some $t \in S$. By (T8), $D \in \mathcal{L}(t)$. Thus, by induction, $t \in D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Hence, by the semantics, $s \in (\forall R.D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
- $C = D \sqcup E$: Let $D \sqcup E \in \mathcal{L}(s)$. First case: $\mathsf{Var}(D) \cap \mathsf{Var}(E) = \emptyset$. By (T9a), $\{D, E\} \cap \mathcal{L}(s) \neq \emptyset$. By induction, $s \in D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ or $s \in E^{\mathcal{I},\theta}$ for

every $\theta \in \Phi$. Since $\mathsf{Var}(D) \cap \mathsf{Var}(E) = \emptyset$, we have that $s \in D^{\mathcal{I},\theta}$ or $s \in E^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Hence, $s \in (D \sqcup E)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
Second case: $\mathsf{Var}(D) \cap \mathsf{Var}(E) \neq \emptyset$. By (T9b), $(D \sqcup E)\theta|_\Upsilon \in \mathcal{L}(s)$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Var}(D) \cap \mathsf{Var}(E)$. In particular, when $\Upsilon = \mathsf{Var}(D) \cap \mathsf{Var}(E)$, $(D \sqcup E)\theta|_\Upsilon = D\theta|_\Upsilon \sqcup E\theta|_\Upsilon$ and $\mathsf{Var}(D\theta|_\Upsilon) \cap \mathsf{Var}(E\theta|_\Upsilon) = \emptyset$. Thus, we can apply the first case above to obtain $s \in ((D \sqcup E)\theta|_\Upsilon)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. By Lemma 3, we thus have that $s \in (D \sqcup E)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$

– $C = \exists R.D$: Let $\exists R.D \in \mathcal{L}(s)$. First case: $D = E_1 \sqcup E_2$ for some concepts E_1, E_2 and $\mathsf{Var}(E_1) \cap \mathsf{Var}(E_2) = \emptyset$. By (T10a), $\{\exists R.E_1, \exists R.E_2\} \cap \mathcal{L}(s) \neq \emptyset$. By induction, $s \in (\exists R.E_1)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ or $s \in (\exists R.E_2)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Since $\mathsf{Var}(E_1) \cap \mathsf{Var}(E_2) = \emptyset$, we have that $s \in (\exists R.E_1)^{\mathcal{I},\theta}$ or $s \in (\exists R.E_2)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Hence, for every $\theta \in \Phi$, $s \in (\exists R.E_1)^{\mathcal{I},\theta} \cup (\exists R.E_2)^{\mathcal{I},\theta}$, i.e,. $s \in (\exists R.E_1 \sqcup \exists R.E_2)^{\mathcal{I},\theta}$. It follows that $s \in (\exists R.(E_1 \sqcup E_2))^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
Second case: $D = E_1 \sqcup E_2$ for some concepts E_1, E_2 and $\mathsf{Var}(E_1) \cap \mathsf{Var}(E_2) \neq \emptyset$. By (T10b), $(\exists R.D)\theta|_\Upsilon \in \mathcal{L}(s)$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Var}(D) \cap \mathsf{Var}(E)$. In particular when $\Upsilon = \mathsf{Var}(D) \cap \mathsf{Var}(E)$, $(\exists R.D)\theta|_\Upsilon = (\exists R.(E_1 \sqcup E_2))\theta|_\Upsilon = \exists R.(E_1\theta|_\Upsilon \sqcup E_2\theta|_\Upsilon)$ and $\mathsf{Var}(E_1\theta|_\Upsilon) \cap \mathsf{Var}(E_2\theta|_\Upsilon) = \emptyset$. Hence, the first case holds, i.e., $s \in (\exists R.(E_1\theta|_\Upsilon \sqcup E_2\theta|_\Upsilon))^{\mathcal{I},\theta} = ((\exists R.(E_1 \sqcup E_2))\theta|_\Upsilon)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Applying Lemma 3, we thus obtain that $s \in (\exists R.(E_1 \sqcup E_2))^{\mathcal{I},\theta}$ for every $\theta \in \Phi$
Third case: D is not a disjunction and $\mathsf{Tvar}(D) = \emptyset$. By (T10c), there is some $t \in S$ such that $\langle s, t\rangle \in \mathcal{E}(R) = R^{\mathcal{I}}$ and $D \in \mathcal{L}(t)$. By induction, $t \in D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. So, we obtain that $s \in (\exists R.D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.
Fourth case: D is not a disjunction and $\mathsf{Tvar}(D) \neq \emptyset$. By (T10d), $(\exists R.D)\theta|_\Upsilon \in \mathcal{L}(s)$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Tvar}(D)$. In particular, when $\Upsilon = \mathsf{Tvar}(D)$, $(\exists R.D)\theta|_\Upsilon = \exists R.D\theta|_\Upsilon$ and $\mathsf{Tvar}(D\theta|_\Upsilon) = \emptyset$. Hence, the third case applies, i.e., $s \in ((\exists R.D)\theta|_\Upsilon)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Finally, using Lemma 3, we conclude that $s \in (\exists R.D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.

In addition, (T11) does not invalidate the claim because of Lemma 3. Thus, the claim is established and we conclude that $\mathcal{I}$ is a model of $\mathcal{T}$ from the claim and the fact that T satisfies (T0).

"Only-if direction": Let $\mathcal{I}$ be a model of $\mathcal{T}$. Let Φ be the set of all total variable assignment from N_V to N_I. We construct a tableau $T = (S, \mathcal{L}, \mathcal{E})$ for $\mathcal{T}$ as follows:

– $S \coloneqq \Delta^{\mathcal{I}}$;
– $\mathcal{E}(R) \coloneqq R^{\mathcal{I}}$;
– for each $s \in S$, $\mathcal{L}(s) \coloneqq \{C \in \mathsf{fclos}(\mathcal{T}) \mid s \in C^{\mathcal{I},\theta} \text{ for all } \theta \in \Phi\}$

We show that T is indeed a tableau for $\mathcal{T}$ satisfying (T0)–(T10).

(T0): Immediate, due to the definition of T and the fact that $\mathcal{I} \models \mathcal{T}$.
(T1): Immediate, due to the definition of T and the semantics.
(T2): Immediate, due to the definition of T and the semantics.
(T3): Immediate, due to the definition of T and the semantics.
(T4): Immediate, due to the definition of T and the semantics.

(T5): Immediate, due to the definition of T and the semantics.

(T6): Immediate, due to the definition of T and the fact that $\mathcal{I} \models \mathcal{T}$.

(T7): Suppose $C \sqcap D \in \mathcal{L}(s)$. The definition of T implies that $s \in (C \sqcap D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. By the semantics, we have that $s \in C^{\mathcal{I},\theta}$ and $s \in D^{\mathcal{I},\theta}$. Hence, by construction of $\mathcal{L}(s)$, $\{C, D\} \subseteq \mathcal{L}(s)$.

(T8): Let $\forall R.C \in \mathcal{L}(s)$ and $\langle s, t\rangle \in \mathcal{E}(R)$ for some $t \in S$. By definition of T, $s \in (\forall R.C)^{\mathcal{I},\theta}$ and $\langle s, t\rangle \in R^{\mathcal{I}}$ for every $\theta \in \Phi$. By the semantics, clearly $t \in C^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Hence, by construction of $\mathcal{L}(t)$, $C \in \mathcal{L}(t)$.

(T9): Suppose $C \sqcup D \in \mathcal{L}(s)$. By definition of T, $s \in (C \sqcup D)^{\mathcal{I},\theta} = C^{\mathcal{I},\theta} \cup D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Then, for every $\theta \in \Phi$, either $s \in C^{\mathcal{I},\theta}$ or $s \in D^{\mathcal{I},\theta}$. To establish (T9a), assume that $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$. Then, either $s \in C^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ or $s \in D^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Note that this only holds because $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$. Hence, definition of T implies that either $C \in \mathcal{L}(s)$ or $D \in \mathcal{L}(s)$, i.e., $\{C, D\} \cap \mathcal{L}(s) \neq \emptyset$, establishing (T9a). Next, for (T9b), assume that $\mathsf{Var}(C) \cap \mathsf{Var}(D) \neq \emptyset$. Then, by Lemma 3, $s \in ((C \sqcup D)\theta_{\Upsilon})^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Var}(C) \cap \mathsf{Var}(D)$. Clearly, by definition of T, $(C \sqcup D)\theta_{\Upsilon} \in \mathcal{L}(s)$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{Var}(C) \cap \mathsf{Var}(D)$

(T10): Let $\exists R.C \in \mathcal{L}(s)$. By definition of T, $s \in (\exists R.C)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$.

First, assume that $C = D \sqcup E$ and $\mathsf{Var}(D) \cap \mathsf{Var}(E) = \emptyset$. Then, we can distribute existential restriction over disjunction, giving us $s \in (\exists R.D \sqcup \exists R.E)^{\mathcal{I},\theta} = (\exists R.D)^{\mathcal{I},\theta} \cup (\exists R.E)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Moreover, since $\mathsf{Var}(D) \cap \mathsf{Var}(E) = \emptyset$, we have that either $s \in (\exists R.D)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ or $s \in (\exists R.E)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Consequently, by definition of T, $\{\exists R.D, \exists R.E\} \cap \mathcal{L}(s) \neq \emptyset$, establishing (T10a).

Secondly, assume that $C = D \sqcup E$, but $\mathsf{Var}(D) \cap \mathsf{Var}(E) \neq \emptyset$. Then, by definition of T and Lemma 3, $s \in (\exists R.C)^{\mathcal{I},\theta} = ((\exists R.C)\theta|_{\Upsilon})^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{N}_V$, including when $\Upsilon \subseteq \mathsf{Var}(D) \cap \mathsf{Var}(E)$. Hence, (T10b) holds by definition of T.

Next, suppose that C is not a disjunction and $\mathsf{Tvar}(C) = \emptyset$. Then, by definition of T, $s \in (\exists R.C)^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Since $\mathsf{Tvar}(C) = \emptyset$, there is a $t \in \Delta^{\mathcal{I}}$ such that $\langle s, t\rangle \in R^{\mathcal{I}} = \mathcal{E}(R)$ and for every $\theta \in \Phi$, $t \in C^{\mathcal{I},\theta}$. Note that the latter holds because $\mathsf{Tvar}(C) = \emptyset$. If $\mathsf{Tvar}(C) \neq \emptyset$, then C is in one of the following form for some $v \in \mathsf{N}_V$: (i) $\{v\}$; (ii) $\{v\} \sqcap C'$; or (iii) $(\{v\} \sqcup C'') \sqcap C'$. For such forms of C, the existence of only one such t may not suffice because $\{v\}$ can be grounded into different individual names which may entail the existence of more than one R-neighbor for s. Now, since $\mathsf{Tvar}(C) = \emptyset$, we have that one such t satisfies $t \in C^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. Hence, $C \in \mathcal{L}(t)$ by construction, and thus, (T10c) is established.

For (T10d), assume that C is not a disjunction and $\mathsf{Tvar}(C) \neq \emptyset$. Then, by definition of T and Lemma 3, $s \in (\exists R.C)^{\mathcal{I},\theta} = (\exists R.C\theta)^{\mathcal{I},\theta} = (\exists R.C\theta|_{\Upsilon})^{\mathcal{I},\theta}$ for every $\theta \in \Phi$ and $\Upsilon \subseteq \mathsf{N}_V$, including when $\Upsilon \subseteq \mathsf{Tvar}(C)$. (T10d) thus holds by definition of T and since Υ exists.

(T11): Let $C \in \mathcal{L}(s)$. By definition of T, $s \in C^{\mathcal{I},\theta}$ for every $\theta \in \Phi$. By Lemma 3, $s \in (C\theta|_{\Upsilon})^{\mathcal{I},\theta}$ for every $\Upsilon \subseteq \mathsf{N}_V$. Hence by definition of T, in particular, (i), (ii), (iii) and (iv) are satisfied.

We now have formalized the notion of tableau that corresponds to a model of a TBox. This will be used to establish the soundness and completeness of the al- gorithm. Intuitively speaking, what we do is to establish a correspondence between tableaux and complete, clash-free completion graphs.

Lemma 6 (Soundness). *Let $\mathcal{T}$ be an $\mathcal{ALCOV}$ TBox. If the tableau expansion rules are applied to the initial completion graph for $\mathcal{T}$ in such a way that they yield in a complete and clash-free completion graph for $\mathcal{T}$, then $\mathcal{T}$ is satisfiable.*

Proof (of Lemma 6). Let $\mathbf{G} = (V, E, \mathsf{L}, \mathsf{M})$ be a complete and clash-free completion graph for $\mathcal{T}$ obtained from exhaustively applying the tableau expansion rules to the initial completion graph for $\mathcal{T}$. By Lemma 5, it suffices to show that a tableau for $\mathcal{T}$ exists. We construct such a tableau by standard unraveling where the idea is that each blockable node x is uniquely represented as a path to x that is rooted in some nominal node of $\mathbf{G}$.

Define a *path* p as a sequence of blockable nodes of the form $p \coloneqq [x_0, \ldots, x_n]$, $n \geqslant 0$. For such a path p, we define $\tau(p) \coloneqq x_n$. Given a path p and a blockable node x, $[p|x]$ is the path obtained by appending x to end of p. Let $\Omega(\mathbf{G})$ be the set of all nominal nodes in $\mathbf{G}$. The set $\Pi(\mathbf{G})$ of all paths in $\mathbf{G}$ is inductively defined as:

- if x is a blockable node that is a successor of a nominal node in $\mathbf{G}$, then $[x] \in \Pi(\mathbf{G})$;
- for each path $p \in \Pi(\mathbf{G})$ and a blockable successor x of $\tau(p)$, if x is not blocked, then $[p|x] \in \Pi(\mathbf{G})$, whereas if x is blocked, then $[p|b(x)] \in \Pi(\mathbf{G})$ where $b(x)$ is the node that blocks x.

Observe that every path contains no blocked node and for any two different blockable nodes x and y, $b(x) \neq b(y)$. Moreover, it is easy to see that for each blockable node x, there is exactly one path $p \in \Pi(\mathbf{G})$ such that $\tau(p) = x$.

Define a tuple $T = (S, \mathcal{L}, \mathcal{E})$ where

$$
\begin{aligned}
S &\coloneqq \Omega(\mathbf{G}) \cup \Pi(\mathbf{G});\\
\mathcal{L}(s) &\coloneqq \begin{cases} \mathsf{L}(s) & \text{if } s \in \Omega(\mathbf{G});\\ \mathsf{L}(\tau(s)) & \text{if } wes \in \Pi(\mathbf{G}) \end{cases}\\
\mathcal{E}(R) &\coloneqq \{\langle s,t\rangle \in \Omega(\mathbf{G}) \times \Omega(\mathbf{G}) \mid \langle s,t\rangle \in E \text{ and } R \in \mathsf{L}(s,t)\}\\
&\quad \cup \{\langle s,p\rangle \in \Omega(\mathbf{G}) \times \Pi(\mathbf{G}) \mid \langle s,\tau(p)\rangle \in E \text{ and } R \in \mathsf{L}(s,\tau(p))\}\\
&\quad \cup \{\langle p,s\rangle \in \Pi(\mathbf{G}) \times \Omega(\mathbf{G}) \mid \langle \tau(p),s\rangle \in E \text{ and } R \in \mathsf{L}(\tau(p),s)\}\\
&\quad \cup \{\langle p,q\rangle \in \Pi(\mathbf{G}) \times \Pi(\mathbf{G}) \mid q = \langle p|x\rangle, \langle \tau(p),x\rangle \in E \text{ and } R \in \langle \tau(p),x\rangle\}
\end{aligned}
$$

It remains to verify that T is a tableau for $\mathcal{T}$, i.e., it satisfies (T0)–(T10).

(T0): Satisfied because $\mathbf{G}$ is complete, hence $TBox$-rule is no longer applicable.
(T1): Satisfied because the construction of $\mathbf{G}$ is started from the initial completion graph for $\mathcal{T}$.

(T2): Satisfied as $\mathbf{G}$ is complete; O-rule for merging two nodes is no longer applicable.
(T3): Satisfied as $\mathbf{G}$ is complete; hence, grv-rule is no longer applicable.
(T4): Satisfied since otherwise, there is some $a \in \mathsf{N}_I$ and $v \in \mathsf{N}_V$ such that $\{\{a\}, \neg\{v\}\} \subseteq \mathcal{L}(s)$. Because grv-rule is no longer applicable, it must be the case that $\neg\{a\} \in \mathcal{L}(s)$, making $\mathbf{G}$ contains a clash which contradicts the fact that $\mathbf{G}$ is clash-free.
(T5): Satisfied by construction of the initialization and clash-freeness of $\mathbf{G}$.
(T6): Satisfied because $\mathbf{G}$ is clash-free.
(T7): Satisfied because $\mathbf{G}$ is complete; $\sqcap$-rule is no longer applicable.
(T8): Satisfied because $\mathbf{G}$ is complete; $\forall$-rule is no longer applicable
(T9): Satisfied as $\mathbf{G}$ is complete, $\sqcup$-rule and $gr\sqcup$-rule are are no longer applicable.
(T10): Satisfied as $\mathbf{G}$ is complete, $\exists$-rule, $gr\exists$-rule and $gr\exists^{\sqcup}$-rule rules are no longer applicable.
(T11): Satisfied as $\mathbf{G}$ is complete, gr-rule is no longer applicable.

Lemma 7 (Completeness). *Let $\mathcal{T}$ be an $\mathcal{ALCOV}$ TBox. If $\mathcal{T}$ is satisfiable, then the tableau expansion rules can be applied to the initial completion graph for $\mathcal{T}$ such that they yield in a complete and clash-free completion graph for $\mathcal{T}$.*

Proof (of Lemma 7). By Lemma 5, it suffices to show that such a complete and clash-free completion graph for $\mathcal{T}$ can be constructed if a tableau for $\mathcal{T}$ exists. Hence, let $T = (S, \mathcal{L}, \mathcal{E})$ be a tableau for $\mathcal{T}$. We show that starting from the initial completion graph $\mathbf{G} = (V, E, \mathsf{L})$ for $\mathcal{T}$, we can apply the tableau expansion rules in Figure 1 in such a way that a complete and clash-free completion graph for $\mathcal{T}$ is generated.

Let $T = (S, \mathcal{L}, \mathcal{E})$ be a tableau for $\mathcal{T}$ and Φ the set of total variable assignment. We construct a complete and clash-free completion graph $\mathbf{G} = (V, E, \mathsf{L})$ such that there exists a mapping $\mu \colon V \to S$ that satisfies:

(C1) for all $s \in V$, $\mathsf{L}(s) \subseteq \mathcal{L}(\mu(s)) \cup \mathsf{pg}(\mathcal{L}(\mu(s)))$ where $\mathsf{pg}(\mathcal{L}(\mu(s))) \coloneqq \{C\theta_{\mathcal{Y}} \mid C \in \mathcal{L}(\mu(s)), \theta \in \Phi, \mathcal{Y} \subseteq \mathsf{Var}(C)\}$;
(C2) for each $s, t \in V$ and $R \in \mathsf{N}_R$, if $\langle s, t\rangle \in E$ and $R \in \mathsf{L}(s, t)$, then $\langle \mu(s), \mu(t)\rangle \in \mathcal{E}(R)$.

Let $\mathbf{G}_i = (V_i, E_i, \mathsf{L}_i, \mathsf{M}_i)$ (resp. μ_i) be the completion graph for $\mathcal{T}$ (resp. a mapping from V_i to S) obtained after the i-th rule application. Initially, we set $\mathbf{G}_0$ as an initial completion graph for $\mathcal{T}$: $V_0 \coloneqq \{r^a \mid a \in \mathsf{Ind}(\mathcal{T})\}$, $E_0 \coloneqq \emptyset$, $\mathsf{M}_0 \coloneqq \emptyset$ and $\mathsf{L}(r^a) \coloneqq \{a\}$, for every $a \in \mathsf{Ind}(\mathcal{T}) = \mathsf{N}_I$. By (T1), it holds that for each $a \in \mathsf{N}_I$, there exists an $s \in S$ such that $\{a\} \in \mathcal{L}(s)$. Thus, we set μ_0 for each $r^a \in V_0$ such that $\mu_0(r^a) \coloneqq s$ iff $\{a\} \in \mathcal{L}(s)$. Clearly, **(C1)** and **(C2)** are satisfied by μ_0.

Next, assume that, for some $i \geqslant 1$, we have constructed $\mathbf{G}_{i-1}$ and the corresponding μ_{i-1}, and we are at the point where the $(i-1)$-st rule application have just been done. We construct μ_i from μ_{i-1} while performing the i-th rule application. We assume that μ_{i-1} already satisfies **(C1)** and **(C2)** and we ensure that the i-the rule application does not violate them. The cases depend on

which rule is being applied. Below, let $r \in V_{i-1}$ be the node on which the i-th rule application is being done.

- *TBox*-rule is applied to some $C \sqsubseteq D \in \mathcal{T}$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{\mathsf{NNF}(\neg C \sqcup D)\}$ and $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$. We set $\mu_i \coloneqq \mu_{i-1}$. Here, **(C1)** and **(C2)** are not violated by μ_i since they are satisfied by μ_{i-1} and (T0) is satisfied by T.
- $\sqcap$-rule is applied to some $C \sqcap D \in \mathsf{L}_{i-1}(r)$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{C, D\}$, and $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated by μ_i since they are satisfied by μ_{i-1} and (T7) is satisfied by T.
- O-rule is applied to some nominal $\{a\} \in \mathsf{L}_{i-1}(r)$. Then, there exists an initial node $s \in V_{i-1}$ to which r is merged. Let $\mathfrak{P}(r)$ be the set of blockable successor of r that are pruned by O-rule. So,

$$
\begin{aligned}
V_i &\coloneqq V_{i-1} \setminus (\{r\} \cup \mathfrak{P}(r)) \\
E_i &\coloneqq (E_{i-1} \setminus \{\langle t, t'\rangle, \langle t', t\rangle \mid t \in \mathfrak{P}(r)\}) \cup \{\langle r', s\rangle \mid \langle r', r\rangle \in E_{i-1}\} \\
&\quad \cup \{\langle s, r'\rangle \mid \langle r, r'\rangle \in E_{i-1}, r \text{ not blockable}\} \\
\mathsf{L}_i(t) &\coloneqq \begin{cases} \mathsf{L}_{i-1}(t) \cup \mathsf{L}_{i-1}(r), & \text{when } t = s, \\ \mathsf{L}_{i-1}(t), & \text{when } t \neq s, t \in V_i, \\ \text{undefined}, & \text{otherwise} \end{cases} \\
\mathsf{M}_i(t) &\coloneqq \begin{cases} \mathsf{M}_{i-1}(t) \cup \mathsf{M}_{i-1}(r), & \text{when } t = s, \\ \mathsf{M}_{i-1}(t), & \text{when } t \neq s, t \in V_i, \\ \text{undefined}, & \text{otherwise} \end{cases} \\
\mu_i &\coloneqq \mu_{i-1} \setminus \{\langle t, \mu_{i-1}(t)\rangle \mid t \in \{r\} \cup \mathfrak{P}(r)\}
\end{aligned}
$$

  Here, V_i, E_i, L_i, and M_i are obtained from V_{i-1}, E_{i-1}, L_{i-1} and M_{i-1} after merging r to s and the subsequent pruning of blockable successors of r. We set μ_i as μ_{i-1} but restricted only to nodes in V_i. T satisfies (T1) and (T2) which imply that $\mu_{i-1}(r) = \mu_{i-1}(s) = \mu_i(s)$. Since μ_{i-1} satisfies **(C1)** and **(C2)**, we have
  - $\mathsf{L}_{i-1}(s) \subseteq \mathcal{L}(\mu_{i-1}(s)) \cup \mathsf{pg}(\mathcal{L}(\mu_{i-1}(s)))$; and
  - $\mathsf{L}_{i-1}(r) \subseteq \mathcal{L}(\mu_{i-1}(r)) \cup \mathsf{pg}(\mathcal{L}(\mu_{i-1}(r))) = \mathcal{L}(\mu_{i-1}(s)) \cup \mathsf{pg}(\mathcal{L}(\mu_{i-1}(s)))$.

  So, $\mathsf{L}_i(s) = \mathsf{L}_{i-1}(s) \cup \mathsf{L}_{i-1}(r) \subseteq \mathcal{L}(\mu(s)) \cup \mathsf{pg}(\mathcal{L}(\mu_i(s)))$, ensuring **(C1)** and **(C2)** are not violated.
- $\forall$-rule is applied to some $\forall R.C \in \mathsf{L}_{i-1}(r)$. Then, $\langle r, s\rangle \in E$ and $R \in \mathsf{L}_{i-1}(r, s)$ for some $s \in V_{i-1}$. Applying this rule yields $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$, $\mathsf{L}_i(s) \coloneqq \mathsf{L}_{i-1}(s) \cup \{C\}$, and $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq s$. Hence, we set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated by μ_i because T satisfies (T8).
- $\sqcup$-rule is applied to some $C \sqcup D \in \mathsf{L}_{i-1}(r)$. Then, $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for all $t \neq r$. Additionally, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{C'\}$ for some $C' \in \{C, D\}$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T9).

– $\exists^{\sqcup}$-rule is applied to some $\exists R.(C \sqcup D) \in \mathsf{L}_{i-1}(r)$. Then, $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$, and $\mathsf{L}(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{\exists R.C'\}$ for some $C' \in \{C, D\}$. Set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T10a).
– $\exists$-rule is applied to some $\exists R.C \in \mathsf{L}_{i-1}(r)$. Then, C is not a disjunction and $\mathsf{Tvar}(C) = \emptyset$. Thus, $V_i \coloneqq V_{i-1} \cup \{s\}$, $E_{i-1} \coloneqq E_{i-1} \cup \{\langle r, s\rangle\}$, $\mathsf{L}_i(s) \coloneqq \{C\}$, $\mathsf{M}_i(s) \coloneqq \emptyset$ where s is a new node. In addition, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ and $\mathsf{M}_i(t) \coloneqq \mathsf{M}_{i-1}(t)$ for every $t \in V_{i-1}$. Set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T10c).
– grv-rule is applied to some $\{v\} \in \mathsf{L}_{i-1}(r)$, $v \in \mathsf{N}_V$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for all $t \neq r$. Also, for some $a \in \mathsf{N}_I$, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{D[v/a] \mid D \in \mathsf{L}_{i-1}(r)\}$ and $\mathsf{M}_i(r) \coloneqq \mathsf{M}_{i-1}(r) \cup \{v/a\}$. Hence, we set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated by μ_i as T satisfies (T3) and (T11).
– $gr\sqcup$-rule is applied to some $C \sqcup D \in \mathsf{L}_{i-1}(r)$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$, and $\mathsf{M}_i(t) \coloneqq \mathsf{M}_{i-1}(t)$ for $t \neq r$. Further, $\mathsf{L}_i(r) : = \mathsf{L}_{i-1}(r) \cup \{C'[v/a] \mid C' \in \mathsf{L}_{i-1}(r)\}$ and $\mathsf{M}_i(r) \coloneqq \mathsf{M}_{i-1}(r) \cup \{v/a\}$ for some $v \in \mathsf{Var}(C) \cap \mathsf{Var}(D)$ and $a \in \mathsf{N}_I$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T9b) and (T11).
– $gr\exists^{\sqcup}$-rule is applied to some $\exists R.(C \sqcup D) \in \mathsf{L}_{i-1}(r)$. Then, $V_i \coloneqq V_{i-1}$, $E_i : = E_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$, and $\mathsf{M}_i(t) \coloneqq \mathsf{M}_{i-1}(t)$ for $t \neq r$. Further, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{C'[v/a] \mid C' \in \mathsf{L}_{i-1}(r)\}$ and $\mathsf{M}_i(r) \coloneqq \mathsf{M}_{i-1}(r) \cup \{v/a\}$ for some $v \in \mathsf{Var}(C) \cap \mathsf{Var}(D)$ and $a \in \mathsf{N}_I$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T10b) and (T11).
– $gr\exists$-rule is applied to some $\exists R.C \in \mathsf{L}_{i-1}(r)$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$, and $\mathsf{M}_i(t) \coloneqq \mathsf{M}_{i-1}(t)$ for $t \neq r$. Further, $\mathsf{L}_i(r) : = \mathsf{L}_{i-1}(r) \cup \{C'[v/a] \mid C' \in \mathsf{L}_{i-1}(r)\}$ and $\mathsf{M}_i(r) \coloneqq \mathsf{M}_{i-1}(r) \cup \{v/a\}$ for some $v \in \mathsf{Var}(C) \cap \mathsf{Var}(D)$ and $a \in \mathsf{N}_I$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T10d) and (T11).
– gr-rule is applied to some $C \in \mathsf{L}_{i-1}(r)$. Then, $V_i \coloneqq V_{i-1}$, $E_i \coloneqq E_{i-1}$, $\mathsf{L}_i(t) \coloneqq \mathsf{L}_{i-1}(t)$ for $t \neq r$, and $\mathsf{M}_i \coloneqq \mathsf{M}_{i-1}$. Further, $\mathsf{L}_i(r) \coloneqq \mathsf{L}_{i-1}(r) \cup \{C'[v/a] \mid C' \in \mathsf{L}_{i-1}(r)\}$ for some $v/a \in \mathsf{M}_{i-1}(r)$. We set $\mu_i \coloneqq \mu_{i-1}$. **(C1)** and **(C2)** are not violated because T satisfies (T11).

Because any sequence of rule applications terminate, there exists an $n \in \mathbb{N}$ for which no rule is applicable to $\mathbf{G}_n$. So, $\mathbf{G}_n$ is complete. Furthermore, as **(C1)** and **(C2)** are satisfied by $\mathbf{G}_n$ and μ_n, we have that $\mathbf{G}_n$ is clash-free because:

(i) If $\bot \in \mathsf{L}_n(r)$ for some $r \in V_n$, then either $\bot \in \mathcal{L}(\mu_n(r))$, or $\bot \in \{C\theta_{\Upsilon} \mid C \in \mathcal{L}(\mu_n(r)), \theta \in \Phi, \Upsilon \subseteq \mathsf{Var}(C)\}$. Here, the second case implies the first case, while the first case itself contradicts the property (T5) of T.
(ii) If $\{A, \neg A\} \subseteq \mathsf{L}_n(r)$ for some $r \in V_n$ where A is a concept name, then it must be the case that $\{A, \neg A\} \subseteq \mathcal{L}(\mu_n(r))$ which contradicts the property (T6) of T. Note that whenever either A or $\neg A$ is in the set $\{C\theta_{\Upsilon} \mid C \in \mathcal{L}(\mu_n(r)), \theta \in \Phi, \Upsilon \subseteq \mathsf{Var}(C)\}$, it must also be in $\mathcal{L}(\mu_n(r))$.
(iii) If $\{\{a\}, \neg\{a\}\} \subseteq \mathsf{L}_n(r)$ for some $r \in V_j$, $a \in \mathsf{N}_I$, then it must be the case that $\{\{a\}, \neg\{a\}\} \subseteq \mathcal{L}(\mu_n(r))$ which contradicts the property (T6) of T. Note that if $\{a\} \in \{C\theta_{\Upsilon} \mid C \in \mathcal{L}(\mu_n(r)), \theta \in \Phi, \Upsilon \subseteq \mathsf{Var}(C)\}$, then either

already $\{a\} \in \mathcal{L}(\mu_n(r))$, or $\{v\} \in \mathcal{L}(\mu_n(r))$ which, by (T1), implies that $\{a\} \in \mathcal{L}(\mu_n(r))$. This holds analogously for $\neg\{a\}$.

5 Delayed Grounding for More Expressive DLs

The framework proposed in this paper can be generalized to more expressive DLs. The general strategy for such a generalization can be realized by adding some preconditions related to occurrences of nominal schemas to the standard tableau rules, if necessary, and then specifying appropriate grounding rules. We will briefly discuss how this can be done for many standard DL constructors known from the literature.

First, it is rather obvious that many role constructors will be not be affected by nominal schemas. This includes inverse roles, role hierarchy and role chains (which generalizes many role characteristics such as transitivity, symmetry, etc.) and Boolean role constructors. For those constructors, known techniques in existing tableau algorithms should be readily usable without modification when the corresponding DLs are extended with nominal schemas. For example, automata translation of role chains in $\mathcal{SROIQ}$ [3] can be readily used including the corresponding rules for universal restrictions. This line of reasoning can also be applied when dealing with unqualified number restriction, and self-restriction ($\exists R.\mathsf{Self}$) as both do not involve nominal schemas.

On the other hand, the adaptation is not so obvious for the only prominent concept constructor that is not discussed in this paper, but may have some interactions with nominal schemas, namely qualified number restrictions. For the first variant, at-least restrictions, i.e., concepts of the form $(\geqslant n\ R.C)$ where $n \in \mathbb{N}$, the behavior is similar to existential restriction. Unfortunately, the approach we use here where we distinguish the filler of an existential restriction into a disjunction and non-disjunction may not be usable. The reason is that although $\exists R.(C \sqcup D) \equiv \exists R.C \sqcup \exists R.D$ holds in general whenever $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$, it does not hold for at-least restrictions, i.e., $(\geqslant n\ R.(C \sqcup D)) \not\equiv (\geqslant n\ R.C) \sqcup (\geqslant n\ R.D)$ even if $\mathsf{Var}(C) \cap \mathsf{Var}(D) = \emptyset$. The only adaptation that can be made is when the filler is not a disjunction. That is, if a concept $(\geqslant n\ R.C)$ where C is not a disjunction appears in a node labeling, the standard tableau expansion rule (i.e., at-least-restriction rule, see e.g., $\geqslant$-rule in [14]) that creates n different individuals, all satisfying C, can be applied as long as $\mathsf{Tvar}(C) = \emptyset$. This means that grounding before the at-least-restriction rule can be applied is only necessary for those variables in $\mathsf{Tvar}(C)$. On the other hand, when C is actually a disjunction, we may need to completely ground C before at-least-restriction can be applied.

For at-most restriction, i.e., concepts of the form $(\leqslant n\ R.C)$, the standard tableau expansion rule (e.g., $\leqslant$-rule in [14]) is applied when there are more than n role-fillers of the corresponding node that satisfy C. In this case, two of them are merged, similar to merging that is done for O-rule. The situation is then a bit simpler than at-least-restriction, because this rule does not involve the

propagation of concept labels. Therefore, it can also be used before performing grounding.

6 Conclusion

We have presented a tableau algorithm for the description logic $\mathcal{ALCOV}$ which is obtained through non-trivial modifications of standard tableau algorithms for $\mathcal{ALCO}$. It improves on the only known algorithm to deal with nominal schemas, which is based on full grounding, by applying grounding in a selective and delayed fashion. We have provided correctness results and an example which shows the advantages of our approach over full grounding, by significantly pruning the size of the tableau which needs to be constructed. Although the basic idea of our approach seems to be intuitively simple (just ground when needed) and described only for selected basic DL constructors, the approach can be adapted easily to deal with more expressive DL constructors known in the literature.

In terms of realization of reasoning with nominal schemas, this paper is only the beginning of our investigations. While our algorithm provides a flexible way of delayed and selective grounding, for implementations it will be necessary to obtain good heuristics for grounding choices. Finally, other automated reasoning approaches, e.g., such based on resolution, may provide even better algorithmizations for DLs with nominal schemas.

References

1. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
2. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S., eds.: OWL 2 Web Ontology Language: Primer. W3C Recommendation 27 October 2009 (2009) Available from http://www.w3.org/TR/owl2-primer/.
3. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible SROIQ. In Doherty, P., Mylopoulos, J., Welty, C., eds.: Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006), AAAI Press (2006) 57–67
4. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. Journal of the ACM **42** (1995) 741–843
5. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D., eds.: RIF Core Dialect. W3C Recommendation 22 June 2010 (2010) Available from http://www.w3.org/TR/rif-core/.
6. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In Polleres, A., et al., eds.: Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Tutorial Lectures. Volume 6848 of Lecture Notes in Computer Science., Springer, Heidelberg (2011) 382–415
7. Motik, B., Sattler, U., Studer, R.: Query answering for OWL DL with rules. Journal of Web Semantics **3**(1) (2005) 41–60
8. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable Rules for OWL 2. In Sheth, A.P., et al., eds.: Proceedings of the 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Volume 5318 of Lecture Notes in Computer Science., Springer (2008) 649–664

9. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In Sadagopan, S., Ramamritham, K., Kumar, A., Ravindra, M., Bertino, E., Kumar, R., eds.: Proceedings of the 20th International World Wide Web Conference, WWW2011, Hyderabad, India, March/April 2011, ACM, New York (2011) 645–654
10. Carral Martinez, D., Krisnadhi, A., Maier, F., Sengupta, K., Hitzler, P.: Reconciling OWL and rules. Technical report, Kno.e.sis Center, Wright State University, Dayton, Ohio, U.S.A. (2011) Available from http://www.pascal-hitzler.de/.
11. Schmidt, R.A., Tishkovsky, D.: A general tableau method for deciding description logics, modal logics and related first-order fragments. Automated Reasoning (2008) 194–209
12. Horrocks, I., Sattler, U.: Ontology reasoning in the SHOQ(D) description logic. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI). Volume 1. (2001) 199–204
13. Hladik, J., Model, J.: Tableau systems for SHIO and SHIQ. In Haarslev, V., Möller, R., eds.: Proceedings of the 2004 International Workshop on Description Logics (DL 2004), CEUR (2004)
14. Horrocks, I., Sattler, U.: A tableau decision procedure for shoiq. Journal of Automated Reasoning **39**(3) (2007) 249–276
15. Krisnadhi, A., Hitzler, P.: A tableau algorithm for description logics with nominal schemas. Technical report, Kno.e.sis Center, Wright State University, Dayton, OH, U.S.A. (2012) Available from http://knoesis.wright.edu/researchers/adila/pmwiki/uploads/Main/KriHitz-TableauNS-TR12.pdf.