Optimizing the Nominal Introduction Rule in (Hyper)Tableau Calculi

Boris Motik, Rob Shearer, and Ian Horrocks

Oxford University Computing Laboratory

1 Introduction

A tableau calculus for $\mathcal{SHIQ}$ has been known for some time, and forms the basis for several highly successful implementations [1, 8, 9]. Extending this calculus to $\mathcal{SHOIQ}$, however, was notoriously difficult due to an interaction between nominals, inverse roles, and number restrictions which results in (partial) loss of the forest-model property for models of $\mathcal{SHOIQ}$ knowledge bases. In this paper, we analyze the problems that arise from this combination of features and present an overview of two solutions.

The first solution, described in [3], extends the $\mathcal{SHIQ}$ calculus with an *NN-rule* which extends the non-forest-like portion of a model with new individuals. Applications of this rule can be highly nondeterministic, however, and can substantially increase the size of the generated models, which leads to inefficiency.

We present an alternative solution based on a new *NI-rule*. Our approach does not introduce new individuals; instead, existing individuals are incorporated into the non-forest-like portion of the model. We implemented our solution in HermiT.[1] As we discuss in [7], this approach seems to work well in practice.

2 Preliminaries

To show that a description logic knowledge base $\mathcal{K} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ is satisfiable, a tableau algorithm constructs a sequence of ABoxes $\mathcal{A} = \mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n$ called a *derivation*, where each $\mathcal{A}_i$ is obtained from $\mathcal{A}_{i-1}$ by an application of one *inference rule*.[2] The inference rules make the information implicit in the axioms of $\mathcal{R}$ and $\mathcal{T}$ explicit, and thus evolve the ABox $\mathcal{A}$ towards a (representation of a) model of $\mathcal{K}$. The algorithm terminates either if no inference rule is applicable to some $\mathcal{A}_n$, in which case $\mathcal{A}_n$ represents a model of $\mathcal{K}$, or if $\mathcal{A}_n$ contains an obvious contradiction, in which case the model construction has failed. The following inference rules are commonly used in DL tableau calculi.

- $\sqcup$-rule: Given $(C_1 \sqcup C_2)(s)$, derive either $C_1(s)$ or $C_2(s)$.
- $\sqcap$-rule: Given $(C_1 \sqcap C_2)(s)$, derive $C_1(s)$ and $C_2(s)$.

[1] `http://web.comlab.ox.ac.uk/oucl/work/boris.motik/HermiT/`

[2] Some formalizations of tableau algorithms work on *completion graphs*, which have a natural correspondence to ABoxes.

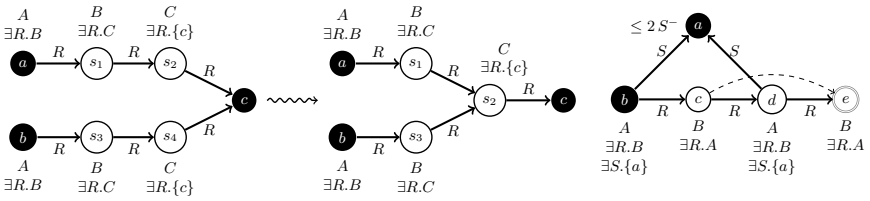
– $\exists$-rule: Given $(\exists R.C)(s)$, derive $R(s,t)$ and $C(t)$ for t a fresh individual.
– $\forall$-rule: Given $(\forall R.C)(s)$ and $R(s,t)$, derive $C(t)$.
– $\sqsubseteq$-rule: Given a GCI $C \sqsubseteq D$ and an individual s, derive $(\neg C \sqcup D)(s)$.
– $\leq$-rule: Given $\leq n\, R.C(s)$, $R(s,t_0), ..., R(s,t_n)$, and $C(t_0), ..., C(t_n)$, choose t_i and t_j for some $0 \leq i < j \leq n$, prune t_i, and merge it into t_j.

The individuals present in the original ABox $\mathcal{A}$ are called *named individuals*, and they can be connected by role assertions in an arbitrary way. The individuals introduced by the $\exists$- and $\geq$-rules are called *blockable individuals*. For example, if $\exists R.C(a)$ is expanded into $R(a,s)$ and $C(s)$, then s is called a blockable individual and it is the *R-successor* of a. It is not difficult to see that no tableau inference rule can connect s with some element of $\mathcal{A}$: the individual s can participate only in inferences that derive an assertion of the form $D(s)$ or create a new successor or s. Hence, each ABox $\mathcal{A}'$ obtained from $\mathcal{A}$ can be seen as a "forest": each named individuals can be arbitrarily connected to other named individuals and to a tree of blockable successors. The *concept label* $\mathcal{L}_{\mathcal{A}}(s)$ is defined as the set of all concepts C such that $C(s) \in \mathcal{A}$, and the *edge label* $\mathcal{L}_{\mathcal{A}}(s,s')$ as the set of all atomic roles such that $R(s,s') \in \mathcal{A}$.

A naïve application of the tableau rules does not terminate if the TBox contains existential quantifiers in cycles: given a TBox $\mathcal{T} = \{\top \sqsubseteq \exists R.\top\}$, naïve applications of the $\exists$-rule produce an infinite chain of blockable individuals. To ensure termination is such cases, tableau algorithms employ *blocking* [4]. DLs such as $\mathcal{SHIQ}$ and $\mathcal{SHOIQ}$ allow for inverse roles and number restrictions, which require *ancestor pairwise blocking* [4]: for s and t blockable individuals and s' and t' any individuals of an ABox $\mathcal{A}$, t blocks s if and only t is an ancestor of s, $\mathcal{L}_{\mathcal{A}}(s) = \mathcal{L}_{\mathcal{A}}(t)$, $\mathcal{L}_{\mathcal{A}}(s') = \mathcal{L}_{\mathcal{A}}(t')$, $\mathcal{L}_{\mathcal{A}}(s,s') = \mathcal{L}_{\mathcal{A}}(t,t')$, and $\mathcal{L}_{\mathcal{A}}(s',s) = \mathcal{L}_{\mathcal{A}}(t',t)$. In tableau algorithms, the $\exists$- and $\geq$-rules are applicable only to nonblocked individuals, which ensures termination: the number of different concept and edge labels is exponential in $|\mathcal{K}|$, so an exponentially long branch in a forest-like ABox must contain a blocked individual, thus limiting the length of each branch in an ABox. Let $\mathcal{A}$ be an ABox to which no tableau inference rule is applicable, and in which s is blocked by t. We can construct a model from $\mathcal{A}$ by *unraveling*—that is, by replicating the fragment between s and t infinitely often. Intuitively, blocking ensures that the part of the ABox between s and s' "behaves" just like the part between t and t', so unraveling indeed generates a model. If our logic were able to connect blockable individuals in a non-tree-like way, then unraveling would not generate a model; in fact, the notion of ancestors, descendants, and blocking would itself be ill-defined.

3 Nominals, Inverse Roles, and Number Restrictions

We now discuss the difficulties that arise when extending tableau calculi to handle nominals, inverse roles, and number restrictions. We summarize both the original solution presented in [3], as well as our novel approach which we incorporated into our hypertableau calculus.



(a) Non-Tree-Like Structures due to Merging (b) Blocking and Unraveling

Fig. 1: Problems caused by nominals. Named individuals are shown in black, blocked individuals in white, and blocking is indicated with a dashed line.

3.1 The Main Problems

There are two primary problems that must be addressed when extending a $\mathcal{SHIQ}$ tableau calculus with nominals.

Nonforest models. Nominals can make ABoxes non-forest-like, as the following simple example demonstrates.

$$\begin{aligned} \mathcal{A}_1 &= \{\ A(a), \quad A(b)\ \} \\ \mathcal{T}_1 &= \{\ A \sqsubseteq \exists R.B,\ \ B \sqsubseteq \exists R.C,\ \ C \sqsubseteq \exists R.\{c\}\ \} \end{aligned} \tag{1}$$

Successive rule applications on $\mathcal{A}_1$ and $\mathcal{T}_1$ can produce the ABox $\mathcal{A}_1^1$ shown on the left-hand side of Figure 1a. This ABox is clearly not forest-shaped: the two R-paths in $\mathcal{A}_1^1$ start at the named individuals a and b, respectively, and end in a named individual c. This by itself is not a problem: termination of model construction on such *extended forest-like* ABoxes can be ensured easily if the DL allows for nominals, but not in the presence of both inverse roles and number restrictions [2].

Assume now that we extend $\mathcal{T}_1$ with the axiom $\top \sqsubseteq \leq 1\, R^-$. On $\mathcal{A}_1^1$, this might merge s_2 into s_4. Note that both s_2 and s_4 are blockable individuals; furthermore, neither individual is an ancestor of the other, so we can merge, say, s_4 into s_2. This produces the ABox $\mathcal{A}_1^2$ shown on the right-hand side of Figure 1a. The assertion $R(s_3, s_2)$ makes $\mathcal{A}_1^2$ not forest-shaped. By extending the example, it is possible to use nominals, inverse roles, and number restrictions to arrange blockable individuals in cycles. This loss of forest-shaped models prevents the use of blocking and pruning, thus invalidating the standard termination arguments.

Blocking and unraveling. Even if the blockable individuals in a tableau maintain a forest structure, nominals can prevent the successful application of blocking and unraveling. Consider the following knowledge base.

$$\begin{aligned} \mathcal{A}_2 &= \{\ A(b)\ \} \\ \mathcal{T}_2 &= \left\{\ A \sqsubseteq \exists R.B,\ \ A \sqsubseteq \exists S.\{a\},\ \ B \sqsubseteq \exists R.A,\ \ \top \sqsubseteq \leq 2\, S^-\ \right\} \end{aligned} \tag{2}$$

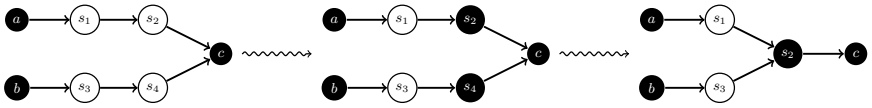


Fig. 2: The Introduction of Root Individuals

A tableau expansion of $\mathcal{A}_2$ and $\mathcal{T}_2$ would produce the ABox $\mathcal{A}_2^1$ shown in Figure 1b. Individual e is blocked in $\mathcal{A}_2^1$ by the individual c, so the derivation terminates. Note that the last axiom from $\mathcal{T}_2$ is satisfied: a is the only individual in $\mathcal{A}_2^1$ that has S^--successors and it has two such successors. To construct a model from $\mathcal{A}_2^1$, we unravel the blocked parts of the ABox—that is, we construct an infinite path that extends past e by "duplicating" the fragment of the model between c and e an infinite number of times. This, however, creates additional S^--successors of e, which invalidates the last axiom from $\mathcal{T}_2$; thus, the unraveled ABox does not define a model of $\mathcal{A}_2$ and $\mathcal{T}_2$.

3.2 A Naïve Solution

To solve the above problems, we need to extend the arbitrarily interconnected part of the ABox: in addition to named and blockable individuals, we introduce *root individuals*—freshly introduced individuals which can be arbitrarily connected to named individuals and other root individuals. We apply the following test (*): if an ABox $\mathcal{A}$ contains assertions $R(s, a)$, $A(s)$, and $\leq n\, R^-.A(a)$, where a is a root or a named individual and s is a blockable individual that is not a successor of a, then we change s into a root individual.

Applied to $\mathcal{A}_1^2$, this condition changes the status of s_2 from a blockable individual into a root individual. After this change, only s_1 and s_3 are blockable in $\mathcal{A}_1^2$, so the ABox has the extended forest-like shape and we can apply blocking and pruning as usual. This is schematically shown in Figure 2.

Promotion of blockable individuals to roots also elegantly solves the blocking and unraveling problem. In $\mathcal{A}_2$ and $\mathcal{T}_2$, because a must satisfy an at-most restriction of the form $\leq 2\, S^-$, as soon as $S(d, a)$ is derived, condition (*) is fulfilled and we turn d into a root individual.

This solution, however, introduces another problem: the number of root individuals can now grow arbitrarily, as shown in the following example.

$$\mathcal{A}_3 = \{\ A(b)\ \} \quad \mathcal{T}_3 = \{\ A \sqsubseteq \exists R.A,\ A \sqsubseteq \exists S.\{a\}\ \top \sqsubseteq \leq 2\, S\ \} \tag{3}$$

On $\mathcal{A}_3$ and $\mathcal{T}_3$, rule applications can produce the derivation sequence shown in Figure 3: after c and d are introduced as descendents of b, they satisfy condition (*) so we change them into root individuals. Subsequently, the third axiom from $\mathcal{T}_3$ is not satisfied, so we merge two neighbors of a; we choose to merge c into b. Since d is now not a blockable individual, we cannot prune it. The first axiom of $\mathcal{T}_3$ is not satisfied for d, so we can extend the ABox with a new successor e which also satisfies (*) and becomes a root individual. If we continue to merge successors of b into b we can repeat the same inferences forever.

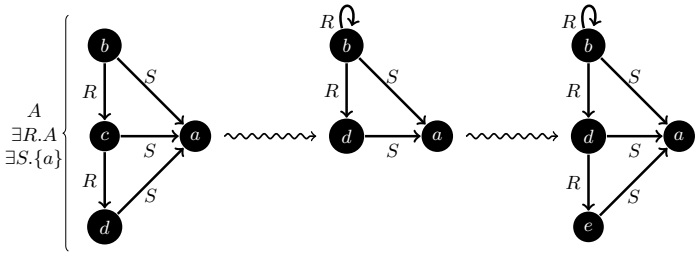


Fig. 3: A Modified Yo-Yo Example

3.3 The Existing Solution

To guarantee termination, the calculus from [3] uses an *NN*-rule which refines condition (*). Assume that an ABox $\mathcal{A}$ contains assertions $R(s, a)$ and $A(s)$ where s is a blockable individual that is not a successor of a root or a named individual a; furthermore, assume that a must satisfy an at-most restriction of the form $\leq n\, R^-.A$. If $\mathcal{A}$ already contains root individuals $z_1, ..., z_n$ such that $\bigcup_{1\leq i\leq n}\{A(z_i), R(z_i, a)\} \cup \{z_i \neq z_j \mid 1 \leq i < j \leq m\} \subseteq \mathcal{A}$, then the $\leq$-rule simply merges s into some z_i; no new root individual needs to be introduced. If $\mathcal{A}$ does not contain such $z_1, ..., z_n$, the *NN*-rule nondeterministically guesses the exact number $m \leq n$ of R^--neighbors of a that are members of A, generates m fresh root individuals $w_1, ..., w_m$, and extends $\mathcal{A}$ with the assertions

$$\{A(w_i), R(w_i, a) \mid 1 \leq i \leq m\} \cup \{w_i \neq w_j \mid 1 \leq i < j \leq m\} \cup \{\leq m\, R.A(a)\}.$$

This allows the *NN*-rule to be applied at most once for each concept of the form $\leq n\, R^-.A$ and each root individual, which ensures termination: unlike in the naïve solution outlined in the previous section, fresh root individuals are not created whenever a root individuals gains a blockable neighbor.

For example, Figure 4 shows applications of the *NN*- and $\leq$-rules to the ABox $\mathcal{A}_4 = \{\ \exists R.\exists R.\{c\}(a),\ \leq 3\, R^-.\top(c)\ \}$. In $\mathcal{A}_4^1$, shown at the left of the figure, the named individual c has a blockable R^--neighbor and must satisfy the restriction $\leq 3\, R^-.\top$. The *NN*-rule nondeterministically chooses whether to introduce one, two, or three fresh root individuals; the parallel branches of the derivation are shown in the center of the figure. The introduction of a single individual, shown in the top branch, results in deterministic application of the $\leq$-rule as the blockable individual b is merged into the fresh root individual z_1, shown in the upper right of the figure. For derivation paths on which more than one fresh root individual is introduced, application of the $\leq$-rule is nondeterministic: b can be merged into any of the new root individuals, and the derivation branches further.

Although the *NN*-rule does ensure termination of the tableau algorithm, it is a potential source of inefficiency in knowledge bases in which large numbers appear within at-most concepts: an application of the *NN*-rule involving a concept $\leq n\, R.C$ guesses among n different possible sizes for the neighbor set, and subsequent applications of the $\leq$-rule must choose how to merge the new roots with blockable individuals. In the case of just a single blockable neighbor, this

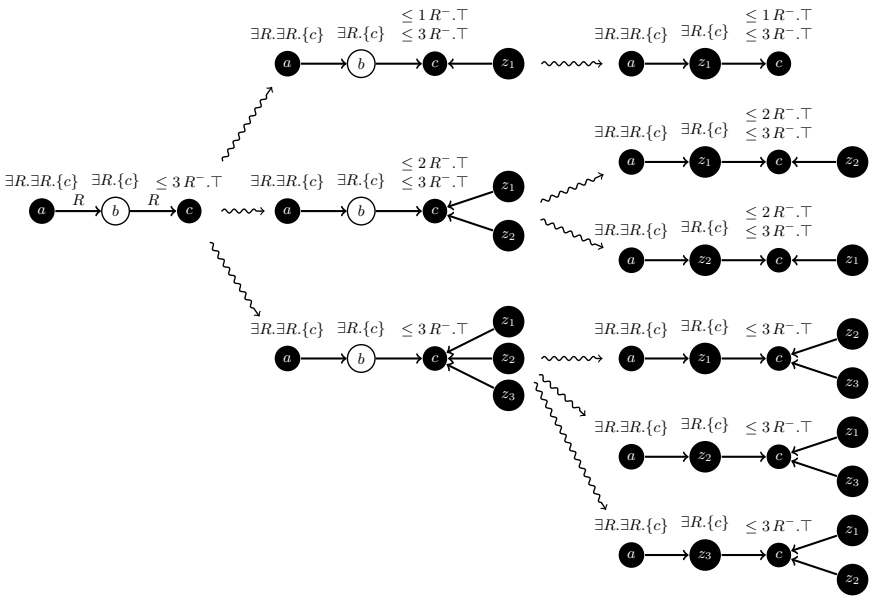


Fig. 4: An application of the NN-rule

results in a derivation tree with n^2 branches. Furthermore, the introduction new root individuals can result in unnecessary processing and large models.

3.4 The *NI*-rule

As a replacement for the NN-rule described above, we introduce a new NI-rule, which refines the generation of root individuals. Let us again consider the ABox $\mathcal{A}$ containing assertions $R(s, a)$ and $A(s)$, and $\leq n\, R^-.A(a)$, where s is a blockable individual that is not a successor of a root or a named individual a. In any model of $\mathcal{A}$, we can have at most n different individuals b_i that participate in assertions of the form $R(b_i, a)$ and $A(b_i)$. Hence, we associate with a in advance a set of n fresh root individuals $\{b_1, \ldots, b_n\}$. Instead of choosing some subset of these individuals to introduce and relying upon the $\leq$-rule to merge them with blockable neighbors, however, we promote blockable individuals to root individuals directly: to turn s into a root individual, we nondeterministically choose b_j from this set and merge s into b_j. In this way, the number of new root individuals that can be introduced for a number restriction in the label of an individual a is limited to n. An application of our NI-rule to the ABox $\mathcal{A}_4 = \{\ \exists R.\exists R.\{c\}(a),\ \leq 3\, R^-.\top(c)\ \}$ is given in Figure 5. By exploiting a bound on the length of paths of blockable individuals in an ABox, we can establish a bound on the number of root individuals introduced in a derivation, which ensures termination of the algorithm.

When formulating the NI-rule, we are faced with a technical problem: concepts of the form $\leq n\, R.A$ are translated in our calculus into DL-clauses, which

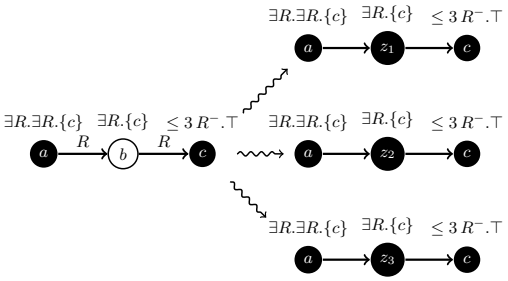


Fig. 5: An application of the NI-rule

makes testing the condition from the previous paragraph difficult. For example, an application of the *Hyp*-rule to the third DL-clause in (3) (obtained from the axiom $\top \sqsubseteq \leq 2\, S^-.\top$) can produce an equality such as $c \approx b$. This equality alone does not reflect the fact that a must satisfy the at-most restriction $\leq 2\, S^-.\top$. To enable the application of the NI-rule, we introduce a notion of *annotated equalities*, in which the annotations establish an association with the at-most restriction. The details of our procedure are presented in the next section.

4 A Hypertableau Calculus for $\mathcal{SHOIQ}$

We now summarize the hypertableau algorithm that can be used to check satisfiability of a $\mathcal{SHOIQ}$ knowledge base $\mathcal{K}$. This is extension of the procedure described in [6] and [5], and a full version of the calculus is presented in [7].

Our algorithm first preprocesses a $\mathcal{SHOIQ}$ knowledge base into an ABox and a set of *DL-clauses*—implications of the form $\bigwedge_{i=1}^{n} U_i \rightarrow \bigvee_{j=1}^{m} V_j$, where U_i are of the form $R(x, y)$ or $A(x)$, and V_j are of the form $R(x, y)$, $A(x)$, $\exists R.C(x)$, $\geq n\, R.C(x)$, or $x \approx y$. Due to lack of space, we leave the details of this transformation to [7]. In fact, the preprocessing produces DL-clauses of a certain syntactic structure, which guarantees termination of the model construction. In the rest of this paper, we often use the following function ar. Given a role R and variables or constants s and t, this function returns an atom that is semantically equivalent to $R(s, t)$ but that contains an atomic role.

$$\mathsf{ar}(R, s, t) = \begin{cases} R(s, t) \text{ if } R \text{ is an atomic role} \\ S(t, s) \text{ if } R \text{ is an inverse role and } R = S^- \end{cases}$$

Definition 1 (HT-Clause). *We assume that the set of atomic concepts N_C contains a* nominal guard concept *O_a for each individual a; these concepts should not occur in any input knowledge base.*

An at-most equality *is an atom of the form $s \approx t\, @^u_{\leq n\, R.B}$, where s, t, and u are constants or variables, n is a nonnegative integer, R is a role, and B is a literal concept. Semantically, this atom is equivalent to $s \approx t$; the* annotation *$@^u_{\leq n\, R.B}$ is only used to ensure termination of the hypertableau calculus.*

An HT-clause *is a DL-clause r of the form*

$$U_1 \wedge \ldots \wedge U_m \rightarrow V_1 \vee \ldots \vee V_n, \tag{4}$$

in which it must be possible to separate the variables into a center variable x *and a set of* branch variables y_i *such that the following properties hold, where A is an atomic concept, B is a literal but not a nominal guard concept, O_a is a nominal guard concept, R and S are atomic roles, and T is a role.*

- *Each atom in the antecedent of r is of the form $A(x)$, $R(x, y_i)$, $R(y_i, x)$, or $A(y_i)$.*
- *Each atom in the consequent of r is of the form $B(x)$, $B(y_i)$, $R(x, y_i)$, $R(y_i, x)$, $x \approx y_i$, or $y_i \approx y_j \, @^x_{\leq n\, T.B}$.*
- *For each atom $R(x, y_i)$ or $R(y_i, x)$ in the consequent of r, the antecedent of r contains an atom $S(x, y_i)$, $S(y_i, x)$, $O_a(x)$, or $O_a(y_i)$.*
- *For each equality $x \approx y_i$ in the consequent of r, the antecedent of r contains an atom $O_a(x)$ or $O_a(y_i)$.*
- *Each equality $y_i \approx y_j \, @^x_{\leq h\, T.A}$ in the consequent of r must occur in a subclause of r of the form (5) and no y_k with $1 \leq k \leq h+1$ should occur elsewhere in r.*

$$\ldots \bigwedge_{k=1}^{h+1} [\mathsf{ar}(T, x, y_k) \wedge A(y_k)] \ldots \rightarrow \ldots \bigvee_{1 \leq k < \ell \leq h+1} y_k \approx y_\ell \, @^x_{\leq h\, T.A} \cdots \tag{5}$$

- *Each equality $y_i \approx y_j \, @^x_{\leq n\, T.\neg A}$ in the consequent of r must occur in a subclause of r of the form (6) and no y_k with $1 \leq k \leq h+1$ should occur elsewhere in r.*

$$\ldots \bigwedge_{k=1}^{h+1} \mathsf{ar}(T, x, y_k) \ldots \rightarrow \ldots \bigvee_{k=1}^{h+1} A(y_k) \vee \bigvee_{1 \leq k \leq \ell \leq h+1} y_k \approx y_\ell \, @^x_{\leq h\, T.\neg A} \cdots \tag{6}$$

We now present the hypertableau calculus for deciding satisfiability of an ABox $\mathcal{A}$ and a set of HT-clauses $\mathcal{C}$. In our algorithm, we call the individuals that occur in the input ABox *named*. Furthermore, for a named individual a, the *NI*-rule might need to introduce individuals that are unique for a, a role R, a literal concept B, and some integer i; we represent such individuals as $a.\langle R, B, i\rangle$. Since the *NI*-rule might be applied to these individuals as well, we introduce the notion of *root individuals*—finite strings of the form $a.\gamma_1.\ldots.\gamma_n$ where a is a named individual and each γ_ℓ is of the form $\langle R.B.i\rangle$.

In standard tableau algorithms, the tree structure of the model is encoded in the edges between individuals, which always point from parents to children. In contrast, our algorithm encodes the parent-child relationships into individuals themselves: it represents individuals as finite strings of the form $s.i_1, i_2, \ldots, i_n$, where s is a root individual and i_j are integers. For example, $a.2$ is the second child of the named individual a. Individuals with $n \geq 1$ are called *blockable*.

Definition 2 (Hypertableau Algorithm).

Individuals. *Given a set of* named *individuals N_I, the set of* root individuals *N_O is the smallest set such that $N_I \subseteq N_O$ and, if $x \in N_O$, then $x.\langle R, B, i\rangle \in N_O$ for each role R, literal concept B, and integer i. The set of* all individuals *N_A is the smallest set such that $N_O \subseteq N_A$ and, if $x \in N_A$, then $x.i \in N_A$ for each integer i. The individuals in $N_A \setminus N_O$ are* blockable individuals. *A blockable individual $x.i$ is a* successor *of x, and x is a* predecessor *of $x.i$.* Descendant *and* ancestor *are the transitive closures of successor and predecessor, respectively.*

ABoxes. *The hypertableau algorithm operates on* generalized ABoxes*, which are obtained by extending the standard notion of ABoxes as follows.*

- *In addition to standard assertions, an ABox can contain at-most equalities and a special assertion $\bot$ that is false in all interpretations. Furthermore, assertions can refer to the individuals from N_A and not only from N_I.*
- *Each (in)equality $s \approx t$ ($s \not\approx t$) also stands for the symmetric (in)equality $t \approx s$ ($t \not\approx s$). The same is true for annotated at-most equalities.*
- *An ABox $\mathcal{A}$ can contain* renamings *of the form $a \mapsto b$ where a and b are root individuals. The relation $\mapsto$ in $\mathcal{A}$ must be acyclic, $\mathcal{A}$ can contain at most one renaming $a \mapsto b$ for an individual a, and, if $\mathcal{A}$ contains $a \mapsto b$, then a should not occur in any assertion or (in)equality in $\mathcal{A}$. An individual b is the* canonical name *of a root individual a in $\mathcal{A}$, written $b = \mathsf{can}_{\mathcal{A}}(a)$, iff $a \mapsto^+_{\mathcal{A}} b$ and no individual c exists such that $b \mapsto^+_{\mathcal{A}} c$, where $\mapsto^+_{\mathcal{A}}$ is the transitive closure of $\mapsto$ in $\mathcal{A}$.*

An ABox that contains only named individuals and no at-most equalities is called an input ABox.

Pairwise Anywhere Blocking. *For A an atomic concept and R a role, concepts of the form A, $\geq n\, R.A$, or $\geq n\, R.\neg A$, are called* blocking-relevant. *The* labels of an individual *s and* of an individual pair *$\langle s, t\rangle$ in an ABox $\mathcal{A}$ are defined as follows:*

$$\mathcal{L}_{\mathcal{A}}(s) = \{C \mid C(s) \in \mathcal{A} \text{ and } C \text{ is a blocking-relevant concept}\}$$
$$\mathcal{L}_{\mathcal{A}}(s,t) = \{R \mid R(s,t) \in \mathcal{A}\}$$

Let $\prec$ be a strict ordering (i.e., a transitive and irreflexive relation) on N_A containing the ancestor relation—that is, if s' is an ancestor of s, then $s' \prec s$. By induction on $\prec$, we assign to each individual s in $\mathcal{A}$ a status as follows:

- *a blockable individual $s.i$ is* directly blocked by *a blockable individual $t.j$ iff*
  - *$t.j$ is not blocked,*
  - *$t.j \prec s.i$,*
  - *$\mathcal{L}_{\mathcal{A}}(s.i) = \mathcal{L}_{\mathcal{A}}(t.j)$ and $\mathcal{L}_{\mathcal{A}}(s) = \mathcal{L}_{\mathcal{A}}(t)$, and*
  - *$\mathcal{L}_{\mathcal{A}}(s, s.i) = \mathcal{L}_{\mathcal{A}}(t, t.j)$ and $\mathcal{L}_{\mathcal{A}}(s.i, s) = \mathcal{L}_{\mathcal{A}}(t.j, t)$;*
- *s is* indirectly blocked *iff it has a predecessor that is blocked; and*
- *s is* blocked *iff it is either directly or indirectly blocked.*

Pruning. *The ABox* $\mathsf{prune}_{\mathcal{A}}(s)$ *is obtained from* $\mathcal{A}$ *by removing all assertions containing a descendent of* s*.*

Merging. *The ABox* $\mathsf{merge}_{\mathcal{A}}(s \to t)$ *is obtained from* $\mathsf{prune}_{\mathcal{A}}(s)$ *by replacing the individual* s *with the individual* t *in all assertions (but not in renamings) and, if both* s *and* t *are root individuals, adding the renaming* $s \mapsto t$*.*

Derivation Rules. *Table 1 specifies* derivation rules *that, given an ABox* $\mathcal{A}$ *and a set of HT-clauses* $\mathcal{C}$*, derive the ABoxes* $\mathcal{A}_1, \ldots, \mathcal{A}_n$*. In the Hyp-rule,* σ *is a mapping from the set of variables* N_V *to the individuals occurring in the assertions of* $\mathcal{A}$*, and* $\sigma(U)$ *is the result of replacing each variable* x *in the atom* U *with* $\sigma(x)$*.*

Rule Precedence. *The* $\approx$*-rule can be applied to a (possibly annotated) equality* $s \approx t$ *in an ABox* $\mathcal{A}$ *only if* $\mathcal{A}$ *does not contain an equality* $s \approx t \,@^{u}_{\leq n\, R.B}$ *to which the NI-rule is applicable.*

Clash. *An ABox* $\mathcal{A}$ *contains a* clash *iff* $\bot \in \mathcal{A}$*; otherwise,* $\mathcal{A}$ *is* clash-free*.*

Derivation. *For a set of HT-clauses* $\mathcal{C}$ *and an ABox* $\mathcal{A}$*, a* derivation *is a pair* (T, λ) *where* T *is a finitely branching tree and* λ *is a function that labels the nodes of* T *with ABoxes such that, for each node* $t \in T$*,*

- $\lambda(t) = \mathcal{A}$ *if* t *the root of* T*,*
- t *is a leaf of* T *if* $\bot \in \lambda(t)$ *or no derivation rule is applicable to* $\lambda(t)$ *and* $\mathcal{C}$*, and*
- t *has children* $t_1, \ldots, t_n$ *such that* $\lambda(t_1), \ldots, \lambda(t_n)$ *are exactly the results of applying one (arbitrarily chosen, but respecting the rule precedence) applicable rule to* $\lambda(t)$ *and* $\mathcal{C}$ *otherwise.*

We stress several important aspects of Definition 2. The *NI*-rule is never applied to an at-most equality of the form $s \approx t \,@^{u}_{\leq n\, R.B}$ if u is not a root individual, so such an equality can be eagerly simplified into $s \approx t$. The *NI*-rule, however, must be applied to $s \approx t \,@^{u}_{\leq n\, R.B}$ even if $s = t$; hence, such an equality must be derived even though it is a logical tautology. Finally, if $\mathcal{C}$ has been obtained by a translation of a DL knowledge base that does not use nominals, inverse roles, or number restrictions, then the precondition of the *NI*-rule will never be satisfied, so we need not keep track of annotations at all.

Theorem 1. *Checking whether a* $\mathcal{SHOIQ}$ *knowledge base* $\mathcal{K}$ *is satisfiable can be performed by computing* $\mathcal{K}' = \Delta(\Omega(\mathcal{K}))$ *and then checking whether some derivation for* $\Xi(\mathcal{K}')$ *contains a leaf node labeled with a clash-free ABox. Furthermore, such an algorithm can be implemented in* 2NExpTime *in* $|\mathcal{K}|$*.*

5 Conclusion

In this paper, we analyzed the problems arising in tableau calculi due to an interaction between nominals, inverse roles, and number restrictions. We also presented a new hypertableau calculus for $\mathcal{SHOIQ}$, which we implemented in our reasoner HermiT. As we report in [7], our reasoner seems to perform well on many practical problems.

Table 1: Derivation Rules of the Tableau Calculus

Hyp-rule	If 1. $U_1 \wedge \ldots \wedge U_m \rightarrow V_1 \vee \ldots \vee V_n \in \mathcal{C}$, and 2. a mapping σ of variables to the individuals of $\mathcal{A}$ exists such that 2.1 $\sigma(x)$ is not indirectly blocked for each variable $x \in N_V$, 2.2 $\sigma(U_i) \in \mathcal{A}$ for each $1 \leq i \leq m$, and 2.3 $\sigma(V_j) \notin \mathcal{A}$ for each $1 \leq j \leq n$, then $\mathcal{A}_1 = \mathcal{A} \cup \{\bot\}$ if $n = 0$; $\mathcal{A}_j := \mathcal{A} \cup \{\sigma(V_j)\}$ for $1 \leq j \leq n$ otherwise.
$\geq$-rule	If 1. $\geq n\, R.C(s) \in \mathcal{A}$, 2. s is not blocked in $\mathcal{A}$, and 3. $\mathcal{A}$ does not contain individuals $u_1, \ldots, u_n$ such that 3.1 $\{\mathsf{ar}(R, s, u_i), C(u_i) \mid 1 \leq i \leq n\} \cup \{u_i \not\approx u_j \mid 1 \leq i < j \leq n\} \subseteq \mathcal{A}$, and 3.2 either s is blockable or no u_i, $1 \leq i \leq n$, is indirectly blocked in $\mathcal{A}$ then $\mathcal{A}_1 := \mathcal{A} \cup \{\mathsf{ar}(R, s, t_i),\ C(t_i) \mid 1 \leq i \leq n\} \cup \{t_i \not\approx t_j \mid 1 \leq i < j \leq n\}$ where $t_1, \ldots, t_n$ are fresh distinct successors of s.
$\approx$-rule	If 1. $s \approx t \in \mathcal{A}$ (the equality can possibly be annotated), and 2. $s \neq t$ then $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(s \rightarrow t)$ if t is a named individual, or t is a root individual and s is not a named individual, or s is a descendant of t; $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(t \rightarrow s)$ otherwise.
$\bot$-rule	If $s \not\approx s \in \mathcal{A}$ or $\{A(s), \neg A(s)\} \subseteq \mathcal{A}$ then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$.
NI-rule	If 1. $s \approx t\, @^u_{\leq n\, R.B} \in \mathcal{A}$ or $t \approx s\, @^u_{\leq n\, R.B} \in \mathcal{A}$, 2. u is a root individual, 3. s is a blockable nonsuccessor of u, and 4. t is a blockable individual then $\mathcal{A}_i := \mathsf{merge}_{\mathcal{A}}(s \rightarrow \mathsf{can}_{\mathcal{A}}(u.\langle R, B, i\rangle))$ for each $1 \leq i \leq n$.

References

1. V. Haarslev and R. Möller. RACER System Description. In *Proc. IJCAR 2001*, pages 701–706, Siena, Italy, June 18–23 2001.
2. I. Horrocks and U. Sattler. Ontology Reasoning in the $\mathcal{SHOQ}(\mathbf{D})$ Description Logic. In *Proc. IJCAI 2001*, pages 199–204, 2001.
3. I. Horrocks and U. Sattler. A Tableaux Decision Procedure for $\mathcal{SHOIQ}$. In *Proc. IJCAI 2005*, pages 448–453, Edinburgh, UK, July 30–August 5 2005.
4. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$. In *Proc. CADE-17*, pages 482–496, Pittsburgh, USA, 2000.
5. B. Motik, R. Shearer, and I. Horrocks. A Hypertableau Calculus for SHIQ. In *Proc. of the 2007 Description Logic Workshop (DL 2007)*, pages 419–426, 2007.
6. B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In *Proc. CADE-21*, volume 4603 of *LNAI*, pages 67–83, 2007.
7. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. Technical report, University of Oxford, 2008. Submitted to an international journal. `http://web.comlab.ox.ac.uk/oucl/work/rob.shearer/HtShoiq.pdf`.
8. B. Parsia and E. Sirin. Pellet: An OWL-DL Reasoner. Poster, In Proc. ISWC 2004, Hiroshima, Japan, November 7–11, 2004.
9. D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proc. IJCAR 2006*, pages 292–297, Seattle, WA, USA, 2006.