# Extending Absorption to Nominal Schemas

Andreas Steigmiller[1], Birte Glimm[1], and Thorsten Liebig[2]

[1] Ulm University, Ulm, Germany, <first name>.<last name>@uni-ulm.de
[2] derivo GmbH, Ulm, Germany, liebig@derivo.de

**Abstract.** Nominal schemas have recently been introduced as a new approach for the integration of DL-safe rules into the Description Logic framework. The efficient processing of knowledge bases with nominal schemas remains, however, challenging. We address this by extending the well-known optimisation of absorption as well as the standard tableau calculus to directly handle the (absorbed) nominal schema axioms. We implement the resulting extension of standard tableau calculi in a novel reasoning system and we integrate further optimisations. In our empirical evaluation, we show the effect of these optimisations and we find that the proposed approach performs well even when compared to other DL reasoners with dedicated rule support.

## 1 Introduction

We address the problem of an efficient handling of so-called nominal schema axioms in tableau calculi for Description Logics (DLs). Nominal schemas have been introduced recently [11] as a feature for expressing arbitrary DL-safe rules (as specified in the W3C standards SWRL [5] or RIF [9]) natively in DLs and, consequently, in OWL ontologies [13]. Hence, DLs with nominal schemas provide a unified basis for OWL and rules. Although some attempts (see, e.g., [10]) have been made to improve the performance of tableau calculi when extended with nominal schemas, handling of nominal schemas remains challenging. We tackle this problem by extending the well-know tableau optimisation of absorption [7]. The resulting calculus extends a standard tableau calculus by additional rules to deal with the absorbed nominal schema axioms and shows a considerable performance improvement over existing techniques.

Nominal schemas extend the nominal constructor that is present in many DLs and which allows for specifying a concept as a singleton set with a named individual as member, e.g., the interpretation of the concept $\{a\}$ consists of the element that represents the named individual $a$. Nominal schemas introduce a new concept constructor $\{x\}$, where $x$ is a variable that can only be bound to a named individual from the ABox of the knowledge base. This restriction ensures decidability and is common for nominal schemas as well as for SWRL rules.

We use the same running example as Krisnadhi and Hitzler [10], which describes a conflicting review assignment for an individual who has to review a paper $x$ that has an author $y$ with whom that individual has a joint publication in the same venue $z$:

$$\begin{aligned}\exists \mathit{hasReviewAssignment}.(\{x\} \sqcap \exists \mathit{hasAuthor}.\{y\} \sqcap \exists \mathit{atVenue}.\{z\}) \\ \sqcap \exists \mathit{hasSubmittedPaper}.(\exists \mathit{hasAuthor}.\{y\} \sqcap \exists \mathit{atVenue}.\{z\}) \\ \sqsubseteq \exists \mathit{hasConflictingAssignedPaper}.\{x\}.\end{aligned}$$

For brevity, we shorten *hasReviewAssignment* to *r*, *hasAuthor* to *a*, *atVenue* to *v*, *hasSubmittedPaper* to *s*, and *hasConflictingAssignedPaper* to *c* in the remainder. Obviously, this axiom can neither be directly expressed in a DL knowledge base nor as ordinary DL-safe rule (e.g., if we were to express the complex concepts as role atoms, we would have to introduce a variable for the submitted paper, which then would only bind to known ABox individuals). However, such nominal schema axioms can be eliminated by *upfront grounding*, i.e., by replacing nominal schema axioms with all possible grounded axioms obtained by replacing nominal schemas with nominals, where the nominal schemas with the same variable are always replaced by the same nominal. Upfront grounding is, however, very inefficient. For example, a nominal schema axiom with 3 variables can be grounded for a knowledge base with 100 ABox individuals in $100^3$ different ways, which is prohibitive even for small examples.

A promising approach for efficient reasoning in OWL DL ontologies extended with nominal schemas, i.e., $\mathcal{SROIQV}$ knowledge bases with $\mathcal{V}$ denoting nominal schemas, is to adapt established tableau algorithms (e.g., [4]), which are dominantly used for sound and complete reasoning systems for expressive DLs. One such approach extends a tableau algorithm such that grounding is *delayed* until it is *required* [10]. However, this requires significant changes to the tableau algorithm and, thus, to existing optimisations, which are crucial for a reasonable performance on real-world ontologies. Furthermore, it is not clear in which way concepts have to be grounded for a well-performing implementation and some concepts even cannot be grounded efficiently.

In this paper, we present a novel approach that works by collecting possible bindings for the nominal schema variables during the application of tableau rules; then, these bindings are used to complete the processing of the nominal schema axioms. For this, we extend the widely used technique of absorption (Section 3) to handle nominal schemas (Section 4.1), and we adapt or add new rules to the tableau calculus (Section 4.2). We further sketch optimisations and empirically evaluate the proposed approach (Section 5), before we conclude (Section 6). Further details and an extended evaluation are available in a technical report [15]. Proofs are also in the appendix.

## 2 Preliminaries

For brevity, we do not introduce DLs (see, e.g., [1]) and we only give a short overview about the used tableau algorithm in the following (for details see, e.g., [4]). We present our approach for $\mathcal{ALCOIQV}$, however, covering $\mathcal{SROIQV}$ is easily possible: role chains can be encoded [4] and the remaining $\mathcal{SROIQ}$ features are easy to support.

Roughly speaking, the tableau algorithm constructs for an input knowledge base $\mathcal{K}$ a completion graph $G = (V, E, \mathcal{L}, \dot{\neq})$ by decomposing complex concepts with a set of expansion rules. Each node $x \in V$ (edge $\langle x, y\rangle \in E$) is labelled with the set of concepts $\mathcal{L}(x)$ (set of roles $\mathcal{L}(\langle x, y\rangle)$) and $\dot{\neq}$ records inequalities between nodes. $G$ is initialised with one node for each ABox individual/nominal in the input knowledge base (w.l.o.g. we assume that the ABox is non-empty). In order to guarantee that each node of the completion graph indeed satisfies all TBox axioms, one can use a tableau rule that checks, for each general concept inclusion (GCI) $C \sqsubseteq D$, whether $C$ is satisfied for a node and only then adds $D$ to the node label. Checking whether a complex left-hand

side is satisfied can, however, be non-trivial. In order to guarantee correctness, one treats such axioms where $C$ is complex as $\top \sqsubseteq \neg C \sqcup D$. Given that $\top$ is satisfied at each node, the disjunction $\neg C \sqcup D$ is then added to the label of each node. In practise, one uses elaborate transformations in a preprocessing step called absorption to avoid axioms of the form $\top \sqsubseteq \neg C \sqcup D$.

Roughly speaking, the absorption algorithm extracts those conditions of a disjunction for which it can be ensured that if one of these conditions is not satisfied for a node in a completion graph, then at least one alternative of the disjunction is trivially satisfiable. These conditions are then used for expressing the disjunction in such a way that non-determinism can be avoided as much as possible in the tableau algorithm. For example, one would like to avoid treating $\exists r.(A_1 \sqcup A_2) \sqsubseteq \exists s.A$ as $\top \sqsubseteq \forall r.(\neg A_1 \sqcap \neg A_2) \sqcup \exists s.A$. Any node that does *not* have an $r$-neighbour trivially satisfies $\forall r.(\neg A_1 \sqcap \neg A_2)$ and, hence, the overall disjunction. Thus, we could only add the disjunction to nodes that have at least one $r$-successor. We can, however, go even further by first identifying nodes that satisfy $A_1$ or $A_2$ and then make sure that their $r^-$-neighbour satisfies $\exists s.A$. Hence, the disjunctive axiom can be rewritten into $A_1 \sqsubseteq T$, $A_2 \sqsubseteq T$ and $T \sqsubseteq \forall r^-.(\exists s.A)$, where $T$ is a fresh atomic concept. Here, $A_1$ and $A_2$ have been *absorbed* (i.e., moved to the left-hand side of the axiom) and the concept $T$ is used to enforce the semantics of the original axiom. We call $\forall r.(\neg A_1 \sqcap \neg A_2)$ *completely absorbable* since it no longer contributes a disjunct. The goal of the absorption preprocessing step is, therefore, the extraction of such easy to verify conditions that allow for expressing a GCI by possibly several axioms that ideally do not require a disjunction.

In order to absorb more complex concepts it is often necessary to join several conditions, say $A_1$ to $A_n$. An efficient way to do this is *binary absorption* [8], where two concepts $A_1$ and $A_2$ imply a fresh atomic concept $T_1$ by the axiom $(A_1 \sqcap A_2) \sqsubseteq T_1$. We can then combine $T_1$ with the next condition $A_3$ and so on, until $(T_{n-2} \sqcap A_n) \sqsubseteq T_{n-1}$, where $T_{n-1}$ can then be used for further absorption.

## 3 Absorption Algorithm

Since our handling of nominal schemas is based on absorption methods, we next present an improved variant of a recursive binary absorption algorithm, which we then extend to nominal schemas in the next section. The improvements allow for absorbing parts of the axioms partially without creating additional disjunctions. For example, the TBox axiom $\exists r.(A \sqcap \forall r.C) \sqsubseteq D$ is, without absorption, handled as $\top \sqsubseteq \forall r.(\neg A \sqcup \exists r.\neg C) \sqcup D$. None of the disjuncts can be absorbed completely, but it is nevertheless possible to delay the processing of the disjunction until there is an $r$-neighbour with the concept $A$ in its label. In order to capture this, the absorption rewrites the axiom such that the disjunction is propagated from a node with $A$ in its label to all $r^-$-neighbours (if there are any), which results in $A \sqsubseteq \forall r^-.(\forall r.(\neg A \sqcup \exists r.\neg C) \sqcup D)$.

In the following, $C_{(i)}, D_{(i)}$ are (possibly complex) concepts, $A_{(i)}, T_{(i)}$ are atomic concepts with $T_{(i)}$ used for fresh concepts and $S$ is a set of concepts. We assume that all concepts are in the well-known negation normal form (NNF) or we use $\mathsf{nnf}(C)$ to transform a concept $C$ to an equivalent one in NNF. Our algorithm uses the following functions to absorb axioms of a TBox $\mathcal{T}$ into a new (global) TBox $\mathcal{T}'$:

**Algorithm 1** isCA($C$) and isPA($C$)

**Output:** Returns whether the concept $C$ is completely absorbable

```
 1: procedure isCA(C)
 2:     if C = C1 ⊔ C2 then
 3:         return isCA(C1) ∧ isCA(C2)                ▷
 4:     else if C = C1 ⊓ C2 then
 5:         return isCA(C1) ∧ isCA(C2)
 6:     else if C = ∀r.C′ then
 7:         return isCA(C′)                           ▷
 8:     else if C = ¬{a} then
 9:         return true
        ...
10:     else if C = ¬A then
11:         return true
12:     end if
13:     return false
14: end procedure
```

**Output:** Returns whether the concept $C$ is partially absorbable

```
 1: procedure isPA(C)
 2:     if C = C1 ⊔ C2 then
 3:         return isPA(C1) ∨ isPA(C2)                ▷
 4:     else if C = C1 ⊓ C2 then
 5:         return isPA(C1) ∧ isPA(C2)
 6:     else if C = ∀r.C′ then
 7:         return true                               ▷
 8:     else if C = ¬{a} then
 9:         return true
        ...
10:     else if C = ¬A then
11:         return true
12:     end if
13:     return false
14: end procedure
```

**Algorithm 2** collectDisjuncts($C$, *absorbable*)

**Output:** Returns the absorbable/not absorbable disjuncts of the concept $C$

```
1: S ← {C}
2: while (C1 ⊔ C2) ∈ S do
3:     S ← (S \ (C1 ⊔ C2)) ∪ {C1, C2}
4: end while
5: if absorbable = true then return { C ∈ S | isPA(C) }
6: else return { C ∈ S | ¬isCA(C) }
7: end if
```

- isCA($C$) (isPA($C$)), shown in Algorithm 1, returns whether the concept $C$ is completely (partially) absorbable. We have tagged the lines 3 and 7 with a comment symbol to highlight where isPA might allow additional absorption in comparison to isCA. We have indicated with "..." that the absorption can further be extended to other constructors, e.g., to constructors of more expressive DLs [15].
- collectDisjuncts($C$, *absorbable*), shown in Algorithm 2, returns the set of (completely or partially) absorbable disjuncts for $C$ if *absorbable* = *true* and the set of not completely absorbable disjuncts otherwise. If $C$ is not a disjunction, then $\{C\}$ itself is returned, in case it conforms to the specified *absorbable* condition.

For simplicity, we assume here that axioms of the form $C \equiv D$ are rewritten into $C \sqsubseteq D$ and $D \sqsubseteq C$. An extension that directly and, hence, more efficiently handles axioms of the form $A \equiv C$ is also possible [15].

To obtain the absorbed TBox $\mathcal{T}'$, we call for each axiom $C \sqsubseteq D \in \mathcal{T}$ the function absorbJoined for the set of absorbable disjuncts, i.e., collectDisjuncts(nnf($\neg C \sqcup D$), *true*), which returns a fresh atomic concept that is used to imply a disjunction of the non-absorbable disjuncts, i.e., collectDisjuncts(nnf($\neg C \sqcup D$), *false*). The methods absorbJoined (Algorithm 3) and absorbConcept (Algorithm 4) are recursively calling

**Algorithm 3** absorbJoined($S$)

**Output:** Returns the atomic concept that is implied by the join of the absorptions of $S$

1: $S' \leftarrow \emptyset$
2: **for all** $C \in S$ **do**
3: $\quad A' \leftarrow$ absorbConcept($C$)
4: $\quad S' \leftarrow S' \cup \{A'\}$
5: **end for**
6: **while** $A_1 \in S'$ **and** $A_2 \in S'$ **and** $A_1 \neq A_2$ **do**
7: $\quad T \leftarrow$ fresh atomic concept
8: $\quad \mathcal{T}' \leftarrow \mathcal{T}' \cup \{(A_1 \sqcap A_2) \sqsubseteq T\}$
9: $\quad S' \leftarrow (S' \cup \{T\}) \setminus \{A_1, A_2\}$
10: **end while**
11: **if** $S' = \emptyset$ **then return** $\top$
12: **else return** the element $A' \in S'$ ▷ $S'$ is a singleton
13: **end if**

**Algorithm 4** absorbConcept($C$)

**Output:** Returns the atomic concept for the absorption of $C$

1: **if** $C = C_1 \sqcap C_2$ **then**
2: $\quad A_1 \leftarrow$ absorbJoined(collectDisjuncts($C_1, true$))
3: $\quad A_2 \leftarrow$ absorbJoined(collectDisjuncts($C_2, true$))
4: $\quad T \leftarrow$ fresh atomic concept
5: $\quad \mathcal{T}' \leftarrow \mathcal{T}' \cup \{A_1 \sqsubseteq T, A_2 \sqsubseteq T\}$
6: $\quad$ **return** $T$
7: **else if** $C = \forall r.C'$ **then**
8: $\quad A_{nb} \leftarrow$ absorbJoined(collectDisjuncts($C', true$))
9: $\quad T \leftarrow$ fresh atomic concept
10: $\quad \mathcal{T}' \leftarrow \mathcal{T}' \cup \{A_{nb} \sqsubseteq \forall r^-.T\}$
11: $\quad$ **return** $T$
12: **else if** $C = \neg\{a\}$ **then**
13: $\quad T \leftarrow$ fresh atomic concept
14: $\quad \mathcal{T}' \leftarrow \mathcal{T}' \cup \{\{a\} \sqsubseteq T\}$
15: $\quad$ **return** $T$
$\quad \ldots$
16: **else return** $A$ ▷ $C$ is of the form $\neg A$
17: **end if**

each other, whereby absorbJoined is joining several atomic concepts with binary absorption axioms and absorbConcept creates the absorption for a specific concept. For instance, a concept of the form $\forall r.C$ can be absorbed (lines 7–11 of Algorithm 4) by creating a propagation from the atomic concept $A_{nb}$, which is obtained for the absorption of $C$, back over the $r$-edge, to trigger a fresh atomic concept $T$. Note, if $C$ cannot be absorbed, then absorbJoined returns $\top$ and the axiom $\top \sqsubseteq \forall r^-.T$ is created, which corresponds to $\exists r.\top \sqsubseteq T$ and, thus, is similar to the well known *role absorption* [16].

The absorbJoined function creates binary absorption axioms (Algorithm 3, lines 6-10) for the atomic concepts returned by absorbConcept. Thus, absorbJoined is joining several conditions into one fresh atomic concept, which can be used for further absorp-

tion or to initiate the addition of the remaining and non-absorbable part of the axiom. One can further reduce the number of produced axioms by reusing absorption axioms for concepts that occur more then once.

One can show that concept satisfiability is indeed preserved for the absorbed TBox:

**Theorem 1** *Let $\mathcal{T}$ denote a TBox, $\mathcal{T}'$ the TBox obtained by absorbing $\mathcal{T}$, and $C$ a concept, then $C$ is satisfiable with respect to $\mathcal{T}$ iff it is satisfiable with respect to $\mathcal{T}'$.*

## 4 Nominal Schema Absorption

In contrast to DL-safe SWRL rules, the left-hand side of axioms with nominal schemas can be satisfied on arbitrary nodes in the completion graph (even though variables can only bind to nodes that represent individuals/nominals). As a consequence, axioms with nominal schemas can influence arbitrary nodes in the completion graph and, thus, blocking, which ensures termination, easily becomes unsound when typical approaches for rule processing, such as Rete [3], are used naively. Our approach to overcome this issue is to emulate such rule processing mechanisms by adapted tableau rules, which propagate bindings of variables for concepts through the completion graph. As a nice side-effect, this propagation means that also complex roles can easily be supported.

### 4.1 Absorption of Axioms with Nominal Schemas

The absorption of axioms with nominal schema variables works very similar to the absorption of ordinary axioms. We could directly extend the absorption algorithm to handle the new concept construct, however, to avoid some special cases for conjunctions $C_1 \sqcap C_2$ in an absorbable disjunct, where different nominal schema variables are used in $C_1$ and $C_2$, we require that conjunctions in absorbable positions are eliminated. This can be done by duplicating the disjunction that is absorbed and by replacing $C_1 \sqcap C_2$ once with $C_1$ and once with $C_2$. For example, the axiom $\{x\} \sqcup A \sqsubseteq \exists r.\{x\}$ is handled as the disjunction $(\neg\{x\} \sqcap \neg A) \sqcup \exists r.\{x\}$ in the absorption and to eliminate $\neg\{x\} \sqcap \neg A$ we replace the original axiom with $\{x\} \sqsubseteq \exists r.\{x\}$ and $A \sqsubseteq \exists r.\{x\}$.

For our absorption algorithm of Section 3, the following two modifications are necessary in order to handle nominal schemas in the remaining axioms:

- $\mathsf{isCA}(C)$ ($\mathsf{isPA}(C)$) is extended to return that a negated occurrence of a nominal schema $\neg\{x\}$ is completely (partially) absorbable.
- $\mathsf{absorbConcept}(C)$ of Algorithm 4 must now also handle a negated occurrence of a nominal schema $\neg\{x\}$ by absorbing it to $O \sqsubseteq \downarrow x.T_x$ for which the fresh atomic concept $T_x$ is returned and $O$ is a special concept that is added to the label of all ABox individuals.

The $\downarrow$ binder operator, as known from Hybrid Logics [2], is introduced to actually bind variables to individuals (or nodes in a completion graph). It is handled by a new tableau rule, which adds, for a node $a$ with $\downarrow x.T_x \in \mathcal{L}(a)$, $T_x$ to the label and records that $x$ is bound to $a$. In the remainder, we assume that knowledge bases contain, for each

**Table 1.** Tableau rule extensions to propagate variable mappings

| | | |
|---|---|---|
| $\forall$-rule: | if | $\forall r.C \in \mathcal{L}(v)$, $v$ not indirectly blocked, there is an $r$-neighbour $w$ of $v$ with $C \notin \mathcal{L}(w)$ or $\mathcal{B}(\forall r.C, v) \not\subseteq \mathcal{B}(C, w)$ |
| | then | $\mathcal{L}(w) \longrightarrow \mathcal{L}(w) \cup \{C\}$ and $\mathcal{B}(C, w) \longrightarrow \mathcal{B}(C, w) \cup \mathcal{B}(\forall r.C, v)$ |
| $\sqsubseteq_1$-rule: | if | $A \sqsubseteq C \in \mathcal{K}, A \in \mathcal{L}(v)$, $v$ not indirectly blocked, and $C \notin \mathcal{L}(v)$ or $\mathcal{B}(A, v) \not\subseteq \mathcal{B}(C, v)$ |
| | then | $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{C\}$ and $\mathcal{B}(C, v) \longrightarrow \mathcal{B}(C, v) \cup \mathcal{B}(A, v)$ |
| $\sqsubseteq_2$-rule: | if | $(A_1 \sqcap A_2) \sqsubseteq C \in \mathcal{K}$, $\{A_1, A_2\} \subseteq \mathcal{L}(v)$, $v$ not indirectly blocked, and<br>1. $\mathcal{B}(A_1, v) \cup \mathcal{B}(A_2, v) = \emptyset$ and $C \notin \mathcal{L}(v)$, or<br>2. $(\mathcal{B}(A_1, v) \bowtie^{\epsilon} \mathcal{B}(A_2, v)) \neq \emptyset$ and $C \notin \mathcal{L}(v)$ or $(\mathcal{B}(A_1, v) \bowtie^{\epsilon} \mathcal{B}(A_2, v)) \not\subseteq \mathcal{B}(C, v)$ |
| | then | $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{C\}$ and $\mathcal{B}(C, v) \longrightarrow \mathcal{B}(C, v) \cup (\mathcal{B}(A_1, v) \bowtie^{\epsilon} \mathcal{B}(A_2, v))$ |
| $\downarrow$-rule: | if | $\downarrow x.C \in \mathcal{L}(v)$, $v$ not indirectly blocked, and $C \notin \mathcal{L}(v)$ or $\{x \mapsto v\} \notin \mathcal{B}(C, v)$ |
| | then | $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{C\}$ and $\mathcal{B}(C, v) \longrightarrow \{\{x \mapsto v\}\}$ |
| $gr$-rule: | if | $gr(C) \in \mathcal{L}(v)$, $v$ not indirectly blocked, there exists a variable mapping $\mu \in \mathsf{comp}^{\mathcal{K}}_{\mathsf{Vars}(C)}(\mathcal{B}(gr(C), v))$ with $C_{[\mu]} \notin \mathcal{L}(v)$ |
| | then | $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{C_{[\mu]}\}$ |

individual $a$, an axiom of the form $\{a\} \sqsubseteq O$, where $O$ is a fresh atomic concept. Since the binders are, therefore, only added to ABox individuals (due to axioms of the form $O \sqsubseteq \downarrow x.T_x$), the decidability is retained, whereas the unrestricted extension of a Description Logic with binders easily leads to undecidability of the standard reasoning problems.

Other concepts can be absorbed as before, however, the final atomic concept $A$ created by the absorption cannot initiate the addition of the remaining, non-absorbed part of the axiom in the same way. If the remaining disjuncts $D_1, \ldots, D_n$ still contain nominal schemas, then the disjunction has to be grounded with those bindings of variables that have been propagated to $A$. In the tableau algorithm this can be done dynamically, e.g., with a new "grounding concept" and a corresponding rule. Therefore, if $D_1, \ldots, D_n$ still contain concepts with nominal schemas, then $A \sqsubseteq gr(D_1 \sqcup \ldots \sqcup D_n)$ has to be added to the TBox, where $gr(\cdot)$ is the new grounding concept. For simplicity, let us assume that $gr(C)$ is always used to add the remaining, non-absorbed part of the axiom, even if $C$ or the axiom does not contain any nominal schemas.

*Example 1.* Our running example $\exists r.(\{x\} \sqcap \exists a.\{y\} \sqcap \exists v.\{z\}) \sqcap \exists s.(\exists a.\{y\} \sqcap \exists v.\{z\}) \sqsubseteq \exists c.\{x\}$ can be almost completely absorbed into the following axioms:

$$O \sqsubseteq \downarrow x.T_x \qquad O \sqsubseteq \downarrow y.T_y \qquad T_y \sqsubseteq \forall a^-.T_1 \qquad O \sqsubseteq \downarrow z.T_z$$
$$T_z \sqsubseteq \forall v^-.T_2 \qquad (T_1 \sqcap T_2) \sqsubseteq T_3 \qquad T_3 \sqsubseteq \forall s^-.T_4 \qquad (T_3 \sqcap T_x) \sqsubseteq T_5$$
$$T_5 \sqsubseteq \forall r^-.T_6 \qquad (T_4 \sqcap T_6) \sqsubseteq T_7 \qquad T_7 \sqsubseteq gr(\exists c.\{x\}),$$

where $T_x, T_y, T_z, T_1, \ldots, T_7$ are fresh atomic concepts. Only $\exists c.\{x\}$ cannot be absorbed and has to be grounded on demand. To keep the example small, we have reused axioms for the absorption of the same concepts, whereas the algorithm of Section 3 would generate for each occurrence of $\neg\{y\}$ and $\neg\{z\}$ a separate binder concept.

### 4.2 Tableau Algorithm Extensions

We can now extend a standard tableau decision procedure to support (absorbed) nominal schema axioms. Note, we assume that GCIs are handled by two rules: the $\sqsubseteq_1$-rule

handles GCIs of the form $A \sqsubseteq C$ (i.e., a non-absorbable axiom $C \sqsubseteq D$ is handled as $\top \sqsubseteq \mathsf{nnf}(\neg C \sqcup D)$) and the $\sqsubseteq_2$-rule handles binary absorption axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$. These rules and the $\forall$-rule (for transitivity support also the $\forall^+$-rule) have to be adapted. The $\downarrow$ binders and $gr(\cdot)$ concepts are handled by new rules. In order to propagate variable bindings, we keep a set of mappings that records bindings for variables, for each concept in a node label.

**Definition 1 (Variable Mapping).** *A* variable mapping $\mu$ *is a (partial) function from variable names to individual names. The set of elements on which* $\mu$ *is defined is the* domain*, written* $\mathsf{dom}(\mu)$*, of* $\mu$*. We use* $\epsilon$ *for the empty variable mapping, i.e.,* $\mathsf{dom}(\epsilon) = \emptyset$*. We associate a concept* $C$ *in the label of a node* $v$ *with a set of variable mappings, denoted by* $\mathcal{B}(C, v)$*.*

When clear from the context, we simply write mapping instead of variable mapping.

Table 1 shows the adapted and new tableau rules and we describe the not so straightforward extensions in more detail below. The mappings have to be propagated by the tableau rules for the concepts and axioms that are used in the absorption. For example, if we apply the adapted $\sqsubseteq_1$-rule to an axiom of the form $A \sqsubseteq C$, we keep the mappings also for the concept $C$. Note that it is only necessary to extend those rules, which are related to concepts and axioms that are used in the absorption, because if the mappings are propagated to a $gr(\cdot)$ concept, the remaining, non-absorbed part of the axiom is grounded and thus corresponds to an ordinary concept.

Some major adjustments are necessary for the $\sqsubseteq_2$-rule that handles binary absorption axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$. First of all, we want to keep the default behaviour if there are no variable mappings associated to the concept facts for which the rule is applied, i.e., if $\mathcal{B}(A_1, v) \cup \mathcal{B}(A_2, v) = \emptyset$, then we add $C$ to the label of $v$. In contrast, if $\mathcal{B}(A_1, v) \neq \emptyset$ or $\mathcal{B}(A_2, v) \neq \emptyset$, we propagate the join of the mapping sets to the implied concept. In the case $\mathcal{B}(A_1, v) = \emptyset$ and $\mathcal{B}(A_2, v) \neq \emptyset$, we extend $\mathcal{B}(A_1, v)$ by the empty mapping $\epsilon$ so that the join of $\mathcal{B}(A_1, v)$ and $\mathcal{B}(A_2, v)$ results in $\mathcal{B}(A_2, v)$, which is then propagated to $C$. We proceed analogously for $\mathcal{B}(A_2, v) = \emptyset$ and $\mathcal{B}(A_1, v) \neq \emptyset$. In principle, the join combines variable mappings that map common variables to the same individual name and to point out that the empty sets of mappings are specially handled, we have extended the join operator $\bowtie$ with the superscript $\epsilon$.

**Definition 2 (Variable Mapping Join).** *Two variable mappings* $\mu_1$ *and* $\mu_2$ *are* compatible *if* $\mu_1(x) = \mu_2(x)$ *for all* $x \in \mathsf{dom}(\mu_1) \cap \mathsf{dom}(\mu_2)$*. For compatible mappings* $\mu_1$ *and* $\mu_2$*,* $\mu_1 \cup \mu_2$ *is defined as* $(\mu_1 \cup \mu_2)(x) = \mu_1(x)$ *if* $x \in \mathsf{dom}(\mu_1)$*, and* $(\mu_1 \cup \mu_2)(x) = \mu_2(x)$ *otherwise. Given two (possibly empty) sets of variable mappings* $M_1$*,* $M_2$*, let* $M_1^\epsilon = \{\epsilon\}$ *(*$M_2^\epsilon = \{\epsilon\}$*) if* $M_1 = \emptyset$ *(*$M_2 = \emptyset$*) and* $M_1^\epsilon = M_1$ *(*$M_2^\epsilon = M_2$*) otherwise. The* join $M_1 \bowtie^\epsilon M_2$ *is defined as* $\{\mu_1 \cup \mu_2 \mid \mu_1 \in M_1^\epsilon, \mu_2 \in M_2^\epsilon$ *and* $\mu_1$ *is compatible with* $\mu_2\} \setminus \{\epsilon\}$.

For a concept $gr(C)$ the $gr$-rule grounds $C$ based on the variable mappings associated to $gr(C)$. Since these mappings might not cover all nominal schema variables that occur in $C$, it is necessary to extend the mappings with every combination of named individuals for the remaining variables. This so-called completion ensures that only fully grounded concepts are added, which can then be handled as ordinary concepts in the completion graph. Therefore, it is also not necessary to further propagate mappings to such newly added concepts.

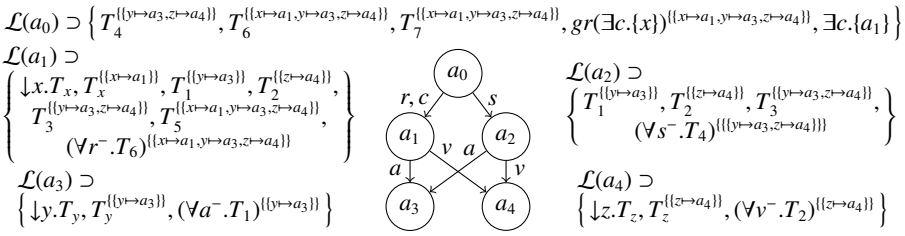


**Fig. 1.** Variable mapping propagation example

**Definition 3 (Grounding, Completion).** *For a concept $C$, $\mathsf{Vars}(C)$ is the set of nominal schema variables that syntactically occur in $C$. A concept $C$ is* grounded *if $\mathsf{Vars}(C) = \emptyset$. Let $\mu$ be a variable mapping. We write $C_{[\mu]}$ to denote the concept obtained by replacing each nominal schema $\{x\}$ that occurs in $C$ and $x \in \mathsf{dom}(\mu)$ with the nominal $\{\mu(x)\}$.*

*Given a set of variables $Y$ and a variable mapping set $M$ with $M^\epsilon$ as the extension by the empty mapping $\epsilon$ if $M = \emptyset$, the* completion $\mathsf{comp}_Y^{\mathcal{K}}(M)$ of $M$ w.r.t. $Y$ and a knowledge base $\mathcal{K}$ *containing the individuals $\mathsf{Inds}(\mathcal{K})$ is*

$$\mathsf{comp}_Y^{\mathcal{K}}(M) := \{\mu \cup \{x_1 \mapsto v_1, \ldots, x_n \mapsto v_n\} \mid \mu \in M^\epsilon, x_1, \ldots, x_n \in (Y \setminus \mathsf{dom}(\mu)), \\ v_1, \ldots, v_n \in \mathsf{Inds}(\mathcal{K})\}.$$

The unrestricted application of generating rules such as the $\exists$-rule can lead to the introduction of infinitely many new tableau nodes. To guarantee termination, one uses a cycle detection technique called *(pairwise) blocking* [6] that restricts the application of such rules. To apply blocking, we distinguish *blockable nodes* from *nominal nodes*, which have a nominal from the knowledge base in their label. A node $v$ with predecessor $v'$ is *blocked* by a node $w$ with predecessor $w'$, if $v, v', w, w'$ are all blockable and the labels of (i) $v$ and $w$ (ii) $v'$ and $w'$ and (iii) $\langle v', v\rangle$ and $\langle w', w\rangle$ coincide. We extend the standard blocking conditions to also require that the bindings for the concepts in the labels of these nodes coincide.

The completion graph in Figure 1 is obtained in the course of testing the consistency of a knowledge base containing the axioms of Example 1 and the assertions: $r(a_0, a_1)$, $s(a_0, a_2)$, $a(a_1, a_3)$, $v(a_1, a_4)$, $a(a_2, a_3)$, $v(a_2, a_4)$. Note, Figure 1 shows only those concepts and variable mappings (in superscripts) that are relevant for the grounding of new concepts in this example. However, since $O$ and thereby also the binder concepts are added to all ABox individuals, additional variable mappings are automatically created for every ABox individual. The joins of the mapping sets are created in the nodes $a_1$ and $a_2$ for the concepts $T_3$ and $T_5$ and finally in node $a_0$ for the concept $T_7$. Only the variable mapping $\{x \mapsto a_1, y \mapsto a_3, z \mapsto a_4\}$ is propagated to the grounding concept $gr(\exists c.\{x\}) \in \mathcal{L}(a_0)$ and, thus, by replacing the nominal schema $\{x\}$ with the nominal $\{a_1\}$, we have $\exists c.\{a_1\}$ as the only grounded concept. Hence, the individual $a_0$ is found to have a conflicting review assignment with the paper $a_1$.

Roughly speaking, it is possible to prove the correctness of our nominal schema absorption technique by a reduction between a completion graph for a TBox with nominal schemas and a standard completion graph for the upfront grounded TBox. Blocking

still guarantees termination since only a limited number of variable mappings are introduced.

**Theorem 2** *Let $\mathcal{T}$ denote an absorbed TBox (possibly with nominal schema axioms), then a tableau decision procedure (as described above) extended by the rules in Table 1 is a decision procedure for the satisfiability of $\mathcal{T}$.*

## 5 Implementation and Evaluation

We have implemented the techniques in the novel reasoning system Konclude that supports $\mathcal{SROIQV}$ by (i) upfront grounding and (ii) tableau extensions with different optimisations to handle the absorbed nominal schema axioms.

A detailed evaluation can be found in the technical report [15]. For brevity, we exemplarily show here some results for the University Ontology Benchmark (UOBM) [12] extended by DL-safe rules, which can straightforwardly be expressed as nominal schema axioms. The DL-safe rules allow for comparing Konclude to the DL reasoners HermiT 1.3.6[3] and Pellet 2.3.0 [14]. To the best of our knowledge, these are the only reasoning systems that support DL-safe rules for such expressive ontologies. The used ontology ($\text{UOBM}_1\backslash\text{D}$, data properties removed) has $\mathcal{SHOIN}$ expressivity and consists of 190,093 axioms, 69 classes, 36 properties, and 25,453 individuals. All experiments were performed on an Intel Core i7 940 quad core processor running at 2.93 GHz. The reasoners are restricted to use one core and all results are averaged over three runs. Exceeding the time (memory) limit of 24 hours (10 GB) is shown as *time* (*mem*).

Table 2 shows the rules and the number of matches for each rule in the consistency check. However, since reasoning with $\text{UOBM}_1\backslash\text{D}$ is non-deterministic, these numbers might vary between different executions and reasoners. Our system requires 1.03 s for preprocessing and 1.09 s for the consistency test for the ontology without rules. Table 3 then shows *the increases* in reasoning time for the ontology with nominal schema axioms. In parenthesis we show the additional preprocessing time for the upfront grounding, which is mostly spend on absorption, lexical normalisation, etc. Upfront grounding fails for R5 since although two variables can be eliminated (see safety condition in [11]) it requires 647,855,209 new axioms. We have also implemented an optimisation where we create a *representative* for a set of variable mappings. Only these representatives are then propagated and considered in the dependency directed backtracking, which saves memory. The direct propagation and the propagation of representatives are depicted (i) with and (ii) without the backward chaining (BC) optimisation, which is used to restrict the creation and propagation of variable mappings, i.e., variable mappings are only created if there is an opportunity to propagate them to a grounding concept. This is realised by additionally absorbing nominal schema axioms, where all nominal schemas are replaced by $O$, and by using the created atomic concept from this absorption to identify "interesting" individuals with possibly the grounding concept in the label. We then use a back propagation, whereby only binder concepts are activated that are in the scope of these "interesting" individuals. However, for example for rule R2, still nearly all variable mappings have to be created and propagated, and thus, the backward chaining only slightly improves the reasoning time.

[3] `http://www.hermit-reasoner.com`

**Table 2.** DL-safe Rules for UOBM-Benchmarks

| Name | DL-safe Rule | Matches |
|---|---|---|
| R1 | *isFirendOf*(?*x*, ?*y*), *like*(?*x*, ?*z*), *like*(?*y*, ?*z*) → *friendWithSameInterest*(?*x*, ?*y*) | 4,037 |
| R2 | *isFirendOf*(?*x*, ?*y*), *takesCourse*(?*x*, ?*z*), *takesCourse*(?*y*, ?*z*) → *friendWithSameCourse*(?*x*, ?*y*) | 82 |
| R3 | *takesCourse*(?*x*, ?*z*), *takesCourse*(?*y*, ?*z*), *hasSameHomeTownWith*(?*x*, ?*y*) → *classmateWithSameHomeTown*(?*x*, ?*y*) | 940 |
| R4 | *hasDoctoralDegreeFrom*(?*x*, ?*z*), *hasMasterDegreeFrom*(?*x*, ?*w*), *hasDoctoralDegreeFrom*(?*y*, ?*z*), *hasMasterDegreeFrom*(?*y*, ?*w*), *worksFor*(?*x*, ?*v*), *worksFor*(?*y*, ?*v*), → *workmateSameDegreeFrom*(?*x*, ?*y*) | 369 |
| R5 | *isAdvisedBy*(?*x*, ?*z*), *isAdvisedBy*(?*y*, ?*z*), *like*(?*x*, ?*w*), *like*(?*y*, ?*w*), *like*(?*z*, ?*w*) → *personWithSameAdviserAllSameInterest*(?*x*, ?*y*) | 286 |

**Table 3.** Comparison of the increases in reasoning time of the consistency tests in seconds

| Rule | upfront grounding | | direct propagation without BC | direct propagation with BC | representative propagation without BC | representative propagation with BC | HermiT 1.3.6 | Pellet 2.3.0 |
|---|---|---|---|---|---|---|---|---|
| R1 | (10.99) | mem | 9.12 | 7.10 | 5.06 | 3.38 | 31.46 | 6.33 |
| R2 | (10.92) | 4.05 | 3.33 | 2.33 | 2.13 | 2.11 | 4.79 | 7.4 |
| R3 | (13.33) | 3.55 | 1.98 | 0.62 | 2.20 | 0.76 | 1.67 | 142.25 |
| R4 | (16.44) | 0.30 | 1.08 | 0.09 | 1.06 | 0.07 | 1.42 | 122.85 |
| R5 | (time) | – | 1.87 | 0.50 | 1.80 | 0.43 | 28.41 | mem |

Table 3 further shows the reasoning time *increase* for HermiT and Pellet when a rule from Table 2 is added. Without rules Konclude requires 1.09 s, HermiT 23.24 s, and Pellet 2.22 s for a consistency test (ignoring loading and preprocessing time). With backwards chaining and the propagation of representatives, the reasoning times for Konclude are significantly faster than HermiT's or Pellet's. HermiT uses, however, significantly less memory than the other systems. This might be because HermiT does not support complex roles, such as *hasSameHomeTownWith* in R3, in the body of rules and its results might be incomplete.

## 6 Conclusions

We have addressed the problem of practical reasoning with nominal schemas through an extended absorption algorithm and with slight modifications of standard tableau calculi. Our approach "collects" the bindings for nominal schema axioms that have to be grounded and considered for a specific node in the completion graph. The presented techniques have been implemented and our empirical evaluation, which focusses on DL-safe rules, shows that our approach works well even compared to reasoners with dedicated rule support.

## Acknowledgements

The first author acknowledges the support of the doctoral scholarship under the Postgraduate Scholarships Act of the Land of Baden-Wuerttemberg (LGFG).

## References


1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Blackburn, P., Tzakova, M.: Hybridizing concept languages. Annals of Mathematics and Artificial Intelligence 24(1–4), 23–49 (1998)
3. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence 19(1), 17–37 (1982)
4. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
5. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B.N., Dean, M.: SWRL: A Semantic Web Rule Language. W3C Member Submission (21 May 2004), available at http://www.w3.org/Submission/SWRL/
6. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. J. of of Logic and Computation 9(3), 385–410 (1999)
7. Horrocks, I., Tobies, S.: Reasoning with axioms: Theory and practice. In: Proc. 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'00). pp. 285–296. Morgan Kaufmann (2000)
8. Hudek, A.K., Weddell, G.E.: Binary absorption in tableaux-based reasoning for description logics. In: Proc. 19th Int. Workshop on Description Logics (DL'06). vol. 189. CEUR (2006)
9. Kifer, M., Boley, H. (eds.): RIF Overview. W3C Working Group Note (22 June 2010), available at http://www.w3.org/TR/rif-overview/
10. Krisnadhi, A., Hitzler, P.: A tableau algorithm for description logics with nominal schema. In: Krötzsch, M., Straccia, U. (eds.) Proc. 6th Int. Conf. on Web Reasoning and Rule Systems (RR'12). LNCS, vol. 7497, pp. 234–237. Springer (2012)
11. Krötzsch, M., Maier, F., Krisnadhi, A., Hitzler, P.: A better uncle for OWL: nominal schemas for integrating rules and ontologies. In: Proc. 20th Int. Conf. on World Wide Web (WWW'11). pp. 645–654. ACM (2011)
12. Ma, L., Yang, Y., Qiu, Z., Xie, G., Pan, Y., Liu, S.: Towards a complete OWL ontology benchmark. In: Proc. 3rd European Semantic Web Conf. (ESWC'06). LNCS, vol. 4011, pp. 125–139. Springer (2006)
13. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-overview/
14. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. of Web Semantics 5(2), 51–53 (2007)
15. Steigmiller, A., Glimm, B., Liebig, T.: Nominal schema absorption. Tech. Rep. UIB-2013-02, Ulm University, Ulm, Germany (2013), available online at http://www.uni-ulm.de/fileadmin/website_uni_ulm/iui/Ulmer_Informatik_Berichte/2013/UIB-2013-02.pdf
16. Tsarkov, D., Horrocks, I.: Efficient reasoning with range and domain constraints. In: Proc. 17th Int. Workshop on Description Logics (DL'04). vol. 104. CEUR (2004)

# A Correctness Proofs

## A.1 Correctness of the Absorption Algorithm

In the following we prove the correctness of Theorem 1, i.e., the correctness of our modified absorption algorithm presented in Section 3. We first show that the complete absorption of a disjunct of an axiom is correct, i.e., it preserves the satisfiability (Lemma 1 and Lemma 2), and then we show that the correctness of a partially absorbed concept disjunct can be reduced to the complete absorption (Lemma 3).

**Lemma 1** *Let $\mathcal{T}$ denote a TBox, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an interpretation such that $\mathcal{I} \models \mathcal{T}$, $C$ a concept that is completely absorbable, $A$ the concept returned by* absorbJoined*($\{C\}$), and $\mathcal{T}'$ the extension of $\mathcal{T}$ with all the axioms created by* absorbJoined*($\{C\}$), then*

1. *for every extension $\mathcal{I}'$ of $\mathcal{I}$ such that $\mathcal{I}' \models \mathcal{T}'$, it holds that $\mathcal{I}' \models \mathcal{T}$,*
2. *for every extension $\mathcal{I}'$ of $\mathcal{I}$ such that $\mathcal{I}' \models \mathcal{T}'$, it holds for all $\delta \in \Delta^{\mathcal{I}'}$ that $\delta \in A^{\mathcal{I}'}$ if $\delta \notin C^{\mathcal{I}'}$, and*
3. *there exists an interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ such that $\mathcal{I}' \models \mathcal{T}'$ with $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ and $\delta \in A^{\mathcal{I}'}$ only if $\delta \notin C^{\mathcal{I}'}$.*

*Proof.* (*Claim 1*) Since $\mathcal{T}'$ is an extension of $\mathcal{T}$, it trivially follows that $\mathcal{I}' \models \mathcal{T}$.
(*Claim 2*) We first prove the simple cases where $C$ is completely absorbable and afterwards we show by induction that the lemma also holds for the complex cases.

- If $C$ is of the form $\neg A$, then absorbConcept($C$) directly returns $A$, which is then also returned by absorbJoined($\{C\}$). Thus, if $\delta \in \Delta^{\mathcal{I}'}$ and $\delta \notin C^{\mathcal{I}'}$, i.e., $\delta \notin (\neg A)^{\mathcal{I}'}$, then $\delta \in A^{\mathcal{I}'}$. Hence, the lemma holds if $C$ is of the form $\neg A$.
- If $C$ is of the form $\neg\{a\}$, then absorbConcept($C$) adds the axiom $\{a\} \sqsubseteq A$ to $\mathcal{T}'$ and returns $A$, which is then also returned by absorbJoined($\{C\}$). Thus, if $\delta \notin C^{\mathcal{I}'}$, i.e., $\delta \notin (\neg\{a\})^{\mathcal{I}'}$, then $\delta \in a^{\mathcal{I}'}$ and because, by assumption, $\mathcal{I}' \models \mathcal{T}'$, i.e., $\mathcal{I}' \models \{a\} \sqsubseteq A$, it follows that $\delta \in A^{\mathcal{I}'}$. Hence, the lemma holds if $C$ is of the form $\neg\{a\}$.

For the complex cases we assume that all nested disjunctions are replaced by a single disjunction with all disjuncts, i.e., $(C_1 \sqcup (C_2 \sqcup C_3))$ is replaced by $(C_1 \sqcup C_2 \sqcup C_3)$. Furthermore, we automatically decompose a disjunction into the set of disjuncts by calling absorbJoined. This simplification is also done by the algorithm with the collectDisjuncts function, which is always called before absorbJoined. Therefore, we can omit collectDisjuncts for calling absorbJoined, which improves the readability. Now, for a disjunct $C_j$, it follows that $C_j$ is not a disjunction itself and it also follows that absorbJoined($\{C_j\}$) only returns the atomic trigger concept that is returned by absorbConcept($C_j$).

Let $C_1, \ldots, C_n$ be completely absorbable concepts and $A_1, \ldots, A_n$ the atomic concepts returned by absorbJoined($\{C_1\}$), ..., absorbJoined($\{C_n\}$). By our induction hypothesis, the lemma holds for $A_1$ w.r.t. $C_1, \ldots, A_n$ w.r.t. $C_n$.

- If $C$ is now of the form $C_1 \sqcup \ldots \sqcup C_n$, then absorbJoined($\{C_1, \ldots, C_n\}$) collects the atomic concepts $A_1, \ldots, A_n$ by calling absorbConcept($C_j$) for each $C_j$, $1 \leqslant j \leqslant n$, and creates the binary absorption axioms $(A_1 \sqcap A_2) \sqsubseteq T_1, (T_1 \sqcap A_3) \sqsubseteq T_2, \ldots, (T_{n-2} \sqcap$

$A_n) \sqsubseteq A$. Thus, if $\delta \notin C^{\mathcal{I}'}$, i.e., $\delta \notin (C_1 \sqcup \ldots \sqcup C_n)^{\mathcal{I}'}$, then $\delta \in (\neg C_1 \sqcap \ldots \sqcap \neg C_n)^{\mathcal{I}'}$ and as a consequence $\delta \in (\neg C_j)^{\mathcal{I}'}$ for $1 \leqslant j \leqslant n$. Therefore, by the induction hypothesis we have $\delta \in A_j^{\mathcal{I}'}$ for all $1 \leqslant j \leqslant n$. Thus, $\delta \in A_1^{\mathcal{I}'}$ and $\delta \in A_2^{\mathcal{I}'}$ and since the interpretation $\mathcal{I}' \models \mathcal{T}'$ with $\{(A_1 \sqcap A_2) \sqsubseteq T_1, (T_1 \sqcap A_3) \sqsubseteq T_2, \ldots, (T_{n-2} \sqcap A_n) \sqsubseteq A\} \subseteq \mathcal{T}'$ it follows that $\delta \in T_1^{\mathcal{I}'}, \delta \in T_2^{\mathcal{I}'}, \ldots, \delta \in A^{\mathcal{I}'}$. Hence, the lemma holds by induction if $C$ is of the form $C_1 \sqcup \ldots \sqcup C_n$.

- If $C$ is of the form $C_1 \sqcap C_2$, then absorbJoined($\{C\}$) returns $A$, which is obtained by calling absorbConcept($C$), where additionally the axioms $A_1 \sqsubseteq A$ and $A_2 \sqsubseteq A$ are created. If $\delta \notin C^{\mathcal{I}'}$, i.e., $\delta \notin (C_1 \sqcap C_2)^{\mathcal{I}'}$, then $\delta \in (\neg C_1 \sqcup \neg C_2)^{\mathcal{I}'}$. There are now two cases: If $\delta \in (\neg C_1)^{\mathcal{I}'}$, then by the induction hypothesis we have $\delta \in A_1^{\mathcal{I}'}$ and due to the axiom $A_1 \sqsubseteq A$ we have $\delta \in A^{\mathcal{I}'}$. For the other case we have $\delta \in (\neg C_2)^{\mathcal{I}'}$ and by the induction hypothesis $\delta \in A_2^{\mathcal{I}'}$ and due to the axiom $A_2 \sqsubseteq A$ we also have $\delta \in A^{\mathcal{I}'}$. Hence, the lemma holds by induction if $C$ is of the form $C_1 \sqcap C_2$.
- If $C$ is of the form $\forall r.C_1$, then absorbConcept($C$) creates $A_1 \sqsubseteq \forall r^-.A$ and $A$ is returned by absorbJoined($\{C\}$). Thus, if $\delta \notin C^{\mathcal{I}'}$, i.e., $\delta \notin (\forall r.C_1)^{\mathcal{I}'}$, then $\delta \in (\exists r.\neg C_1)^{\mathcal{I}'}$. It follows that there exists $\gamma \in \Delta^{\mathcal{I}'}$, $(\delta, \gamma) \in r^{\mathcal{I}'}$ with $\gamma \in (\neg C_1)^{\mathcal{I}'}$ and by the induction hypothesis we have $\gamma \in A_1^{\mathcal{I}'}$. As a consequence of the axiom $A_1 \sqsubseteq \forall r^-.A$ we also have $\delta \in A^{\mathcal{I}'}$. Hence, the lemma holds by induction if $C$ is of the form $\forall r.C_1$.

(*Claim 3*) We construct the interpretation $\mathcal{I}'$ from $\mathcal{I}$ such that $\delta \in A^{\mathcal{I}'}$ only if $\delta \notin C^{\mathcal{I}'}$. Therefore, let $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ be an interpretation with $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ and $\cdot^{\mathcal{I}'}$ reduced from $\cdot^{\mathcal{I}}$ such that only the atomic concepts, atomic roles, and individuals occurring in $\mathcal{T}$ are interpreted. Obviously, it still holds that $\mathcal{I}' \models \mathcal{T}$ since the interpretation of all axioms in $\mathcal{T}$ coincides with $\mathcal{I}$. We now define the interpretation of the fresh atomic concepts $A_1, \ldots, A_m$ introduced for the absorption of $C$ in $\mathcal{I}'$. Note that we treat absorption axioms of the form $A' \sqsubseteq \forall r.A_i$ in their equivalent form $\exists r^-.A' \sqsubseteq A_i$.

Now, for $1 \leqslant i \leqslant m$ and for each axiom $H \sqsubseteq A_i$ generated by the absorption, we exhaustively add $\delta \in \Delta^{\mathcal{I}'}$ to $A_i^{\mathcal{I}'}$ if (i) $H = A'$ and $\delta \in A'^{\mathcal{I}'}$, (ii) $H = \{a\}$ and $\delta \in \{a\}^{\mathcal{I}'}$, (iii) $H = (A' \sqcap A'')$ and $\delta \in A'^{\mathcal{I}'} \cap A''^{\mathcal{I}'}$, or (iv) $H = \exists r^-.A'$ and $\delta \in (\exists r^-.A')^{\mathcal{I}'}$, i.e., $\delta$ has some $r$-neighbour $\gamma$ such that $(\gamma, \delta) \in r^{\mathcal{I}'}$ and $\gamma \in A'^{\mathcal{I}'}$. We have $\delta \in A_i^{\mathcal{I}'}$ only if $\delta$ satisfies the left-hand side of an axiom $A' \sqsubseteq A_i$, $\{a\} \sqsubseteq A_i$, or $(A' \sqcap A'') \sqsubseteq A_i$, or $\exists r^-.A' \sqsubseteq A_i$. Consequently, it follows that $\mathcal{I}' \models \mathcal{T}'$. Furthermore, $\delta \notin A^{\mathcal{I}'}$ if $\delta \in C^{\mathcal{I}'}$, because of the following cases:

- If $C$ is of the form $\neg A$ and $\delta \in C^{\mathcal{I}'}$, i.e., $\delta \in (\neg A)^{\mathcal{I}'}$, then $\delta \notin A^{\mathcal{I}'}$.
- If $C$ is of the form $\neg\{a\}$ for which the absorption has generated $\{a\} \sqsubseteq A$ and if $\delta \in C^{\mathcal{I}'}$, i.e., $\delta \in (\neg\{a\})^{\mathcal{I}'}$, then $\delta \notin \{a\}^{\mathcal{I}'}$ and then $\delta \notin A^{\mathcal{I}'}$, because the left-hand side of $\{a\} \sqsubseteq A$ is not satisfied and there is also no other axiom that implies $A$, because $A$ is freshly used for $\{a\} \sqsubseteq A$.

For the remaining cases, we again assume that the lemma holds for $A_1$ w.r.t. $C_1, \ldots, A_n$ w.r.t. $C_n$, where $A_1, \ldots, A_n$ are the atomic trigger concepts for absorbing the completely absorbable concepts $C_1, \ldots, C_n$. Therefore, it follows by induction that $\delta \notin A^{\mathcal{I}'}$ if $\delta \in C^{\mathcal{I}'}$, because:

- If $C$ is of the form $C_1 \sqcup \ldots \sqcup C_n$ and $\delta \in C^{\mathcal{I}'}$, i.e., $\delta \in (C_1 \sqcup \ldots \sqcup C_n)^{\mathcal{I}'}$, then there exists a $C_j$, $1 \leqslant j \leqslant n$ with $\delta \in C_j^{\mathcal{I}'}$. By the induction hypothesis it follows that $\delta \notin A_j^{\mathcal{I}'}$

and by the binary axiom chain $(A_1 \sqcap A_2) \sqsubseteq T_1, (T_1 \sqcap A_3) \sqsubseteq T_2, \ldots, (T_{j-2} \sqcap A_j) \sqsubseteq T_{j-1}, \ldots, (T_{n-2} \sqcap A_n) \sqsubseteq A$, which is generated for absorbing $C_1 \sqcup \ldots \sqcup C_n$, we have $\delta \notin A^{\mathcal{I}'}$, because the left-hand side of the axiom $(T_{j-2} \sqcap A_j) \sqsubseteq T_{j-1}$ cannot be satisfied.

- If $C$ is of the form $C_1 \sqcap C_2$ and $\delta \in C^{\mathcal{I}'}$, i.e., $\delta \in (C_1 \sqcap C_2)^{\mathcal{I}'}$, then $\delta \in C_1^{\mathcal{I}'}$ and $\delta \in C_2^{\mathcal{I}'}$. By the induction hypothesis we have $\delta \notin A_1^{\mathcal{I}'}$ and $\delta \notin A_2^{\mathcal{I}'}$. The left-hand side of the axioms $A_1 \sqsubseteq A$ and $A_2 \sqsubseteq A$ is not satisfied and the absorptions does not generate other axioms that imply $A$. Thus, $\delta$ is not added to $A^{\mathcal{I}'}$.
- If $C$ is of the form $\forall r.C_1$ and $\delta \in C^{\mathcal{I}'}$, i.e., $\delta \in (\forall r.C_1)^{\mathcal{I}'}$, then for all $\gamma \in \Delta^{\mathcal{I}'}$ with $(\delta, \gamma) \in r^{\mathcal{I}'}$ we also have $\gamma \in C_1^{\mathcal{I}'}$. By the induction hypothesis it follows that $\gamma \notin A_1^{\mathcal{I}}$ and since the left-hand side of the generated axiom $\exists r^-.A_1 \sqsubseteq A_i$ is not satisfied, and there are not any other axioms that imply $A$, we do not add $\delta$ to $A^{\mathcal{I}'}$ and, thus, $\delta \notin A^{\mathcal{I}'}$. □

We can now use Lemma 1 to show the correctness of the absorption for the case of a completely absorbable concept $C$ in an axiom $C \sqsubseteq D$.

**Lemma 2** *For $\mathcal{T}$ a TBox and $C \sqcup D$ a disjunction, where $C$ is completely absorbable and $D$ is neither completely nor partially absorbable, let $\mathcal{T}_1$ denote the TBox with $\mathcal{T}_1 = \mathcal{T} \cup \{\top \sqsubseteq C \sqcup D\}$ and $\mathcal{T}_2$ denote the TBox with $\mathcal{T}_2 = \mathcal{T} \cup \{A \sqsubseteq D\} \cup X$, where $X$ are the axioms created by $A \leftarrow$ absorbJoined$(\{C\})$. Then, a concept $C'$ is satisfiable with respect to $\mathcal{T}_1$ iff it is satisfiable with respect to $\mathcal{T}_2$.*

*Proof.* If direction: For $\mathcal{I}_2$ an interpretation with $C'^{\mathcal{I}_2} \neq \emptyset$ and $\mathcal{I}_2 \models \mathcal{T}_2$, we show that $\mathcal{I}_2 \models \mathcal{T}_1$. Because of the axiom $A \sqsubseteq D \in \mathcal{T}_2$ for each $\delta \in \Delta^{\mathcal{I}_2}$ it holds that either $\delta \notin A^{\mathcal{I}_2}$ (and thus $\delta \in C^{\mathcal{I}_2}$ by Lemma 1) or $\delta \in D^{\mathcal{I}_2}$. Thus, the axiom $\top \sqsubseteq (C \sqcup D) \in \mathcal{T}_1$ is satisfied for every $\delta \in \Delta^{\mathcal{I}_2}$ and, therefore, $\mathcal{I}_2 \models \mathcal{T}_1$.

Only if direction: For $\mathcal{I}_1$ an interpretation with $C'^{\mathcal{I}_1} \neq \emptyset$ and $\mathcal{I}_1 \models \mathcal{T}_1$, we construct an interpretation $\mathcal{I}'_1$ with $C'^{\mathcal{I}'_1} \neq \emptyset$ and $\mathcal{I}'_1 \models \mathcal{T}_2$. Since $\mathcal{I}_1 \models \mathcal{T}_1$ and $\mathcal{T}_1$ is an extension of $\mathcal{T}$, it follows that $\mathcal{I}_1 \models \mathcal{T}$. Because of Lemma 1, there exists an interpretation $\mathcal{I}'_1$ that can be constructed from $\mathcal{I}_1$ for which it holds that $\mathcal{I}'_1 \models \mathcal{T} \cup X$ and for all $\delta \in \Delta^{\mathcal{I}'_1}$ that $\delta \in A^{\mathcal{I}'_1}$ only if $\delta \notin C^{\mathcal{I}'_1}$. Thus, it also follows that $\mathcal{I}'_1 \models A \sqsubseteq D$, because $\Delta^{\mathcal{I}'_1} = \Delta^{\mathcal{I}_1}$ and for all $\delta \in \Delta^{\mathcal{I}'_1}$ it holds that either $\delta \in C^{\mathcal{I}'_1}$ and thus $\delta \notin A^{\mathcal{I}'_1}$ or $\delta \in D^{\mathcal{I}'_1}$. Thus, if $C'^{\mathcal{I}_1} \neq \emptyset$, then $C'^{\mathcal{I}'_1} \neq \emptyset$. □

In order to show the correctness of the partial absorption of a disjunction $C \sqcup D$, where $C$ is partially absorbable and $D$ is neither completely nor partially absorbable, we reduce the problem to the complete absorption of $C' \sqcup C \sqcup D$, where for $C'$ it holds that $C'$ is completely absorbable and $C' \sqsubseteq C$. We show that the partial absorption of $C$ is equivalent to the complete absorption of the concept $C'$. Therefore, the partial absorption of $C \sqcup D$ corresponds to the complete absorption of $C' \sqcup C \sqcup D$, which is obviously equisatisfiable to $C \sqcup D$ since $C$ subsumes $C'$.

**Lemma 3** *Let $C$ be a partially absorbable concept, then* absorbJoined$(\{C\})$ *generates the absorption of a concept $C'$ for which it holds that $C' \sqsubseteq C$ and $C'$ is completely absorbable.*

*Proof.* If $C$ is already completely absorbable, then the lemma trivially holds since in this case $C'$ is $C$. Thus, we show in the following for all cases where $C$ is partially absorbable but not completely absorbable that $\mathsf{absorbJoined}(\{C\})$ generates the absorption of a more specific concept $C'$ for which it holds $C' \sqsubseteq C$ and $C'$ is completely absorbable.

- If $C$ is of the form $\forall r.D'$ and $D'$ ($\mathsf{nnf}(\neg D')$) is neither completely nor partially absorbable, then $\mathsf{absorbConcept}(C)$ creates $\top \sqsubseteq \forall r^{-}.A$, which corresponds to the complete absorption of $\forall r.\neg\top$ for which it holds that $\forall r.\neg\top \sqsubseteq \forall r.D'$.

To prove the complex cases by induction, we assume that the concepts $D_1, \ldots, D_m$ are partially absorbable and the lemma holds for $D_1, \ldots, D_m$, i.e., the absorption completely absorbs the concepts $D'_1, \ldots, D'_m$, for which it holds that $D'_1 \sqsubseteq D_1, \ldots, D'_m \sqsubseteq D_m$, and let $A_1, \ldots, A_m$ be the atomic trigger concepts that are achieved for absorbing $D'_1, \ldots, D'_m$.

- If $C$ is of the form $\forall r.D_1$ and $D_1$ is partially absorbable, then $\mathsf{absorbConcept}(C)$ creates $A_1 \sqsubseteq \forall r^{-}.A$, where $A_1$ is the atomic trigger concept that is returned by $\mathsf{absorbConcept}(D_1)$ for completely absorbing $D'_1$. The absorption of $C$ corresponds to the complete absorption of $\forall r.D'_1$ and, by the induction hypothesis, we have $D'_1 \sqsubseteq D_1$. Thus, it also holds that $\forall r.D'_1 \sqsubseteq \forall r.D_1$.
- If $C$ is of the form $D_1 \sqcup \ldots \sqcup D_m \sqcup C_1 \sqcup \ldots \sqcup C_n$ with $D_1, \ldots, D_m$ partially absorbable and $C_1, \ldots, C_m$ neither partially nor completely absorbable, then the absorption creates the binary axiom chain $(A_1 \sqcap A_2) \sqsubseteq T_1, (T_1 \sqcap A_3) \sqsubseteq T_2, \ldots, (T_{m-2} \sqcap A_m) \sqsubseteq A$, which corresponds to the complete absorption of $D'_1 \sqcup \ldots \sqcup D'_m$, where $A_1, \ldots, A_m$ are again the atomic trigger concepts for absorbing $D'_1, \ldots, D'_m$. Because of the induction hypothesis it holds that $D'_1 \sqcup \ldots \sqcup D'_m \sqsubseteq D_1 \sqcup \ldots \sqcup D_m \sqcup C_1 \sqcup \ldots \sqcup C_n$.
- If $C$ is of the form $D_1 \sqcap D_2$ with $D_1, D_2$ partially absorbable, then the absorption creates the axioms $A_1 \sqsubseteq A$ and $A_2 \sqsubseteq A$, which corresponds to the complete absorption of $D'_1 \sqcap D'_2$, where $A_1$ and $A_2$ are the atomic trigger concepts for absorbing $D'_1$ and $D'_2$. Because of the induction hypothesis it holds that $D'_1 \sqcap D'_2 \sqsubseteq D_1 \sqcap D_2$. □

### A.2 Correctness of Nominal Schema Absorption

In the following we prove the correctness of Theorem 2, i.e., our nominal schema absorption technique presented in Section 4. For this, we roughly proceed as follows: Given a nominal schema axiom $C \sqsubseteq D$ and an absorbed TBox $\mathcal{T}$, then for $\mathcal{T}_{ns}$ and $\mathcal{T}_{ug}$ as the TBoxes obtained from absorbing $\mathcal{T} \cup \{C \sqsubseteq D\}$ and $\mathcal{T} \cup \{U_1, \ldots, U_h\}$, respectively, where $U_1, \ldots, U_h$ are the upfront grounded axioms of $C \sqsubseteq D$, we show that a fully expanded and clash free completion graph $G_{ns}$ for $\mathcal{T}_{ns}$ can be converted to a fully expanded and clash free completion graph $G_{ug}$ for $\mathcal{T}_{ug}$. Furthermore, we show that our extended tableau algorithm constructs a complete and clash free completion graph $G_{ns}$ for $\mathcal{T}_{ns}$ if there exists a fully expanded and clash free completion graph $G_{ug}$ for $\mathcal{T}_{ug}$ that is constructed by a standard tableau algorithm.

Please note that we only work with TBoxes instead of knowledge bases. This assumption is w.l.o.g. since in the presence of nominals ABoxes can be internalised (e.g., $C(a)$ is equivalent to the GCI $\{a\} \sqsubseteq A$, $r(a_1, a_2)$ to $\{a\} \sqsubseteq \exists r.\{b\}$, etc.). We assume, therefore, that a completion $\mathsf{comp}^{\mathcal{T}}_{Y}(M)$ is analogously defined to the completion $\mathsf{comp}^{\mathcal{K}}_{Y}(M)$ with $\mathcal{K} = (\mathcal{T}, \emptyset)$.

To simplify the conversion between a completion graph for $\mathcal{T}_{ns}$ and a standard completion graph for $\mathcal{T}_{ug}$, we ensure that all concept facts can directly be converted into concept facts for the other completion graph. Therefore, we make the following simplifying assumptions: We assume that the absorption of nominals of the form $\neg\{a\}$ generates $\{a\} \sqsubseteq \top \sqcap A$ instead of $\{a\} \sqsubseteq A$ (cf. Algorithm 4, line 14), which is obviously logically equivalent. As a result, binder concepts such as $\downarrow x.A$ can be directly converted to concepts of the form $\top \sqcap A$. We also assume that the absorption of the upfront grounded axiom $C_{[\mu]} \sqsubseteq D_{[\mu]}$, by the variable mapping $\mu$, creates a new special grounding concept $gr_\mu(D)$ to add the remaining, non-absorbable part of the axiom instead of directly implying $D_{[\mu]}$. This new concept construct retains the mapping $\mu$ and corresponds to the grounding concept $gr(D)$ that is created for the absorption of the nominal schema axiom $C \sqsubseteq D$.

Before introducing the actual conversion, we first define the notion of concept and axiom set closure:

**Definition 4 (Closure).** *The* closure $\mathsf{clos}(C')$ *of a concept $C'$ is a set of concepts that is closed under sub-concepts of $C'$ and also contains $C'$. Additionally,* $\mathsf{fclos}(Z)$ *is the extension to a set of axioms $Z$:*

$$\mathsf{fclos}(Z) := \bigcup_{C' \sqsubseteq D' \in Z} \mathsf{clos}(\neg C' \sqcup D').$$

*For a TBox $\mathcal{T}$ and an axiom $C' \sqsubseteq D'$ with* $\mathsf{nnf}(\neg C')$ *completely and $D'$ not completely absorbable, the* absorption closure $\mathsf{aclos}_{\mathcal{T}}(C' \sqsubseteq D')$ for $\mathcal{T}$ and $C' \sqsubseteq D'$ *contains the new concepts introduced by the absorption of $C' \sqsubseteq D'$ and is defined as:*

$$\mathsf{aclos}_{\mathcal{T}}(C' \sqsubseteq D') := \mathsf{fclos}(X'_1, \ldots, X'_n) \setminus (\mathsf{fclos}(\mathcal{T}) \cup \mathsf{clos}(D')),$$

*where $X'_1, \ldots, X'_n$ are the axiom introduced by the absorption of $C' \sqsubseteq D'$.*

Note that the concepts in the absorption closure are those that are relevant for the conversion between completion graphs since these are the concepts with variable mappings.

Now, the actual conversion of concepts and axioms obtained from the absorption is defined as follows:

**Definition 5 (Conversion).** *Let $C \sqsubseteq D$ be a nominal schema axiom where* $\mathsf{nnf}(\neg C)$ *is completely and $D$ not completely absorbable, and let $\mu$ be a mapping with* $\mathsf{dom}(\mu) = \mathsf{Vars}(\neg C \sqcup D)$ *. Furthermore, let $\mathcal{T}$ be an absorbed TBox, $\mathcal{T}_{ns}$ and $\mathcal{T}_{ug}$ TBoxes obtained by absorbing $\mathcal{T} \cup \{C \sqsubseteq D\}$ and $\mathcal{T} \cup \{U_1, \ldots, U_h\}$, respectively, where $U_1, \ldots, U_h$ are the axioms obtained by the upfront grounding of $C \sqsubseteq D$. We denote the axioms (in creation order) and fresh atomic concepts obtained by absorbing* $\mathsf{nnf}(\neg C \sqcup D)$ *with $X_1, \ldots, X_n$ and $A_1, \ldots, A_g$, respectively. Similarly, we use $X^\mu_1, \ldots, X^\mu_n$ and $A^\mu_1, \ldots, A^\mu_g$ for the case of absorbing* $\mathsf{nnf}((\neg C \sqcup D)_{[\mu]})$.

*For the concept $C'$, we inductively define the* concept conversion $\mathsf{conv}_\mu(C')$ of $C'$ w.r.t. $\mathcal{T}$, $C \sqsubseteq D$ and $\mu$ *as*

$$\mathsf{conv}_\mu(C') = \begin{cases} C' & \text{if } C' \notin \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D) \\ (\top \sqcap \mathsf{conv}_\mu(C'')) & \text{if } C' = \downarrow x.C'' \\ gr_\mu(D) & \text{if } C' = gr(D) \\ C'_{[A_1/A^\mu_1, \ldots, A_g/A^\mu_g]} & \text{otherwise,} \end{cases}$$

*where* $C'_{[A_1/A_1^\mu,\ldots,A_g/A_g^\mu]}$ *denotes the syntactic replacement of each occurrence of* $A_i$ *in* $C'$ *with* $A_i^\mu$*, for* $1 \leq i \leq g$*. The extension to axioms* $\mathsf{fconv}_\mu(X)$ *is defined as:*

$$\mathsf{fconv}_\mu(X) = \begin{cases} \{\mu(x)\} \sqsubseteq \top \sqcap \mathsf{conv}_\mu(D') & \text{if } X = O \sqsubseteq \downarrow x.D' \\ \mathsf{conv}_\mu(C') \sqsubseteq \mathsf{conv}_\mu(D') & \text{otherwise.} \end{cases}$$

In the remainder of the section, we use $C \sqsubseteq D, \mu, \mathcal{T}, \mathcal{T}_{ns}, \mathcal{T}_{ug}, A_1, \ldots, A_g, A_1^\mu, \ldots, A_g^\mu$, $X_1, \ldots, X_n$, and $X_1^\mu, \ldots, X_n^\mu$ as in the above definition.

Note that the restrictions on $C \sqsubseteq D$ are w.l.o.g. since any nominal schema axiom can be transformed into the desired form in an equivalence preserving manner. If $\mathsf{nnf}(\neg C)$ is only partially absorbable, then a completely absorbable concept $\mathsf{nnf}(\neg C')$ can be extracted from $C$ (cf. Lemma 3), which can be used to obtain an axiom $C' \sqsubseteq D'$, where it holds that $C' \sqsubseteq C$, $C'$ is completely absorbable and $D' = \mathsf{nnf}(\neg C') \sqcup D$ is not completely absorbable. Also note that $\neg\top$ and $\bot$ can always be used to extend a disjunction that corresponds to an axiom in order to obtain a completely absorbable and not completely absorbable disjunct w.r.t. our absorption algorithm.

We can now show that we can convert the axioms obtained by absorbing $\mathsf{nnf}(\neg C \sqcup D)$ from the nominal schema axiom $C \sqsubseteq D$ into the axioms that are obtained by absorbing the grounded version $\mathsf{nnf}((\neg C \sqcup D)_{[\mu]})$, which is the first step in the conversion of a completion graph with nominal schema concepts to a standard completion graph:

**Lemma 4** *Let* $\mathcal{T}$ *be an absorbed TBox,* $C \sqsubseteq D$ *a nominal schema axiom,* $U_1, \ldots, U_h$ *the upfront grounding,* $\mu$ *a mapping,* $\mathcal{T}_{ns}$ *and* $\mathcal{T}_{ug}$ *TBoxes, and* $X_1, \ldots, X_n$ *and* $X_1^\mu, \ldots, X_n^\mu$ *axioms as in Definition 5. The set* $\{\mathsf{fconv}_\mu(X_1), \ldots, \mathsf{fconv}_\mu(X_n)\}$ *is identical to the set* $\{X_1^\mu, \ldots, X_n^\mu\}$*.*

*Proof.* Let $A_1, \ldots, A_g$ and $A_1^\mu, \ldots, A_g^\mu$ be the fresh atomic concepts introduced by the absorption of $\mathsf{nnf}(\neg C \sqcup D)$ and $\mathsf{nnf}((\neg C \sqcup D)_{[\mu]})$, respectively. Since the concepts $\mathsf{nnf}(\neg C \sqcup D)$ and $\mathsf{nnf}((\neg C \sqcup D)_{[\mu]})$ only differ in the nominal schemas that are replaced by nominals, the absorption of $\mathsf{nnf}(\neg C \sqcup D)$ and $\mathsf{nnf}((\neg C \sqcup D)_{[\mu]})$ is identical expect for axioms of the form $O \sqsubseteq \downarrow x.A_i$ and $A_g \sqsubseteq gr(D)$ in $\mathcal{T}_{ns}$, which correspond to axioms of the form $\{a\} \sqsubseteq (\top \sqcap A_i^\mu)$ and $A_g^\mu \sqsubseteq gr_\mu(D)$ in $\mathcal{T}_{ug}$. Hence, by Definition 5, the claim holds. □

For the conversion, we use the implicitly associated sets of variable mappings, which are defined as follows:

**Definition 6 (Implicitly Associated Mappings).** *The implicitly associated set of variable mappings* $\mathsf{mapp}^G(C'(v))$ *for a concept fact* $C'(v)$ *and* $C'$ *in the absorption closure w.r.t. a completion graph* $G = (V, E, \mathcal{L}, \mathcal{B})$ *is defined as:*

$$\mathsf{mapp}^G(C'(v)) = \begin{cases} \{\{x \mapsto v\}\} & \text{if } C' = \downarrow x.D' \\ \mathcal{B}(C', v) & \text{if } \mathcal{B}(C', v) \neq \emptyset \\ \{\epsilon\} & \text{otherwise.} \end{cases}$$

Now, let $G_{ns}$ be a completion graph showing the satisfiability of the TBox $\mathcal{T}_{ns}$. We can replace each concept fact $C'(v)$ with the implicitly associated variable mappings $M$ and $C' \in \mathsf{aclos}_\mathcal{T}(C \sqsubseteq D)$, by the concept facts $(\mathsf{conv}_{\mu_1}(C'))(v), \ldots, (\mathsf{conv}_{\mu_k}(C'))(v)$,

where $\mu_1, \ldots, \mu_k$ are the mappings obtained from the completion $\mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(M)$ of $M$. As a result, we obtain a fully expanded completion graph $G_{ug}$ that shows the satisfiability of the upfront grounded TBox $\mathcal{T}_{ug}$.

**Lemma 5 (Soundness)** *Let $\mathcal{T}$ be an absorbed TBox, $C \sqsubseteq D$ a nominal schema axiom, $U_1, \ldots, U_h$ the upfront grounding for $C \sqsubseteq D$, and $\mathcal{T}_{ns}$ and $\mathcal{T}_{ug}$ TBoxes as in Definition 5. If there is a fully expanded and clash free completion graph for $\mathcal{T}_{ns}$, then there is a fully expanded and clash free completion graph for $\mathcal{T}_{ug}$.*

*Proof.* Let $G_{ns} = (V_{ns}, E_{ns}, \mathcal{L}_{ns}, \dot{\neq}_{ns}, \mathcal{B}_{ns})$ be a fully expanded and clash free completion graph for $\mathcal{T}_{ns}$. We convert $G_{ns}$ into a fully expanded and clash free completion graph $G_{ug}$ by replacing every concept fact $C'(v)$, $C' \in \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$, $v \in V_{ns}$, with the implicitly associated variable mappings $M = \mathsf{mapp}^{G_{ns}}(C'(v))$, by the concept facts $(\mathsf{conv}_{\mu_1}(C'))(v), \ldots, (\mathsf{conv}_{\mu_k}(C'))(v)$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(M)$. Furthermore, let $\lambda_2, \ldots, \lambda_\ell$ be all possible variable mappings for $\mathsf{Vars}(\neg C \sqcup D)$ w.r.t. $\mathcal{T}$, i.e., $\{\lambda_2, \ldots, \lambda_\ell\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\epsilon\})$.

In the following we show that none of the standard tableau rules for the concepts and axioms used in the absorption are applicable to $G_{ug}$. Please note that the extended tableau rules (Table 1) coincide with the standard tableau rules [4] if no variable mappings are associated to the concept facts. Also note that the concept facts and axioms, which are not related to the absorption, are not affected by the conversion. Thus, the corresponding rules are not applicable for these concepts and axioms. Furthermore, since identical node labels are converted in the same way, blocking is not affected, i.e., if a node is blocked before the conversion, then it is also blocked after the conversion.

- We firstly consider the application of the $\forall$-rule, which is not applicable for $G_{ug}$, because $C' = \forall r.D'(v)$ is converted to $(\mathsf{conv}_{\mu_1}(\forall r.D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(\forall r.D'))(v)$ and for each $r$-neighbour node $w$ of $v$ the concept fact $D'(w)$ is either also not associated with variable mappings (which is ensured by the absorption algorithm by creating separate axioms with fresh atomic concepts for the absorption of concepts that do not contain nominal schemas) or is at least also associated with the same variable mappings (otherwise the $\forall$-rule would be applicable for $G_{ns}$) and thus $D'(w)$ is at least also converted to $(\mathsf{conv}_{\mu_1}(D'))(w), \ldots, (\mathsf{conv}_{\mu_k}(D'))(w)$.
- We now consider the application of the $\sqsubseteq_1$-rule. The absorption creates axioms of the form $H \sqsubseteq D'$ with $D' \in \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$ and $H = \{a\}$ or $H = A$. If $D' \neq \downarrow x.D''$ (the replacement axioms for $O \sqsubseteq \downarrow x.D''$ are considered together with the $\downarrow$-concepts), $H \notin \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$ and $H = A$ or $H = \{a\}$, then we would have the axioms $H \sqsubseteq \mathsf{conv}_{\lambda_2}(D'), \ldots, H \sqsubseteq \mathsf{conv}_{\lambda_\ell}(D')$ in $\mathcal{T}_{ug}$ and the $\sqsubseteq_1$-rule is not applicable, because, for every node $v$ in $G_{ns}$ with the concept fact $H(v)$, $D'(v)$ is also present and $\mathcal{B}_{ns}(D', v) = \emptyset$. Thus, $D'(v)$ is replaced by $(\mathsf{conv}_{\lambda_2}(D'))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(v)$. If $A \in \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$, then we would have the axioms $\mathsf{conv}_{\lambda_2}(A) \sqsubseteq \mathsf{conv}_{\lambda_2}(D'), \ldots, \mathsf{conv}_{\lambda_\ell}(A) \sqsubseteq \mathsf{conv}_{\lambda_\ell}(D')$ and the $\sqsubseteq_1$-rule is not applicable, because for every node $v$ in $G_{ns}$ with the concept fact $A(v)$ and the associated variable mappings $\mu_1, \ldots, \mu_k$, $A(v)$ would be replaced by $(\mathsf{conv}_{\mu_1}(A))(v), \ldots, (\mathsf{conv}_{\mu_k}(A))(v)$, and $D'$ is either also not associated with variable mappings (which is ensured by the absorption algorithm) or is at least also associated with the variable mappings

$\mu_1, \ldots, \mu_k$ (otherwise $G_{ns}$ would not be fully expanded), and is at least also replaced by $(\mathsf{conv}_{\mu_1}(D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(D'))(v)$. Thus, the $\sqsubseteq_1$-rule is not applicable for $G_{ug}$.

- Next, we consider the application of the $\sqsubseteq_2$-rule for an axiom $(A_1 \sqcap A_2) \sqsubseteq D'$. There are three cases:
  1. If $\mathcal{B}_{ns}(A_1, v) = \emptyset$ and $\mathcal{B}_{ns}(A_2, v) = \emptyset$, then $\mathcal{B}_{ns}(D, v) = \emptyset$ and every concept fact $D'(v)$ is replaced by $(\mathsf{conv}_{\lambda_2}(D'))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(v)$ and thus the rule is not applicable for $(A_1 \sqcap A_2) \sqsubseteq \mathsf{conv}_{\lambda_2}(D'), \ldots, (A_1 \sqcap A_2) \sqsubseteq \mathsf{conv}_{\lambda_\ell}(D')$.
  2. If $\mathcal{B}_{ns}(A_1, v) \neq \emptyset$ ($\mathcal{B}_{ns}(A_2, v) \neq \emptyset$), then the $\sqsubseteq_2$-rule is analogously to the $\sqsubseteq_1$-rule not applicable, because either there is no variable mapping that is associated to $A_2(v)$ ($A_1(v)$) and, as a consequence, there is also no variable mapping associated to $D'(v)$ (which is ensured by the absorption algorithm), or every variable mapping that is associated to $A_2(v)$ ($A_1(v)$) is also associated to $D'(v)$ if $A_1$ ($A_2$) is also in the label of $v$. Thus, the $\sqsubseteq_2$-rule cannot add a $\mathsf{conv}_{\lambda_j}(D')$ concept to $v$ that is not already present, because the corresponding $\mathsf{conv}_{\lambda_j}(A_1)$ ($\mathsf{conv}_{\lambda_j}(A_2)$) is missing.
  3. If $\mathcal{B}_{ns}(A_1, v) \neq \emptyset$ and $\mathcal{B}_{ns}(A_2, v) \neq \emptyset$, the $\sqsubseteq_2$-rule is again not applicable after the conversion, because $A_1(v)$ and $A_2(v)$ are replaced by the concept facts $(\mathsf{conv}_{\mu_1}(A_1))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_1))(v)$ and $(\mathsf{conv}_{\mu'_1}(A_2))(v), \ldots, (\mathsf{conv}_{\mu'_{k'}}(A_2))(v)$, respectively, where $\mu_1, \ldots, \mu_k$ and $\mu'_1, \ldots, \mu'_{k'}$ are the completion of the set of variable mappings $\mathsf{mapp}^{G_{ns}}(A_1(v))$ and $\mathsf{mapp}^{G_{ns}}(A_2(v))$. The $\sqsubseteq_2$-rule is, however, only applicable for an axiom $(\mathsf{conv}_\mu(A_1) \sqcap \mathsf{conv}_\mu(A_2)) \sqsubseteq \mathsf{conv}_\mu(D')$ if $\mathsf{conv}_\mu(A_1)$ as well as $\mathsf{conv}_\mu(A_2)$ is in the same label, but $\mathsf{conv}_\mu(D')$ is not already present, i.e., $\mu \in \{\mu_1, \ldots, \mu_k\}$ and $\mu \in \{\mu'_1, \ldots, \mu'_{k'}\}$, but $\mu \notin \{\mu_1, \ldots, \mu_k\} \bowtie^\epsilon \{\mu'_1, \ldots, \mu'_{k'}\}$, which is a contradiction, because $\{\mu_1, \ldots, \mu_k\} \bowtie^\epsilon \{\mu'_1, \ldots, \mu'_{k'}\}$ is the same as the completion of $\mathcal{B}_{ns}(A_1, v) \bowtie^\epsilon \mathcal{B}_{ns}(A_2, v)$ to all possible variables used in $C \sqsubseteq D$.
- The $\downarrow$-concepts are more complicated. Concept facts of the form $\downarrow x.D'(a)$ are not explicitly associated with variable mappings. However, because of the axiom $O \sqsubseteq \downarrow x.D'$, they only occur in the label of ABox individual nodes. Thus, we can use the implicit information that $x$ will be bound to the ABox individual node $a$, and we use the completion of the variable mapping $\{x \mapsto a\}$ for $\mu_1, \ldots, \mu_k$. Therefore, we replace $\downarrow x.D'(a)$ with the concept facts $(\top \sqcap \mathsf{conv}_{\mu_1}(D'))(a), \ldots, (\top \sqcap \mathsf{conv}_{\mu_k}(D'))(a)$. It is not hard to see that $(\top \sqcap \mathsf{conv}_{\mu_1}(D')), \ldots, (\top \sqcap \mathsf{conv}_{\mu_k}(D'))$ cannot be unfolded in $G_{ug}$, because the $\downarrow$-rule ensures that $D'$ is also already present in the label of the node and is associated with the variable mapping $\{x \mapsto a\}$ and, thus, $D'$ is also replaced by $\mathsf{conv}_{\mu_1}(D'), \ldots, \mathsf{conv}_{\mu_k}(D')$. Analogously, for the axioms $\{a\} \sqsubseteq \top \sqcap \mathsf{conv}_{\mu_1}(D'), \ldots, \{a\} \sqsubseteq \top \sqcap \mathsf{conv}_{\mu_k}(D')$ that we have to consider in $G_{ug}$ instead of $O \sqsubseteq \downarrow x.D'$, the rules for these axioms are also not applicable, because the concept $\downarrow x.D'$ in the label of $a$ has been replaced by the concepts $\top \sqcap \mathsf{conv}_{\mu_1}(D'), \ldots, \top \sqcap \mathsf{conv}_{\mu_k}(D')$ and $\downarrow x.D'$ is in the label of $a$, because it is added to every ABox individual node due to the axiom $O \sqsubseteq \downarrow x.D'$.
- The argumentation for the $gr$-concepts and the corresponding rules is very similar. As mentioned before, we assume that the grounding concept is always used to add the remaining, non-absorbable part of the axiom. Thus, $gr(D)$ is always in $\mathsf{aclos}_\mathcal{T}(C \sqsubseteq D)$, even if $\mathsf{Vars}(D) = \emptyset$. Furthermore, we also use the assumption that the absorption of an upfront grounded axiom, by the variable mapping $\mu$, also

uses a special grounding concept $gr_\mu(D)$, which has to be unfolded to $D_{[\mu]}$ and is, therefore, not problematic for the tableau algorithm, because it corresponds to a conjunction with only one conjunct. Thus, a concept fact $gr(D)(v)$ is replaced by $(\mathsf{conv}_{\mu_1}(gr(D)))(v), \ldots, (\mathsf{conv}_{\mu_k}(gr(D)))(v)$, which is the same as $(gr_{\mu_1}(D))(v)$, $\ldots, (gr_{\mu_k}(D))(v)$. Obviously, these replaced grounding concepts cannot be unfolded to $D_{[\mu_1]}, \ldots, D_{[\mu_k]}$, because $D_{[\mu_1]}, \ldots, D_{[\mu_k]}$ are already present due to the application of the $gr$-rule for $gr(D)(v)$, for which also the completion of the associated set of variable mappings is used for the grounding of $D$. □

Next, we show that we can steer our extended tableau algorithm to construct a complete and clash free completion graph $G_{ns}$ for $\mathcal{T}_{ns}$ if there exists a fully expanded and clash free completion graph $G_{ug}$ for $\mathcal{T}_{ug}$ that is constructed by a standard tableau algorithm.

**Lemma 6 (Completeness)** *Let $\mathcal{T}$ be an absorbed TBox, $C \sqsubseteq D$ a nominal schema axiom, $U_1, \ldots, U_h$ the upfront grounding for $C \sqsubseteq D$, and $\mathcal{T}_{ns}$ and $\mathcal{T}_{ug}$ TBoxes as in Definition 5. If there is a fully expanded and clash free completion graph for $\mathcal{T}_{ug}$, then there is a fully expanded and clash free completion graph for $\mathcal{T}_{ns}$.*

*Proof.* Let $G_{ug}$ be a completion graph for $\mathcal{T}_{ug}$ that is obtained by applying only rules for concepts and axioms of $\mathcal{T}$. Since our extended rules coincide with the standard tableau rules if no variable mappings are associated to concept facts, our extended tableau algorithm can create $G_{ns}$, which exactly coincides with $G_{ug}$. We show that the application of a rule in Table 1 to $G_{ns}$ deterministically adds only concept facts and possibly variable mappings, for which the conversion of these facts and variable mappings are also consequences in $G_{ug}$ that are added in the course of applying standard tableau rules to $G_{ug}$. Thus, $G_{ug}$ can obviously be used for steering the non-deterministic decisions for $G_{ns}$ to construct a fully expanded and clash free completion graph if $G_{ug}$ is fully expanded and clash free.

Now, let $G_{ns}$ and $G_{ug}$ be completion graphs for $\mathcal{T}_{ns}$ and $\mathcal{T}_{ug}$, respectively, and $G_{ns}$ and $G_{ug}$ coincide with the inferred facts so far, i.e., the conversion of concept facts and variable mappings from $G_{ns}$ corresponds to the contained concept facts in $G_{ug}$. To show by induction that each rule application for $G_{ns}$ only adds concept facts and variable mappings, for which the conversion of these facts and variable mappings are also consequences in $G_{ug}$, let $\lambda_2, \ldots, \lambda_\ell$ be all possible variable mappings, i.e., $\{\lambda_2, \ldots, \lambda_\ell\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\epsilon\})$. Please note, it suffices to consider only the extended rules for concepts and axioms used for absorbing $C \sqsubseteq D$, because only the concepts in $\mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$ can be associated with variable mappings, for which the extended rules differ to standard rules.

- First, we consider the $\forall$-rule for a concept fact $\forall r.D'(v)$, $\forall r.D' \in \mathsf{aclos}_{\mathcal{T}}(C \sqsubseteq D)$, which adds the concept fact $D'(w)$ to an $r$-neighbour $w$ of $v$ in $G_{ns}$ and possibly the variable mapping $\mu \in \mathcal{B}_{ns}(\forall r.D', v)$ to $B_{ns}(D', w)$. If the $\forall$-rule only adds the concept fact $D'(w)$ for cases where $\mathcal{B}(\forall r.D', v) = \emptyset$, then $\mathsf{mapp}^{G_{ns}}(D'(w)) = \{\epsilon\}$ (which is ensured by the absorption algorithm) and we have to show that in the completion graph $G_{ug}$ the concept facts $(\mathsf{conv}_{\lambda_2}(D'))(w), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(w)$ are also added by rule applications. Obviously, this is the case, because the concept fact $\forall r.D'(v)$

corresponds to $(\mathsf{conv}_{\lambda_2}(\forall r.D'))(v)$, ..., $(\mathsf{conv}_{\lambda_\ell}(\forall r.D'))(v)$ in $G_{ug}$ and by applying the $\forall$-rule for all concept facts $(\mathsf{conv}_{\lambda_j}(\forall r.D'))(v)$, $1 \leq j \leq \ell$, we have the concepts $\mathsf{conv}_{\lambda_2}(D'), \ldots, \mathsf{conv}_{\lambda_\ell}(D')$ in the label of all neighbour nodes. If the $\forall$-rule adds a variable mapping $\mu \in \mathcal{B}_{ns}(\forall r.D', v)$ to $B_{ns}(D', w)$, then we have to show that $(\mathsf{conv}_{\mu_1}(D'))(w), \ldots, (\mathsf{conv}_{\mu_k}(D'))(w)$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\mu\})$ are added to $G_{ug}$ by rule applications. But this is also the case since $\forall r.D'(v)$ corresponds to $(\mathsf{conv}_{\mu_1}(\forall r.D'))(w)$, ..., $(\mathsf{conv}_{\mu_k}(\forall r.D'))(w)$ in $G_{ug}$ and applying the $\forall$-rule for $(\mathsf{conv}_{\mu_1}(\forall r.D'))(w)$, ..., $(\mathsf{conv}_{\mu_k}(\forall r.D'))(w)$ adds $(\mathsf{conv}_{\mu_1}(D'))(w)$, ..., $(\mathsf{conv}_{\mu_k}(D'))(w)$ to the label of all neighbour nodes.

- Next, we consider the $\sqsubseteq_1$-rule for an axiom $H \sqsubseteq D'$ with $H = A$ or $H = \{a\}$ and $D' \neq \downarrow x.D''$ (we consider the addition of the binder concepts together with the $\downarrow$-rule). If $\mathcal{B}_{ns}(H, v) = \emptyset$ and the $\sqsubseteq_1$-rule adds only the concept fact $D'(v)$ to a node, then we have to show that $(\mathsf{conv}_{\lambda_2}(D'))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(v)$ are also added to $G_{ug}$ by rule applications. Again, this is obviously the case, because for $G_{ug}$ we have the rules $\mathsf{conv}_{\lambda_2}(H) \sqsubseteq \mathsf{conv}_{\lambda_2}(D'), \ldots, \mathsf{conv}_{\lambda_\ell}(H) \sqsubseteq \mathsf{conv}_{\lambda_\ell}(D')$. If $\mathcal{B}_{ns}(H, v) \neq \emptyset$, then $H = A$, the $\sqsubseteq_1$-rule adds also a variable mapping $\mu$ to $\mathcal{B}_{ns}(D', v)$ and we have to show that $(\mathsf{conv}_{\mu_1}(D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(D'))(v)$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\mu\})$ are added to $G_{ug}$ by rule applications. Again, this is a consequence of the concept facts $(\mathsf{conv}_{\mu_1}(A))(v)$, ..., $(\mathsf{conv}_{\mu_k}(A))(v)$ in $G_{ug}$ and the axioms $\mathsf{conv}_{\mu_1}(A) \sqsubseteq \mathsf{conv}_{\mu_1}(D'), \ldots, \mathsf{conv}_{\mu_k}(A) \sqsubseteq \mathsf{conv}_{\mu_k}(D')$ that we have to consider for $G_{ug}$.
- Let us now consider the $\sqsubseteq_2$-rule for an axiom $(A_1 \sqcap A_2) \sqsubseteq D'$. If the $\sqsubseteq_2$-rule only adds the concept fact $D'(v)$, then we have to show that $(\mathsf{conv}_{\lambda_2}(D'))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(v)$ are also added to $G_{ug}$ by rule applications. However, this is the case, because $A_1(v)$ and $A_2(v)$ corresponds to $(\mathsf{conv}_{\lambda_j}(A_1))(v)$ and $(\mathsf{conv}_{\lambda_j}(A_2))(v)$ in $G_{ug}$, respectively, and, since we have the axiom $(\mathsf{conv}_{\lambda_j}(A_1) \sqcap \mathsf{conv}_{\lambda_j}(A_2)) \sqsubseteq \mathsf{conv}_{\lambda_j}(D)$ for each $1 \leq j \leq \ell$, it follows that all $(\mathsf{conv}_{\lambda_2}(D'))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(D'))(v)$ are also added to $G_{ug}$. If the $\sqsubseteq_2$-rule also adds the variable mapping $\mu$ to $\mathcal{B}_{ns}(D', v)$, then we have to show that $(\mathsf{conv}_{\mu_1}(D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(D'))(v)$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\mu\})$ are also added to $G_{ug}$ by rule applications. Let us first assume that $\mathcal{B}_{ns}(A_1, v) = \emptyset$ ($\mathcal{B}_{ns}(A_2, v) = \emptyset$). As a consequence, we have in $G_{ug}$ the concept facts $(\mathsf{conv}_{\lambda_2}(A_2))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(A_2))(v)$ and $(\mathsf{conv}_{\mu_1}(A_2))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_2))(v)$ ($(\mathsf{conv}_{\lambda_2}(A_1))(v), \ldots, (\mathsf{conv}_{\lambda_\ell}(A_1))(v)$ and $(\mathsf{conv}_{\mu_1}(A_1))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_1))(v)$). As a consequence of the axioms $(\mathsf{conv}_{\lambda_j}(A_1) \sqcap \mathsf{conv}_{\lambda_j}(A_2)) \sqsubseteq \mathsf{conv}_{\lambda_j}(D)$, for all $1 \leq j \leq \ell$, the concept facts $(\mathsf{conv}_{\mu_1}(D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(D'))(v)$ are also added to $G_{ug}$ by rule applications. Let us now assume that $\mathcal{B}_{ns}(A_1, v) \neq \emptyset$ as well as $\mathcal{B}_{ns}(A_2, v) \neq \emptyset$. We show that $(\mathsf{conv}_{\mu_1}(D'))(v), \ldots, (\mathsf{conv}_{\mu_k}(D'))(v)$ has to be added to $G_{ug}$, because $(\mathsf{conv}_{\mu_1}(A_1))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_1))(v)$ as well as $(\mathsf{conv}_{\mu_1}(A_2))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_2))(v)$ are in $G_{ug}$. Obviously, there exists the variable mappings $\mu' \in \mathcal{B}_{ns}(A_1, v)$ and $\mu'' \in \mathcal{B}_{ns}(A_2, v)$ with $\mu = \mathsf{dom}(\mu') \cup \mathsf{dom}(\mu'')$ and for each $x \in (\mathsf{dom}(\mu') \cap \mathsf{dom}(\mu''))$ it holds that $\mu'(x) = \mu''(x)$. Thus, $\mu' \subseteq \mu$ and $\mu'' \subseteq \mu$ and as a consequence of the completion of $\mu'$ and $\mu''$ it follows that $\{\mu_1, \ldots, \mu_k\} \subseteq \{\mu'_1, \ldots, \mu'_k\}$ and $\{\mu_1, \ldots, \mu_k\} \subseteq \{\mu''_1, \ldots, \mu''_k\}$. Therefore, $(\mathsf{conv}_{\mu_1}(A_1))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_1))(v)$ and $(\mathsf{conv}_{\mu_1}(A_2))(v), \ldots, (\mathsf{conv}_{\mu_k}(A_2))(v)$ are at least also in $G_{ug}$.
- The $\downarrow$-rule for a concept fact $\downarrow x.D'(a)$ adds $D'$ to the label of $a$ and the variable mapping $\{x \mapsto a\}$ to $\mathcal{B}_{ns}(D', a)$. We have to show that the concept facts

$(\mathsf{conv}_{\mu_1}(D'))(a), \ldots, (\mathsf{conv}_{\mu_k}(D'))(a)$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{x \mapsto a\})$ are also added to $G_{ug}$ by rule applications. But this is obviously the case, because in $G_{ug}$ we have the concept facts $(\mathsf{conv}_{\mu_1}(\downarrow x.D'))(a), \ldots, (\mathsf{conv}_{\mu_k}(\downarrow x.D'))(a)$, which is nothing else than $(\top \sqcap \mathsf{conv}_{\mu_1}(D'))(a), \ldots, (\top \sqcap \mathsf{conv}_{\mu_k}(D'))(a)$. Furthermore, we have to show that $(\top \sqcap \mathsf{conv}_{\mu_1}(D'))(a), \ldots, (\top \sqcap \mathsf{conv}_{\mu_k}(D'))(a)$ is added to $G_{ug}$, because, as a consequence of the axiom $O \sqsubseteq \downarrow x.D'$, $\downarrow x.D'(a)$ is added to $G_{ns}$. Obviously, this is the case, because for $G_{ug}$ we have the axioms $\{a\} \sqsubseteq \top \sqcap \mathsf{conv}_{\mu_1}(D'), \ldots, \{a\} \sqsubseteq \top \sqcap \mathsf{conv}_{\mu_k}(D')$.

- The application of the $gr$-rule adds for a concept fact $gr(D')(v)$ and a (possibly empty) variable mapping $\mu \in \mathcal{B}^{\epsilon}_{ns}(gr(D'), v)$ the concept facts $D_{[\mu_1]}, \ldots, D_{[\mu_k]}$ with $\{\mu_1, \ldots, \mu_k\} = \mathsf{comp}^{\mathcal{T}}_{\mathsf{Vars}(\neg C \sqcup D)}(\{\mu\})$. We have to show that $D_{[\mu_1]}, \ldots, D_{[\mu_k]}$ is also added to $G_{ug}$ by rule applications. Again, this is obviously the case, because in $G_{ug}$ we have the concept facts $(\mathsf{conv}_{\mu_1}(gr(D')))(v), \ldots, (\mathsf{conv}_{\mu_k}(gr(D')))(v)$, which is the same as $gr_{\mu_1}(D')(v), \ldots, gr_{\mu_k}(D')(v)$. □

The extended tableau algorithm is still terminating. This is due to the fact that the number of variable mappings is limited by the number of ABox individuals and the number of variables in axioms. Thus, blocking is ensured since the nodes in the completion graph can only be labelled with a limited number of concepts and only a limited number of variable mappings can be associated to these concepts.

**Lemma 7 (Termination)** *The tableau algorithm extended by the rules in Table 1 is terminating for absorbed TBoxes with nominal schema axioms.*