# Bridging the Gap between Tableau and Consequence-Based Reasoning

Yevgeny Kazakov and Pavel Klinov

The University of Ulm, Germany
{yevgeny.kazakov, pavel.klinov}@uni-ulm.de

**Abstract.** We present a non-deterministic consequence-based procedure for the description logic $\mathcal{ALCHI}$. Just like the similar style (deterministic) procedures for $\mathcal{EL}$ and Horn-$\mathcal{SHIQ}$, our procedure explicitly derives subsumptions between concepts, but due to non-deterministic rules, not all of these subsumptions are consequences of the ontology. Instead, the consequences are only those subsumptions that can be derived regardless of the choices made in the application of the rules. This is similar to tableau-based procedures, for which an ontology is inconsistent if every expansion of the tableau eventually results in a clash. We report on a preliminary experimental evaluation of the procedure using a version of SNOMED CT with disjunctions, which demonstrates some promising potential.

## 1 Introduction and Motivation

Consequence-based and tableau-based methods are two well-known approaches to reasoning in Description Logics (DLs). The former work by deriving logical consequences of axioms in the ontology using inference rules while the latter by building countermodels for conjectures. Historically, consequence-based methods have been applied to Horn DLs, most prominently, the $\mathcal{EL}$ family [1, 10] and Horn-$\mathcal{SHIQ}$ [6], for which they are more efficient than tableau, both theoretically and empirically. Tableau-based methods, in turn, continued to dominate for non-Horn logics and very expressive fragments of the DL family, e.g., $\mathcal{SROIQ}$. This was mostly because, first, reasoning in expressive logics requires various forms of case analysis, for which designing complete goal-directed inference rule systems has not been easy (but tableau can handle cases via backtracking), and second, the tableau algorithms have been deemed easier to extend to new constructors, because they follow more closely their semantics.

The most straightforward attempt to combine the best from both worlds is to implement both a consequence-based and a tableau-based procedure and use one or another depending on the ontology. Indeed, several tableau reasoners, including Pellet and Konclude, switch to a consequence-based algorithm for $\mathcal{EL}$ inputs. A more advanced approach, is to automatically split the ontology into parts and use a consequence-based algorithm for the deterministic part (which often covers nearly all of the ontology) and tableau for the rest. Such reasoners as MORe [11] and Chainsaw [14] use ontology modularity techniques for such splitting and then combine the results.

Another line of research is to extend consequence-based calculi to deal with non-determinism. One notable approach has been developed for ConDOR, a reasoner for full $\mathcal{ALCH}$ [13]. The ConDOR's procedure deals with disjunctions using deterministic inference rules akin to ordered resolution [3]. The calculus is complete, retains the optimal worst-case complexity, and still enables the "one pass" classification. It shows impressive performance speed-up over the existing (hyper)tableau-based reasoners [13].

In this paper we present another consequence-based procedure for a non-Horn logic, this time for $\mathcal{ALCHI}$, but instead of resolution we incorporate some of the tableau features into our inference rule system. Specifically, we present non-deterministic rules with alternative conclusions to deal with non-determinism (instead of deriving disjunctions) and prove soundness, completeness, and termination of the corresponding saturation procedure. We show that our procedure is another step towards combining the best from the consequence-based and tableau-based worlds: It retains the optimal worst-case complexity, goal directedness, and the "granularity" property of consequence-based reasoning (i.e. computation of subsumers for different concepts can be done with limited interaction which enables concurrent [9] and incremental reasoning [7]). At the same time, the non-deterministic rules enable tableau-style backtracking to be used during consequence-based reasoning which has advantages over the resolution-style procedure, for example, can re-use memory instead of storing all derived disjunctions.

Finally, we present results of a preliminary evaluation using Horn ontologies and the same disjunctive extension of the anatomical part of SNOMED CT which was used to evaluate ConDOR (SCT-SEP) [13]. The results demonstrate that the procedure improves over ConDORs on SCT-SEP, is comparable with ELK [10] on large $\mathcal{EL}^+$ ontologies, and several times faster than the fastest available tableau reasoner Konclude [5].

## 2 Preliminaries

The syntax of $\mathcal{ALCHI}$ is defined using a vocabulary consisting of countably infinite sets of *atomic roles* and *atomic concepts*. We use the letters $R, S$ for roles, $C, D$ for concepts, and $A, B$ for atomic concepts. Complex *concepts* are defined by the grammar

$$C_{(i)} ::= \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists R.C \mid \forall R.C,$$

where $R$ is either an atomic role or its inverse (written $R^-$). Semantics is defined in the standard way; formal details can be found in the technical report [8].

An $\mathcal{ALCHI}$ *ontology* is a finite set of *concept inclusion* and *role inclusion* axioms of the form $C \sqsubseteq D$ and $R \sqsubseteq S$, respectively. The *ontology classification task* requires to compute all entailed subsumptions between atomic concepts occurring in $\mathcal{O}$.

An $\mathcal{ALCHI}$ ontology $\mathcal{O}$ is *Horn* if for every axiom $C \sqsubseteq D$, the concepts $C$ and $D$ satisfy, respectively, the following grammar definitions:

$$C_{(i)} ::= \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C,$$
$$D_{(i)} ::= \top \mid \bot \mid A \mid D_1 \sqcap D_2 \mid \exists R.D \mid \forall R.D \mid \neg C.$$

That is, negations and universal restrictions should not occur on the left-hand side of concept inclusions, and disjunctions should not occur on the right-hand side.

A consequence-based procedure for Horn-$\mathcal{ALCHI}$ can be presented using the rules in Fig. 1. These rules are similar to the rules for Horn-$\mathcal{SHIQ}$ [6]; the main difference

$$\mathsf{R_0}\ \frac{}{H \sqsubseteq C} : H = K \sqcap C \qquad \mathsf{R}^+_\sqcap\ \frac{H \sqsubseteq C_1 \quad H \sqsubseteq C_2}{H \sqsubseteq C_1 \sqcap C_2} : C_1 \sqcap C_2 \in \mathrm{sub}(\mathcal{O})$$

$$\mathsf{R}^+_\top\ \frac{}{H \sqsubseteq \top} : \top \in \mathrm{sub}(\mathcal{O}) \qquad \mathsf{R}^+_\sqcup\ \frac{H \sqsubseteq C_1 \mid H \sqsubseteq C_2}{H \sqsubseteq C_1 \sqcup C_2} : C_1 \sqcup C_2 \in \mathrm{sub}(\mathcal{O})$$

$$\mathsf{R}_\sqsubseteq\ \frac{H \sqsubseteq C}{H \sqsubseteq D} : C \sqsubseteq D \in \mathcal{O} \qquad \mathsf{R}^\leftarrow_\forall\ \frac{H \sqsubseteq \exists R.K \quad K \sqsubseteq \forall S.C}{H \sqsubseteq C} : R \sqsubseteq^*_\mathcal{O} S^-$$

$$\mathsf{R}^-_\sqcap\ \frac{H \sqsubseteq C_1 \sqcap C_2}{H \sqsubseteq C_1 \quad H \sqsubseteq C_2} \qquad \mathsf{R}^\leftarrow_\exists\ \frac{H \sqsubseteq \exists R.K \quad K \sqsubseteq C}{H \sqsubseteq \exists S.C} : \begin{array}{l} \exists S.C \in \mathrm{sub}(\mathcal{O}) \\ R \sqsubseteq^*_\mathcal{O} S \end{array}$$

$$\mathsf{R}^-_\neg\ \frac{H \sqsubseteq C \quad H \sqsubseteq \neg C}{H \sqsubseteq \bot} \qquad \mathsf{R}^\rightarrow_\forall\ \frac{H \sqsubseteq \exists R.K \quad H \sqsubseteq \forall S.C}{H \sqsubseteq \exists R.(K \sqcap C)} : R \sqsubseteq^*_\mathcal{O} S$$

$$\mathsf{R}^\leftarrow_\bot\ \frac{H \sqsubseteq \exists R.K \quad K \sqsubseteq \bot}{H \sqsubseteq \bot} \qquad \mathsf{R}^\rightarrow_\exists\ \frac{H \sqsubseteq \exists R.K \quad H \sqsubseteq C}{H \sqsubseteq \exists R.(K \sqcap \exists S.C)} : \begin{array}{l} \exists S.C \in \mathrm{sub}(\mathcal{O}) \\ R \sqsubseteq^*_\mathcal{O} S^- \end{array}$$

**Fig. 1.** The consequence-based rules for reasoning in Horn-$\mathcal{ALCHI}$

is that they can operate directly on complex concepts without requiring the ontology to be normalized. Here we denote by $\mathrm{sub}(\mathcal{O})$ the set of concepts occurring in the ontology $\mathcal{O}$ (possibly as subconcepts). The rules derive *subsumptions* of the form $H \sqsubseteq U$ with

$$U ::= \bot \mid C \mid \exists R.K, \tag{1}$$

$$H, K ::= \textstyle\bigsqcap_{i \in I} C_i, \tag{2}$$

where $C_i \in \mathrm{sub}(\mathcal{O})$ $(i \in I)$ and $C \in \mathrm{sub}(\mathcal{O})$. The premises of each rule (if any) are given above the horizontal line, the conclusion below, and the *side conditions* that determine when each rule is applicable (in terms of $\mathcal{O}$) after the colon.[1] Note that the concept inclusion axioms in $\mathcal{O}$ are used not as premises of the rules but as side condition of $\mathsf{R}_\sqsubseteq$. The side conditions of the rules $\mathsf{R}^\leftarrow_\forall$, $\mathsf{R}^\leftarrow_\exists$, $\mathsf{R}^\rightarrow_\forall$, and $\mathsf{R}^\rightarrow_\exists$ use the precomputed role hierarchy $\sqsubseteq^*_\mathcal{O}$, which is the smallest reflexive transitive relation on roles such that $R \sqsubseteq S \in \mathcal{O}$ implies $R \sqsubseteq^*_\mathcal{O} S$ and $R^- \sqsubseteq^*_\mathcal{O} S^-$. It can be shown (and follows from the more general result of this paper) that the inference system in Fig. 1 is *sound* and *complete* for deriving all entailed subsumptions with concepts occurring in the ontology, namely that for every $C \in \mathrm{sub}(\mathcal{O})$ and every $H$ of the form (2), we have $\mathcal{O} \models H \sqsubseteq C$ iff either $H \sqsubseteq \bot$ or $H \sqsubseteq C$ is derivable by the rules in Fig. 1 for $\mathcal{O}$.

## 3 Non-Deterministic Consequence-Based Reasoning in $\mathcal{ALCHI}$

Extending consequence-based procedures beyond Horn DLs is difficult because dealing with non-deterministic constructors, such as disjunction, requires reasoning by case. In tableau-style procedures which construct (representations of) models, dealing with non-deterministic constructors is straightforward: if a domain element is an instance of $B \sqcup C$ then it should be an instance of $B$ or $C$. In consequence-based procedures that derive entailed subsumptions directly, this kind of principle is not valid: if $A \sqsubseteq B \sqcup C$ is entailed,

[1]Rule $\mathsf{R}^+_\sqcup$ stands for two rules with single premises $H \sqsubseteq C_1$ and $H \sqsubseteq C_2$ respectively.

then it is not true that either $A \sqsubseteq B$ or $A \sqsubseteq C$ is entailed. The consequence-based procedure for (non-Horn) $\mathcal{ALCH}$ [13] solves this problem by retaining disjunctions and recombining them using resolution-style inference rules. For example, if ontology $\mathcal{O}$ contains $B \sqsubseteq D$ and $C \sqsubseteq D$, then from $A \sqsubseteq B \sqcup C$ using $B \sqsubseteq D \in \mathcal{O}$, one can derive $A \sqsubseteq D \sqcup C$, from which similarly "resolving" on $C \sqsubseteq D \in \mathcal{O}$ one obtains $A \sqsubseteq D \sqcup D$, which is factorized to $A \sqsubseteq D$ by merging duplicate disjuncts.

One of the well-known disadvantages of resolution-based procedures compared to search-based (DPLL-style) procedures, is that resolution can produce many long disjunctions which can occupy a considerable amount of space and make inferences slow because merging disjunctions is not a trivial operation. In this paper we, thus, reconsider the former idea of splitting disjunctions using a non-deterministic rule:

$$\mathbf{R}_{\sqcup}^{-}\ \frac{H \sqsubseteq C_1 \sqcup C_2}{H \sqsubseteq C_1 \quad \mid \quad H \sqsubseteq C_2} \tag{3}$$

Just like in tableau-based procedures, this rule creates two branches in which subsumptions can be independently derived. As discussed above, it is not true that all subsumptions derived in one of the branches will be entailed by the ontology. However, it can be the case for the subsumptions that are derived *on every branch*, i.e., regardless of the non-deterministic choices made. For example, if we split subsumption $A \sqsubseteq B \sqcup C$ on $A \sqsubseteq B$ and $A \sqsubseteq C$, then in both branches we can derive $A \sqsubseteq D$ using axioms $B \sqsubseteq D \in \mathcal{O}$ and $C \sqsubseteq D \in \mathcal{O}$ respectively. This idea of considering common conclusions in all non-deterministic branches is somewhat reminiscent of the Stålmarck's procedure for propositional logic (see, e.g., [12]). Unfortunately, adding just the rule $\mathbf{R}_{\sqcup}^{-}$ to the system in Fig. 1 is not sufficient for obtaining a sound inference system:

*Example 1.* Consider the ontology $\mathcal{O}$ containing the following axioms:

$A \sqsubseteq \exists R.A, \quad A \sqsubseteq B \sqcup C, \quad \exists S.C \sqsubseteq B, \quad R \sqsubseteq S.$

Then the subsumers of $A$ can computed using two branches after the splitting rule:

$$\begin{array}{ccl}
 & A \sqsubseteq A & \text{by } \mathbf{R_0}(), \\
 & \mid & \\
 & A \sqsubseteq \exists R.A & \text{by } \mathbf{R}_{\sqsubseteq}(A \sqsubseteq A)\text{: } A \sqsubseteq \exists R.A \in \mathcal{O}, \\
 & \mid & \\
 & A \sqsubseteq B \sqcup C & \text{by } \mathbf{R}_{\sqsubseteq}(A \sqsubseteq A)\text{: } A \sqsubseteq B \sqcup C \in \mathcal{O}, \\
 & \swarrow \quad \searrow & \\
A \sqsubseteq B & A \sqsubseteq C & \text{by } \mathbf{R}_{\sqcup}^{-}(A \sqsubseteq B \sqcup C), \\
 & \mid & \\
 & A \sqsubseteq \exists S.C & \text{by } \mathbf{R}_{\exists}^{\leftarrow}(A \sqsubseteq \exists R.A,\, A \sqsubseteq C)\text{: } R \sqsubseteq_{\mathcal{O}}^{*} S, \\
 & \mid & \\
 & A \sqsubseteq B & \text{by } \mathbf{R}_{\sqsubseteq}(A \sqsubseteq \exists S.C)\text{: } \exists S.C \sqsubseteq B \in \mathcal{O}.
\end{array}$$

As can be seen, $A \sqsubseteq B$ can be derived in both of these branches, but $\mathcal{O} \not\models A \sqsubseteq B$.

To understand the problem in Example 1, we first need to understand how to interpret rules like (3). Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we say that an element $a \in \Delta^{\mathcal{I}}$ *satisfies* a subsumption $C \sqsubseteq D$ in $\mathcal{I}$ (or $C \sqsubseteq D$ *is satisfied* in $\mathcal{I}$ by $a$) if $a \in C^{\mathcal{I}}$ implies $a \in D^{\mathcal{I}}$. So, rule $\mathbf{R}_{\sqcup}^{-}$ can be understood as follows: for each $a \in \Delta^{\mathcal{I}}$ if $a$ satisfies the premise $H \sqsubseteq C_1 \sqcup C_2$ then it satisfies one of the conclusions $H \sqsubseteq C_1$ or $H \sqsubseteq C_2$. Clearly, this is true for this rule (and every $\mathcal{I}$), however, this property fails for rules $\mathsf{R}^{\leftarrow}_{\bot}$, $\mathsf{R}^{\leftarrow}_{\forall}$, and $\mathsf{R}^{\leftarrow}_{\exists}$ in Fig. 1, even if $\mathcal{I} \models \mathcal{O}$. For a counter-example, take $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with $\Delta^{\mathcal{I}} = A^{\mathcal{I}} = \{a, b\}$, $B^{\mathcal{I}} = \{b\}$, $C^{\mathcal{I}} = \{a\}$, and $S^{\mathcal{I}} = R^{\mathcal{I}} = \{\langle a, b\rangle, \langle b, a\rangle\}$. Clearly, $\mathcal{I}$ is a model of $\mathcal{O}$ in Example 1, and $a$ satisfies both $A \sqsubseteq \exists R.A$ and $A \sqsubseteq C$, but, $a$ does not satisfy the conclusion $A \sqsubseteq \exists S.C$ obtained from these premises by $\mathsf{R}^{\leftarrow}_{\exists}$. Hence, the inference by $\mathsf{R}^{\leftarrow}_{\exists}$ in Example 1 is unsound.

To "fix" rule $\mathsf{R}^{\leftarrow}_{\exists}$, we use a similar observation as in ConDOR [13]. Specifically, if $a$ satisfies $H \sqsubseteq \exists R.K$ in $\mathcal{I}$ but does not satisfy $H \sqsubseteq \exists S.C$, and $\mathcal{I} \models R \sqsubseteq S$, then $a$ should satisfy $H \sqsubseteq \exists R.(K \sqcap \neg C)$. Thus, we can generalize rule $\mathsf{R}^{\leftarrow}_{\exists}$ as follows:

$$\mathsf{R}^{\leftarrow}_{\exists}\ \frac{H \sqsubseteq \exists R.K \quad K \sqsubseteq C}{H \sqsubseteq \exists S.C \ \mid \ H \sqsubseteq \exists R.(K \sqcap \neg C)} : \begin{array}{l} \exists S.C \in \mathrm{sub}(\mathcal{O}) \\ R \sqsubseteq^{*}_{\mathcal{O}} S \end{array} \tag{4}$$

*Example 2.* Consider the ontology $\mathcal{O}$ from Example 1. Using the updated rule (4), we can now obtain the following inferences for $A$:

$$\begin{array}{cccl}
 & A \sqsubseteq A & & \text{by } \mathsf{R_0}(), \\
 & \mid & & \\
 & A \sqsubseteq \exists R.A & & \text{by } \mathsf{R}_{\sqsubseteq}(A \sqsubseteq A)\text{: } A \sqsubseteq \exists R.A \in \mathcal{O}, \\
 & \mid & & \\
 & A \sqsubseteq B \sqcup C & & \text{by } \mathsf{R}_{\sqsubseteq}(A \sqsubseteq A)\text{: } A \sqsubseteq B \sqcup C \in \mathcal{O}, \\
 / & & \backslash & \\
A \sqsubseteq B & & A \sqsubseteq C & \text{by } \mathsf{R}^{-}_{\sqcup}(A \sqsubseteq B \sqcup C, A \sqsubseteq \neg B), \\
 & / & \backslash & \\
 & A \sqsubseteq \exists S.C & A \sqsubseteq \exists R.(A \sqcap \neg C) & \text{by } \mathsf{R}^{\leftarrow}_{\exists}(A \sqsubseteq \exists R.A, A \sqsubseteq C). \\
 & \mid & & \\
 & A \sqsubseteq B & &
\end{array}$$

Note that in the rightmost branch, we derive the subsumption $A \sqsubseteq \exists R.(A \sqcap \neg C)$ to which rule $\mathsf{R}^{\leftarrow}_{\exists}$ can potentially be applied. In order to check if this rule is applicable, we need to derive subsumptions for $A \sqcap \neg C$:

$$\begin{array}{cccl}
 & A \sqcap \neg C \sqsubseteq A & & \text{by } \mathsf{R_0}(), \\
 & \mid & & \\
 & A \sqcap \neg C \sqsubseteq \exists R.A & & \text{by } \mathsf{R}_{\sqsubseteq}(A \sqcap \neg C \sqsubseteq A)\text{: } A \sqsubseteq \exists R.A \in \mathcal{O}, \\
 & \mid & & \\
 & A \sqcap \neg C \sqsubseteq B \sqcup C & & \text{by } \mathsf{R}_{\sqsubseteq}(A \sqcap \neg C \sqsubseteq A)\text{: } A \sqsubseteq B \sqcup C \in \mathcal{O}, \\
 / & & \backslash & \\
A \sqcap \neg C \sqsubseteq B & & A \sqcap \neg C \sqsubseteq C & \text{by } \mathsf{R}^{-}_{\sqcup}(A \sqcap \neg C \sqsubseteq B \sqcup C), \\
 & & \mid & \\
 & & A \sqcap \neg C \sqsubseteq \bot & \text{by } \mathsf{R}_{\bot}(A \sqcap \neg C \sqsubseteq C).
\end{array}$$

If we now consider the rightmost branch in the derivation for $A$ and the leftmost branch for $A \sqcap \neg C$ we obtain a set of subsumptions for which all inferences are applied but neither $A \sqsubseteq B$ nor $A \sqsubseteq \bot$ is derived. Hence, we can conclude that $\mathcal{O} \not\models A \sqsubseteq B$.

Figure 2 presents a generalization and extension of the rules in Fig. 1 to (full) $\mathcal{ALCHI}$ using the idea similar to rule (4). The new rules now derive subsumptions

$$
\begin{array}{ll}
\mathsf{R_0}\ \dfrac{}{H \sqsubseteq C} : H = K \sqcap C & \mathsf{R}^+_\sqcap\ \dfrac{H \sqsubseteq C_1 \quad H \sqsubseteq C_2}{H \sqsubseteq C_1 \sqcap C_2} : C_1 \sqcap C_2 \in \mathrm{sub}(\mathcal{O}) \\[2ex]
\mathsf{R}_\bot\ \dfrac{H \sqsubseteq C}{H \sqsubseteq \bot} : H = K \sqcap \neg C & \mathsf{R}^+_\sqcup\ \dfrac{H \sqsubseteq C_1 \mid H \sqsubseteq C_2}{H \sqsubseteq C_1 \sqcup C_2} : C_1 \sqcup C_2 \in \mathrm{sub}(\mathcal{O}) \\[2ex]
\mathsf{R}^+_\top\ \dfrac{}{H \sqsubseteq \top} : \top \in \mathrm{sub}(\mathcal{O}) & \mathsf{R}^+_\neg\ \dfrac{}{H \sqsubseteq \neg C \mid H \sqsubseteq C} : \neg C \in \mathrm{sub}(\mathcal{O}) \\[2ex]
\mathsf{R}_\sqsubseteq\ \dfrac{H \sqsubseteq C}{H \sqsubseteq D} : C \sqsubseteq D \in \mathcal{O} & \mathsf{R}^+_\forall\ \dfrac{}{H \sqsubseteq \forall R.C \mid H \sqsubseteq \exists R.\neg C} : \forall R.C \in \mathrm{sub}(\mathcal{O}) \\[2ex]
\mathsf{R}^-_\sqcap\ \dfrac{H \sqsubseteq C_1 \sqcap C_2}{H \sqsubseteq C_1 \quad H \sqsubseteq C_2} & \mathsf{R}^\leftarrow_\forall\ \dfrac{H \sqsubseteq \exists R.K \quad K \sqsubseteq \forall S.C}{H \sqsubseteq C \mid H \sqsubseteq \exists R.(K \sqcap \neg \forall S.C)} : R \sqsubseteq^*_\mathcal{O} S^- \\[2ex]
\mathsf{R}^-_\sqcup\ \dfrac{H \sqsubseteq C_1 \sqcup C_2}{H \sqsubseteq C_1 \mid H \sqsubseteq C_2} & \mathsf{R}^\leftarrow_\exists\ \dfrac{H \sqsubseteq \exists R.K \quad K \sqsubseteq C}{H \sqsubseteq \exists S.C \mid H \sqsubseteq \exists R.(K \sqcap \neg C)} : \begin{array}{l} \exists S.C \in \mathrm{sub}(\mathcal{O}) \\ R \sqsubseteq^*_\mathcal{O} S \end{array} \\[2ex]
\mathsf{R}^-_\neg\ \dfrac{H \sqsubseteq C \quad H \sqsubseteq \neg C}{H \sqsubseteq \bot} & \mathsf{R}^\rightarrow_\forall\ \dfrac{H \sqsubseteq \exists R.K \quad H \sqsubseteq \forall S.C}{H \sqsubseteq \exists R.(K \sqcap C)} : R \sqsubseteq^*_\mathcal{O} S \\[2ex]
\mathsf{R}^\leftarrow_\bot\ \dfrac{H \sqsubseteq \exists R.K \quad K \sqsubseteq \bot}{H \sqsubseteq \bot} & \mathsf{R}^\rightarrow_\exists\ \dfrac{H \sqsubseteq \exists R.K \quad H \sqsubseteq C}{H \sqsubseteq \exists R.(K \sqcap \exists S.C)} : \begin{array}{l} \exists S.C \in \mathrm{sub}(\mathcal{O}) \\ R \sqsubseteq^*_\mathcal{O} S^- \end{array}
\end{array}
$$

**Fig. 2.** The non-deterministic consequence-based rules for reasoning in $\mathcal{ALCHI}$

of the form $H \sqsubseteq U$, where $U$ is of the form (1), and

$$
H, K := \bigsqcap_{i \in I} C_i \sqcap \bigsqcap_{j \in J} \neg D_j, \tag{5}
$$

where $C_i \in \mathrm{sub}(\mathcal{O})$ $(i \in I)$, and $D_j \in \mathrm{sub}(\mathcal{O})$ $(j \in J)$. Note that the rule $\mathsf{R}^\leftarrow_\bot$ has remained the same, and is still formally unsound under our new interpretation. This rule will be dealt with by our procedure in a special way, which does not affect soundness.

### 3.1 The Rule Application Strategy

Just like for other consequence-based procedures [1, 6, 13], every set closed under the rules in Fig. 2 contains all "relevant" subsumptions entailed by the ontology, that is, our rules are *complete*. Specifically, we say that a set $M$ of subsumptions is *closed* under a rule $\mathsf{R}$ if, whenever all premises of $\mathsf{R}$ belong to $M$, all its conclusions (respectively one of the conclusions if $\mathsf{R}$ is non-deterministic) belong to $M$.

**Theorem 1 (Completeness).** *Let $\mathcal{O}$ be an $\mathcal{ALCHI}$ ontology, $H$ a conjunction of form* (5)*, $D$ a concept occurring in $\mathcal{O}$ such that $\mathcal{O} \models H \sqsubseteq D$, and $M$ a set of subsumptions closed under all rules in Fig. 2. Then either $H \sqsubseteq D \in M$ or $H \sqsubseteq \bot \in M$.*

The proof of Theorem 1 (cf. [8]) works along the same lines as for $\mathcal{EL}$ [1] and Horn-$\mathcal{SHIQ}$ [6], i.e., by constructing a canonical model of $\mathcal{O}$ from $M$.

*Remark 1.* Just like for $\mathcal{EL}$, if the set of *goal* subsumptions $H \sqsubseteq D$ for which entailment should be checked is known in advance (e.g., for classification), it is possible to sharpen Theorem 1 by relaxing the requirement on the closure of $M$ [10]. Specifically, it is necessary to produce $H \sqsubseteq U$ by rules only if there is a goal subsumption $H \sqsubseteq D$ for some $D$, or if $K \sqsubseteq \exists R.H$ has been derived for some $K$ earlier. Furthermore, the rules $\mathsf{R}^{+}_{\top}$, $\mathsf{R}^{+}_{\sqcap}$, $\mathsf{R}^{+}_{\sqcap}$, $\mathsf{R}^{+}_{\neg}$, $\mathsf{R}^{+}_{\forall}$, and $\mathsf{R}^{\leftarrow}_{\exists}$ introducing new concepts in $\mathrm{sub}(\mathcal{O})$ (see the side conditions), need to be applied only if the new concept is a subconcept of the left-hand side of some concept inclusion in $\mathcal{O}$ or of some $D$ in the right-hand side of goal subsumptions. Both optimizations can be particularly useful to mitigate "blind guessing" in rules $\mathsf{R}^{+}_{\neg}$ and $\mathsf{R}^{+}_{\forall}$, which do not have any premises, and thus apply to every $H$.

It is possible to prove the converse of Theorem 1, namely that if $\mathcal{O} \not\models H \sqsubseteq D$, then there exists a set of subsumptions $M$ closed under the rules in Fig. 2 that contains neither $H \sqsubseteq D$ nor $H \sqsubseteq \bot$. This result, however, is not very useful in practice because in order to check whether $\mathcal{O} \models H \sqsubseteq D$ using this result, one would need to enumerate all $M$ closed under the rules in Fig. 2. Just like tableau-based procedures do not need to enumerate all completion graphs, we do not need to enumerate all such closed sets $M$. In particular, subsumptions derived for different conjunctions $H$ of the form (5) can be in most cases considered independently.

The following *concept satisfiability procedure* (short CSAT) checks satisfiability of conjunctions $H$ of the form (5) using the rules in Fig. 2:

1. We apply the inference rules in Fig. 2 (introducing conjunctions $H$ as necessary according to Remark 1) and record every derived subsumptions $H \sqsubseteq U$ in a separate *local branch* for each $H$, just like in Example 2.
2. If a subsumption $H \sqsubseteq U$ was derived using a non-deterministic rule, we also remember the alternative conclusion of this rule in a *branching point*.
3. If we derive a *local clash* $H \sqsubseteq \bot$, we remove all conclusions on the branch for $H$ starting from the last non-deterministically derived conclusion, and produce the alternative conclusion, which we now consider as *deterministically derived*.
4. If local clash $H \sqsubseteq \bot$ is derived and there are no non-deterministically derived conclusions on the branch for $H$ anymore, we mark $H$ as *inconsistent*.
5. Rule $\mathsf{R}^{\leftarrow}_{\bot}$ is applied only if for the right premise of the rule $K \sqsubseteq \bot$ (local clash in the branch for $K$), the conjunction $K$ is marked as inconsistent.
6. The procedure terminates when no more rules can be applied and for each derived local clash $H \sqsubseteq \bot$ (on the branch for $H$), $H$ is marked as inconsistent.

Note that even though we apply rule $\mathsf{R}^{\leftarrow}_{\bot}$ in a restricted way, because of the last condition, the resulting set $M$ of subsumptions will be closed under all rules in Fig. 2. Also note that if some conclusion $H \sqsubseteq U$ was derived using a premise $K \sqsubseteq V$ with $K \neq H$ (using rules $\mathsf{R}^{\leftarrow}_{\bot}$, $\mathsf{R}^{\leftarrow}_{\forall}$ or $\mathsf{R}^{\leftarrow}_{\exists}$), we do not remove $K \sqsubseteq V$ when we backtrack $H \sqsubseteq U$ as we remove only conclusion on the local branch for $H$. Likewise, if we remove the premise $K \sqsubseteq V$, we do not need to remove the conclusion $H \sqsubseteq U$ derived from this premise. In other words, premises from other local branches can never result in inconsistency of $H$, and so, deriving and backtracking of conclusions can be performed for each local branch independently.

By Theorem 1, each $H$ that is not marked as inconsistent by our procedure must be satisfiable since $H \sqsubseteq \bot \notin M$ for the resulting closure $M$. We will also prove the converse, namely that only unsatisfiable $H$ can be marked as inconsistent by CSAT.

The rules in Fig. 2 can be used not only for checking satisfiability of $H$, but also for computing its entailed subsumers. The following *branch exploration procedure* (short BEXP) continues after CSAT, and computes the set $S(H)$ of concepts $D$ such that $H \sqsubseteq D$ appear on every branch of $H$ that does not contain a local clash:

1. For each $H$ not marked as inconsistent, create a set $S(H)$ of *candidate subsumers* consisting of those $D \in \text{sub}(\mathcal{O})$ such that $H \sqsubseteq D$ is on the branch for $H$.
2. Remove all conclusions on the branch for $H$ starting from the last non-deterministic conclusion, and produce the alternative conclusion.
3. Apply the rules like in CSAT (this may extend other branches), backtracking if necessary, but do not mark $H$ as inconsistent when no backtracking in $H$ is possible.
4. After all rules are reapplied and there is no local clash in $H$, remove from $S(H)$ all $D$ for which $H \sqsubseteq D$ no longer occurs on the branch.
5. Repeat from Step 2 until no non-deterministic conclusions are left on the branch.

It follows from Theorem 1, that $S(H)$ contains all concepts $D \in \text{sub}(\mathcal{O})$ such that $\mathcal{O} \models H \sqsubseteq D$ since $D$ is removed from $S(H)$ only when we find $M$ closed under the rules in Fig. 2 such that $H \sqsubseteq D \notin M$. In the next section we prove the converse, namely that if $\mathcal{O} \not\models H \sqsubseteq D$ then there is a branch of $H$ that does not contain $H \sqsubseteq D$.

### 3.2 Soundness, Termination, and Complexity

To prove soundness of the procedures CSAT and BEXP described in the previous section, we need to formally define our non-deterministic derivation strategy.

**Definition 1 (Tableau, Local Tableau).** *A* tableau *(for $\mathcal{O}$) is a triple $T = (\boldsymbol{H}, \prec, t)$, where $\boldsymbol{H}$ is a set of conjunctions of the form* (5)*, $\prec$ is a partial order on $\boldsymbol{H}$, and $t$ is a function that assigns to every $H \in \boldsymbol{H}$ a tree $t(H)$ called the* local tableau for $H$ *whose nodes are labeled by subsumptions $H \sqsubseteq U$ (with this $H$), $U$ of the form* (1)*, such that:*

1. *For each node $n$ of $t(H)$, there is a rule in Fig. 2 for $\mathcal{O}$ such the label of $n$ is a conclusion of this rule and all premises of the form $H \sqsubseteq U$ (with this $H$) are labels of some ancestors of $n$ (the nodes on the path from the root of $t(H)$ to $n$ without $n$); if the inference is by $\mathsf{R}^{\leftarrow}_{\bot}$, the second premise $K \sqsubseteq \bot$ must be such that $K \prec H$.*
2. *If the label is obtained by a non-deterministic rule ($\mathsf{R}^{-}_{\sqcup}$, $\mathsf{R}^{+}_{\neg}$, $\mathsf{R}^{+}_{\forall}$, $\mathsf{R}^{\leftarrow}_{\forall}$, $\mathsf{R}^{\leftarrow}_{\exists}$), the node should have a sibling node labeled by the alternative conclusion of the rule.*
3. *All labels of the nodes on the same branch in $t(H)$ must be different.*
4. *If $K \prec H$ then all branches of $t(K)$ must contain the local clash $K \sqsubseteq \bot$.*

Intuitively, each local tableau $t(H)$ for $H$ describes the set of branches that are encountered for $H$ by procedure BEXP described in Sect. 3.1. For example, the two trees in Example 2 are local tableaux respectively for $H = A$ and $H = A \sqcap \neg C$. The partial order $\prec$ is the order in which the conjunctions $H$ are marked as inconsistent. According to the last condition, we set $K \prec H$ if $K$ is marked as inconsistent and $H$ is not. Thus, rule $\mathsf{R}^{\leftarrow}_{\bot}$ derives $H \sqsubseteq \bot$ from $K \sqsubseteq \bot$ by our procedure only if $K \prec H$.

**Theorem 2 (Soundness).** *Let $T = (\boldsymbol{H}, \prec, t)$ be a tableau for $\mathcal{O}$. Then for each $H \in \boldsymbol{H}$ and each concept $D$ (not necessarily occurring in $\mathcal{O}$) such that $\mathcal{O} \not\models H \sqsubseteq D$, there exists a branch in $t(H)$ that contains neither $H \sqsubseteq \bot$ nor $H \sqsubseteq D$.*

*Proof.* The proof is by induction on $\prec$. Assume that the claim holds for each $K \prec H$, and prove it for $H$. Since $\mathcal{O} \not\models H \sqsubseteq D$, there is a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{O}$ and $a \in \Delta^{\mathcal{I}}$ such that $a$ does not satisfy $H \sqsubseteq D$ in $\mathcal{I}$, i.e., $a \in H^{\mathcal{I}} \setminus D^{\mathcal{I}}$. We prove that there is a branch in $t(H)$ such that every label $H \sqsubseteq U$ on this branch is satisfied in $\mathcal{I}$ by $a$, that is $a \in U^{\mathcal{I}}$. This implies that neither $H \sqsubseteq \bot$ nor $H \sqsubseteq D$ occurs on this branch.

It is sufficient to show that for every node $n$ in $t(H)$, if $a$ satisfies the labels of all ancestors of $n$ then $a$ satisfies the label of $n$ or of its sibling node (if there is one). According to Definition 1, these labels are obtained by some rule in Fig. 2 in which all premises of the form $H \sqsubseteq U$ must be the labels of some ancestors of $n$. That is, all premises of the rule with the left-hand side $H$ are satisfied in $\mathcal{I}$ by $a$. We will show that in this case, the conclusion of the rule (respectively, one of the non-deterministic conclusions if the rule is non-deterministic) is also satisfied in $\mathcal{I}$ by $a$:

- Rules $\mathsf{R_0}$, $\mathsf{R}_\bot$, $\mathsf{R}_\top^+$, $\mathsf{R}_\sqsubseteq$, $\mathsf{R}_\sqcap^-$, $\mathsf{R}_\sqcap^-$, $\mathsf{R}_\neg^-$, $\mathsf{R}_\sqcap^+$, $\mathsf{R}_\sqcup^+$, $\mathsf{R}_\neg^+$, $\mathsf{R}_\forall^\rightarrow$, $\mathsf{R}_\exists^\rightarrow$: These are deterministic rules that contain only premises with the left-hand side $H$. It is easy to see that for all these rules the conclusion is a logical consequence of the premises and $\mathcal{O}$. Since $\mathcal{I} \models \mathcal{O}$, $a$ satisfies the conclusion because it satisfies all premises.
- Rules $\mathsf{R}_\sqcup^-$, $\mathsf{R}_\neg^+$, $\mathsf{R}_\forall^+$, $\mathsf{R}_\forall^\leftarrow$, $\mathsf{R}_\exists^\leftarrow$: These are non-deterministic rules. For each rule we show how to choose one of the conclusions that is satisfied in $a$.
  For $\mathsf{R}_\sqcup^-$, since $a \in (C_1 \sqcup C_2)^{\mathcal{I}}$, we have either $a \in {C_1}^{\mathcal{I}}$ or $a \in {C_2}^{\mathcal{I}}$. Hence $a$ satisfies either $H \sqsubseteq C_1$ or $H \sqsubseteq C_2$. The proofs for $\mathsf{R}_\neg^+$ and $\mathsf{R}_\forall^+$ are similar.
  For $\mathsf{R}_\forall^\leftarrow$, we have $a \in (\exists R.K)^{\mathcal{I}}$. That is, there exists $b \in K^{\mathcal{I}}$ such that $\langle a, b\rangle \in R^{\mathcal{I}}$. Since $R \sqsubseteq^*_{\mathcal{O}} S^-$, we have $\langle b, a\rangle \in S^{\mathcal{I}}$. If $a \in C^{\mathcal{I}}$, then $a$ satisfies the left conclusion $H \sqsubseteq C$. Otherwise, $a \notin C^{\mathcal{I}}$ and so $b \notin (\forall S.C)^{\mathcal{I}}$ since $\langle b, a\rangle \in S^{\mathcal{I}}$. Hence $b \in (K \sqcap \neg\forall S.C)^{\mathcal{I}}$ and $a \in \exists R.(K \sqcap \neg\forall S.C)$. So, $a$ satisfies the right conclusion $H \sqsubseteq \exists R.(K \sqcap \neg\forall S.C)$. The proof for $\mathsf{R}_\exists^\leftarrow$ is similar.
- Rule $\mathsf{R}_\bot^\leftarrow$: This is a deterministic rule, but for its right premise $K \sqsubseteq \bot$ we may have $K \neq H$. By Definition 1, however, we must have $K \prec H$, and so, every branch of $t(K)$ contains $K \sqsubseteq \bot$. Since $K \prec H$, by induction hypothesis, $\mathcal{O} \models K \sqsubseteq \bot$ (if $\mathcal{O} \not\models K \sqsubseteq \bot$ there would be a branch in $t(K)$ without $K \sqsubseteq \bot$). So, this case is not possible since $a$ must satisfy the premise $H \sqsubseteq \exists R.K$, but $a \notin (\exists R.K)^{\mathcal{I}} = \emptyset$. $\square$

*Remark 2.* Note that the second premises of rules $\mathsf{R}_\forall^\leftarrow$ and $\mathsf{R}_\exists^\leftarrow$ were not used in the proof of Theorem 2. That is, these premises do not have any impact on soundness. The purpose of these premises, is to restrict the non-deterministic "guessing" made by these rules to the cases that are sufficient for completeness. Note also that if the second premises of these rules are entailed by the ontology, e.g., they are derived deterministically, then the right conclusion is equivalent to $H \sqsubseteq \bot$, thus it cannot be satisfied by $a$ that does not satisfy $H \sqsubseteq D$. Hence the right conclusion is not necessary in this case, and so, these rules could be applied deterministically as in Fig. 1. This strategy makes sure that our procedure is deterministic when applied to Horn-$\mathcal{ALCHI}$ ontologies.

To combine Theorems 1 and 2 and obtain soundness and completeness for our procedure, we need to define when the tableau is fully expanded under inferences.

**Definition 2 (Fully Expanded Tableau).** *Let $T = (\boldsymbol{H}, \prec, t)$ be a tableau. A branch of $t(H)$ is* closed *if $H \sqsubseteq \bot$ occurs on this branch. Otherwise, the branch is called* open. *A* *branch is* fully expanded *if there exists a set $M$ of subsumptions closed under the rules in Fig. 2 such that all subsumptions of the form $H \sqsubseteq U$ in $M$ (with this $H$) occur on this branch. $T$ is fully expanded if every open branch of every $t(H)$ is fully expanded.*

Note that each time all rules are reapplied without producing a local clash at Step 4 of procedure BEXP in Sect. 3.1, we obtain a set $M$ closed under the rules (possibly different from the previous one). So, we "fully expand" every branch in our procedure. However, BEXP is far from enumerating all possible sets $M$ closed under the inference rules. E.g., we do not explore all possible combinations of branches in local tableaux.

**Corollary 1 (Correctness).** *Let $T = (\boldsymbol{H}, \prec, t)$ be a fully expanded tableau for $\mathcal{O}$. Then for every $H \in \boldsymbol{H}$ and every $D$ occurring in $\mathcal{O}$, we have $\mathcal{O} \models H \sqsubseteq D$ iff the subsumption $H \sqsubseteq D$ is derived in every open branch of $t(H)$.*

It is possible to prove that BEXP runs in exponential time in the size of $\mathcal{O}$. Informally, the running time is bounded by the sum of sizes for each local tableau. There are at most exponentially many local tableaux (one per each conjunct $H$), and it can further be shown that the size of each local tableau is also exponential. See the technical report [8] for more formal and detailed arguments.

## 4 Discussion and Preliminary Evaluation

The branch exploration procedure BEXP described in Sect. 3.1 can be readily used for classification, i.e., computing the entailed subsumption between atomic concepts in $\mathcal{O}$. We need to iterate over all singleton conjuncts $H$ of atomic concepts, and for each one compute the set $S(H)$ of atomic concepts, subsumptions with which occur on every branch. Another way to compute atomic subsumers for $H$, is to reduce subsumption entailment to concept satisfiability testing, as done in tableau-based procedures: $\mathcal{O} \models H \sqsubseteq D$ iff $\mathcal{O} \models H \sqcap \neg D \sqsubseteq \bot$.We can use our first procedure CSAT for this purpose. The next example demonstrates the key differences between these two approaches.

*Example 3 (Long Fork).* Consider the ontology $\mathcal{O}$ containing the following axioms:

$$A \sqsubseteq B \sqcup C, \quad B \sqsubseteq D, \quad C \sqsubseteq D, \quad D \sqsubseteq D_1 \sqcap \cdots \sqcap D_n, \quad (n > 0).$$

The two branches in the tree for $H = A$ contain the following sets of atomic subsumers for $A$: $S_1(A) = \{A, B, D, D_1, \ldots, D_n\}$, and $S_2(A) = \{A, C, D, D_1, \ldots, D_n\}$. Hence the procedure BEXP can find the set of common subsumers $S(A) = S_1(A) \cap S_2(A) = \{A, D, D_1, \ldots, D_n\}$ in $O(n)$ steps. If we use CSAT instead and test satisfiability for each candidate $A \sqsubseteq D_i$, $(1 \leq i \leq n)$, then we potentially need $O(n^2)$ steps since for each $H_i = A \sqcap \neg D_i$ we can produce $O(n)$ conclusions $A \sqcap \neg D_i \sqsubseteq D_j$, $1 \leq j \leq n$.

This difference between the two strategies can be observed not only on toy examples but also on real ontologies (cf. [8] for evaluation results).

Tableau reasoners usually implement further optimizations exploiting a partially constructed taxonomy that can reduce the number of subsumption tests [2, 4]. If in Example 3 the reasoner computes the subsumers for $D$ first (using $O(n)$ steps because they are derived deterministically), then after the first (positive) subsumption test for $A \sqsubseteq D$, all subsumers for $A$ are completely determined. This optimization can be also applied for BEXP to reduce the set of candidate subsumers $S(H)$ to be verified.

**Table 1.** Comparison of the performance (in seconds) for classification of ontologies between the new implementation and other reasoners. $\text{ELK}_\text{N}$ stands for ELK with N parallel worker threads. Total time includes CSAT, BEXP, and taxonomy construction (transitive reduction).

| Ontology | CSAT | BEXP | Total | ConDOR | Konclude | $\text{ELK}_1$ | $\text{ELK}_4$ |
|---|---|---|---|---|---|---|---|
| SNOMED CT | 22.7 | – | 29.8 | 38.2 | 45.2 | 25.7 | 13.1 |
| GALEN-EL | 2.3 | – | 2.6 | 3.6 | 2.7 | 2.1 | 1.0 |
| SCT-SEP | 16.9 | 15.9 | 37.5 | 59.4 | 246.6 | N/A | N/A |

### 4.1 Preliminary Experimental Evaluation

We have implemented our non-deterministic procedure, currently for a subset of $\mathcal{ALCH}$ without universals and negations (but with disjunctions). The implementation supports transitive roles via the well-known encoding [6]. The main goal is to make a preliminary comparison of the new non-deterministic procedure with the deterministic resolution-style procedure of ConDOR [13], which supports this fragment. We have also made a comparison with Konclude (v.0.5.0), which was the fastest tableau-based reasoner for our test ontologies, and with ELK (v.0.4.1) on $\mathcal{EL}^+$ ontologies. The latter is done to evaluate the potential overheads (our procedure works deterministically on Horn ontologies, so, in theory, it should behave like the $\mathcal{EL}^+$ procedure on $\mathcal{EL}^+$ ontologies).

We used the July 2013 release of SNOMED CT,[2] an $\mathcal{EL}^+$-restricted version of GALEN, and the version of SNOMED CT with disjunctions mentioned in the introduction (SCT-SEP). Performance is tested on classification and the results are averaged over 5 runs (excluding 2 warm-ups). We used a PC with Intel Core i5-2520M 2.50GHz CPU, running Java 1.6 with 4GB of RAM available to JVM.

Performance results of classification are presented in Table 1. The procedure works in two steps as described in Sect. 3.1: satisfiability testing for all atomic concepts using CSAT, then exploring branches to compute subsumers using BEXP. First, the results show that our implementation is slightly faster than ConDOR and considerably faster than Konclude on SCT-SEP. Second, it is comparable to ELK on deterministic $\mathcal{EL}^+$ ontologies. The numbers for ELK with 1 and 4 worker threads demonstrate that the difference is mostly due to concurrency: our implementation does not perform computations in parallel yet. BEXP is not necessary for this case as everything is deterministic.

### 4.2 Summary and Future Research Directions

In this paper, we presented a non-deterministic consequence-based reasoning procedure for $\mathcal{ALCHI}$, proved its correctness, and presented some promising preliminary evaluation. The procedure retains many nice properties of previously known consequence-based methods, such as optimal worst-case complexity, but also employs some tableau-like features, such as backtracking, without need for blocking to achieve termination.

Our future plan is to extend the implementation to fully support $\mathcal{ALCHI}$, integrate some optimizations used in ELK (including concurrency), and investigate extensions of the procedure with other constructors, such as number restrictions. We conjecture that it is easier to support new features using non-deterministic rules rather than resolution.

[2] `http://www.ihtsdo.org/snomed-ct/`

## References


1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L., Saffiotti, A. (eds.) Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05). pp. 364–369. Professional Book Center (2005)
2. Baader, F., Hollunder, B., Nebel, B., Profitlich, H.J., Franconi, E.: Am empirical analysis of optimization techniques for terminological representation systems. J. of Applied Intelligence 4(2), 109–132 (1994)
3. Bachmair, L., Ganzinger, H.: Equational reasoning in saturation-based theorem proving. In: Bibel, W., Schmitt, P.H. (eds.) Automated Deduction – A Basis for Applications, vol. I, chap. 11, pp. 353–397. Kluwer (1998)
4. Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. J. of Web Semantics 14, 84–101 (2012)
5. Gonçalves, R.S., Bail, S., Jiménez-Ruiz, E., Matentzoglu, N., Parsia, B., Glimm, B., Kazakov, Y.: OWL reasoner evaluation (ORE) workshop 2013 results: Short report. In: Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany. pp. 1–18 (2013)
6. Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Boutilier, C. (ed.) Proc. 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09). pp. 2040–2045. IJCAI (2009)
7. Kazakov, Y., Klinov, P.: Incremental reasoning in OWL EL without bookkeeping. In: International Semantic Web Conference. pp. 232–247 (2013)
8. Kazakov, Y., Klinov, P.: Bridging the gap between tableau and consequence-based reasoning. Technical report, The University of Ulm, Germany (2014), `http://elk.semanticweb.org/publications/alc-tr-2014.pdf`
9. Kazakov, Y., Krötzsch, M., Simančík, F.: Concurrent classification of $\mathcal{EL}$ ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) Proc. 10th Int. Semantic Web Conf. (ISWC'11). LNCS, vol. 7032, pp. 305–320. Springer (2011)
10. Kazakov, Y., Krötzsch, M., Simančík, F.: The incredible ELK: From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. J. of Automated Reasoning 53(1), 1–61 (2014)
11. Romero, A.A., Grau, B.C., Horrocks, I., Jiménez-Ruiz, E.: MORe: a modular OWL reasoner for ontology classification. In: Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany. pp. 61–67 (2013)
12. Sheeran, M., Stålmarck, G.: A tutorial on Stålmarck's proof procedure for propositional logic. Formal Methods in System Design 16(1), 23–58 (2000)
13. Simančík, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond Horn ontologies. In: Walsh, T. (ed.) Proc. 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11). pp. 1093–1098. AAAI Press/IJCAI (2011)
14. Tsarkov, D., Palmisano, I.: Chainsaw: a metareasoner for large ontologies. In: Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany (2012)