

# Advancing ELK: Not Only Performance Matters

Yevgeny Kazakov and Pavel Klinov

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

**Abstract.** This paper reports on the recent development of ELK, a consequence-based reasoner for  $\mathcal{EL}_\perp^+$  ontologies. It covers novel reasoning techniques which aim at improving efficiency and providing foundation for new reasoning services. On the former front we present a simple optimization for handling of role composition axioms, such as transitivity, which substantially reduces the number of rule applications. For the latter, we describe a new rule application strategy that takes advantage of concept definitions to avoid many redundant inferences without making rules dependent on derived conclusions. This improvement is not visible to the end user but considerably simplifies implementation for incremental reasoning and proof generation. We also present a rewriting of low-level inferences used by ELK to higher-level proofs that can be defined in the standard DL syntax, and thus be used for automatic verification of reasoning results or (visual) ontology debugging. We demonstrate the latter capability using a new ELK Protégé plugin.

## 1 Introduction

ELK is an ontology reasoner designed for top classification performance on OWL EL ontologies [1]. Its characteristic features are consequence-based calculus, highly parallelizable reasoning, and aggressive optimizations to reduce the number of derived axioms sufficient for classification (deriving all subsumptions between concept names).

Sheer performance has been the sole goal for the first few versions of ELK and it enabled it to become the reasoner of choice in biomedical circles where large  $\mathcal{EL}$  ontologies are built to manage scientific terminologies [2–7]. After that ELK started to evolve towards providing additional reasoning-related services, such as incremental classification [8] and proof tracing [9]. It turned out that some traits of ELK’s classification procedure, in particular, the non-deterministic saturation, can complicate the development or weaken the guarantees of such extra services. For example, the composition/decomposition optimization (cf. [1]) has to be off when incrementally retracting inferences [8]. Also, the proof tracing method guarantees only that all proofs performed by ELK will be generated, not all proofs supported by the calculus [9]. This, in particular, means that one cannot in general obtain *all* justifications [10] from proofs.

In this paper we describe the steps towards adapting the main reasoning procedure to rectify this sort of issues without major performance setbacks. On the performance front we show a technique to reduce the number of inferences on roles. We also present a rewriting of the low-level traced inferences into higher-level proof-based explanations which could be shown to the user or verified using automated reasoning tools. Due to the space constraints, some results are deferred to the technical report [11].

$$\begin{array}{ll}
\mathbf{E}_0 \frac{}{C \sqsubseteq C} & \mathbf{E}_\sqcap^+ \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} \\
\mathbf{E}_\top \frac{}{C \sqsubseteq \top} & \mathbf{E}_\perp \frac{C \sqsubseteq \exists R.D \quad D \sqsubseteq \perp}{C \sqsubseteq \perp} \\
\mathbf{E}_\sqsubseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} & \mathbf{E}_\exists \frac{C \sqsubseteq \exists R.D \quad D \sqsubseteq E}{C \sqsubseteq \exists R.E} \\
\mathbf{E}_\sqcap^- \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1 \quad C \sqsubseteq D_2} & \mathbf{E}_\circ \frac{C_{i-1} \sqsubseteq \exists R_i.C_i \quad (1 \leq i \leq k, k \geq 0)}{C_0 \sqsubseteq \exists R.C_k} : R_1 \cdots R_k \sqsubseteq R \in \mathcal{O}
\end{array}$$

Fig. 1. Basic inference rules for reasoning in  $\mathcal{EL}_\perp^+$

## 2 Consequence-Based Reasoning in $\mathcal{EL}_\perp^+$

We first describe ELK's consequence-based procedure for  $\mathcal{EL}_\perp^+$  reasoning. Most theoretical results, such as completeness, redundancy elimination, and goal-directed rule application, are minor variations of [1] but the rules for dealing with role chain axioms without binarization and rules for reasoning with reflexive roles are new.

### 2.1 The Description Logic $\mathcal{EL}_\perp^+$

The syntax of  $\mathcal{EL}_\perp^+$  is defined using a vocabulary consisting of countably infinite sets of (*atomic*) *roles* and *atomic concepts*.  $\mathcal{EL}_\perp^+$  *concepts* are defined using the grammar  $\mathbf{C} ::= A \mid \top \mid \perp \mid C_1 \sqcap C_2 \mid \exists R.C$ , where  $A$  is an atomic concept,  $R$  a role, and  $C_{(i)} \in \mathbf{C}$ .  $\mathcal{EL}_\perp^+$  *role chains* are defined using the grammar  $\mathbf{P} ::= \epsilon \mid R \cdot \rho$ , where  $\epsilon$  is the *empty role chain*,  $R$  a role and  $\rho \in \mathbf{P}$ . We usually write role chains as  $R_1 \cdot R_2 \cdots R_n$  instead of  $R_1 \cdot (R_2 \cdots (R_n \cdot \epsilon))$ . An  $\mathcal{EL}_\perp^+$  *axiom* is either a *concept inclusion*  $C_1 \sqsubseteq C_2$  for  $C_1, C_2 \in \mathbf{C}$  or a *role inclusion*  $\rho \sqsubseteq R$  for  $\rho \in \mathbf{P}$  and a role  $R$ . We regard the *concept equivalence*  $C_1 \equiv C_2$  as an abbreviation for two concept inclusions  $C_1 \sqsubseteq C_2$  and  $C_2 \sqsubseteq C_1$ . We also call  $\epsilon \sqsubseteq R$  a *role reflexivity axiom*. An  $\mathcal{EL}_\perp^+$  *ontology*  $\mathcal{O}$  is a finite set of  $\mathcal{EL}_\perp^+$  axioms. Semantics of  $\mathcal{EL}_\perp^+$  is defined in the usual way ( $\epsilon$  is interpreted as identity). A concept  $C$  is *subsumed* by  $D$  w.r.t.  $\mathcal{O}$  if  $\mathcal{O} \models C \sqsubseteq D$ . In this case, we call  $C \sqsubseteq D$  an *entailed subsumption* (w.r.t.  $\mathcal{O}$ ). The *ontology classification task* requires to compute all entailed subsumptions between atomic concepts occurring in  $\mathcal{O}$ .

### 2.2 Inference Rules

Classification of  $\mathcal{EL}_\perp^+$  ontologies is usually performed by applying rules that derive logical consequences of axioms. Figure 1 lists the  $\mathcal{EL}_\perp^+$ -rules that are similar to those usually considered in the literature [12, 13]. The *premises* of the rules are written above the horizontal line, the *conclusions* below, and the axioms in the ontology (a.k.a. *side conditions*) that trigger rule applications after the colon. Note that rule  $\mathbf{E}_\circ$  can be used with  $k = 0$ , in which case it has no premises and uses the reflexivity axiom  $\epsilon \sqsubseteq R \in \mathcal{O}$ .

*Example 1.* Consider the  $\mathcal{EL}_\perp^+$  ontology  $\mathcal{O} = \{A \sqsubseteq \exists R.B, B \sqsubseteq \exists S.C, R \cdot S \cdot H \sqsubseteq V, \epsilon \sqsubseteq H\}$ . Then it is possible to derive  $A \sqsubseteq \exists V.C$  using the rules in Figure 1 as follows:

$$A \sqsubseteq A \quad \text{by } \mathbf{E}_0(), \quad (1)$$

$$A \sqsubseteq \exists R.B \quad \text{by } \mathbf{E}_{\sqsubseteq}(A \sqsubseteq A) : A \sqsubseteq \exists R.B \in \mathcal{O}, \quad (2)$$

$$B \sqsubseteq B \quad \text{by } \mathbf{E}_0(), \quad (3)$$

$$B \sqsubseteq \exists S.C \quad \text{by } \mathbf{E}_{\sqsubseteq}(B \sqsubseteq B) : B \sqsubseteq \exists S.C \in \mathcal{O}, \quad (4)$$

$$C \sqsubseteq \exists H.C \quad \text{by } \mathbf{E}_o() : \epsilon \sqsubseteq H \in \mathcal{O}, \quad (5)$$

$$A \sqsubseteq \exists V.C \quad \text{by } \mathbf{E}_o(A \sqsubseteq \exists R.B, B \sqsubseteq \exists S.C, C \sqsubseteq \exists H.C) : R \cdot S \cdot H \sqsubseteq V \in \mathcal{O}. \quad (6)$$

Formally, a *derivation* for  $\mathcal{EL}$  ontology  $\mathcal{O}$  (using the rules in Figure 1) is a sequence of  $\mathcal{EL}_{\perp}^+$  axioms  $d = \{\alpha_i \mid i \geq 1\}$  such that each  $\alpha_i$  with  $i \geq 0$  is obtained from axioms  $\{\alpha_j \mid 1 \leq j < i\}$  using one of the rules in Figure 1 and axioms in  $\mathcal{O}$  as side conditions. The size  $\|d\|$  of  $d$  is the number of axioms in  $d$ . For example, the sequence of axioms (1)–(6) in Example 1 forms a derivation, in which every axiom is obtained from the previous axioms by the rules in Figure 1 as indicated next to the axioms.

The rules in Figure 1 are simple to understand but not very efficient to implement. The problem is caused by rule  $\mathbf{E}_o$ , which may produce many conclusions for ontologies with deep role hierarchies. For example, consider  $\mathcal{O} = \{R_{j-1} \sqsubseteq R_j \mid 1 \leq j \leq m\} \cup \{C_{i-1} \sqsubseteq \exists R_0.C_i, \exists R_m.D_i \sqsubseteq D_{i-1} \mid 1 \leq i \leq n\} \cup \{C_n \sqsubseteq D_n\}$ . Then one can only derive  $C_0 \sqsubseteq D_0$  by the rules in Figure 1 by deriving quadratically-many intermediate axioms  $C_{i-1} \sqsubseteq \exists R_j.C_i$  by  $\mathbf{E}_o$  using  $R_{j-1} \sqsubseteq R_j \in \mathcal{O}$  ( $1 \leq i \leq n, 1 \leq j \leq m$ ). Therefore, ELK implements optimized rules listed in Figure 2 that help avoiding this problem by deriving subsumptions on role (chains) separately [1]. To formulate these rules, we have slightly extended the syntax of  $\mathcal{EL}_{\perp}^+$ . First, we can derive role chains on the right-hand side of role inclusions:  $\rho_1 \sqsubseteq \rho_2$  ( $\mathcal{I} \models \rho_1 \sqsubseteq \rho_2$  if  $\rho_1^{\mathcal{I}} \subseteq \rho_2^{\mathcal{I}}$ ). Second, we allow role chains to occur in existential restrictions:  $\exists(R \cdot \rho).C$  is rewritten to  $\exists R.C$  if  $\rho = \epsilon$ , or to  $\exists R.\exists \rho.C$  otherwise (whenever we write  $\exists \rho.C$  we assume that  $\rho \neq \epsilon$ ). The *extended  $\mathcal{EL}_{\perp}^+$  axioms* can be used in derivations, but not in the ontology  $\mathcal{O}$ .

*Example 2.* Below is the derivation for  $A \sqsubseteq \exists V.C$  for the ontology  $\mathcal{O}$  in Example 1 using the rules in Figure 2:

$$A \sqsubseteq A \quad \text{by } \mathbf{C}_0(), \quad (7)$$

$$A \sqsubseteq \exists R.B \quad \text{by } \mathbf{C}_{\sqsubseteq}(A \sqsubseteq A) : A \sqsubseteq \exists R.B \in \mathcal{O}, \quad (8)$$

$$R \sqsubseteq R \quad \text{by } \mathbf{R}_0(), \quad (9)$$

$$B \sqsubseteq B \quad \text{by } \mathbf{C}_0(), \quad (10)$$

$$B \sqsubseteq \exists S.C \quad \text{by } \mathbf{C}_{\sqsubseteq}(B \sqsubseteq B) : B \sqsubseteq \exists S.C \in \mathcal{O}, \quad (11)$$

$$S \sqsubseteq S \quad \text{by } \mathbf{R}_0(), \quad (12)$$

$$C \sqsubseteq C \quad \text{by } \mathbf{C}_0(), \quad (13)$$

$$\epsilon \sqsubseteq \epsilon \quad \text{by } \mathbf{R}_0(), \quad (14)$$

$$\epsilon \sqsubseteq H \quad \text{by } \mathbf{R}_{\sqsubseteq}(\epsilon \sqsubseteq \epsilon) : \epsilon \sqsubseteq H \in \mathcal{O}, \quad (15)$$

$$S \sqsubseteq S \cdot H \quad \text{by } \mathbf{R}_r^c(S \sqsubseteq S, \epsilon \sqsubseteq H), \quad (16)$$

$$A \sqsubseteq \exists(R \cdot S \cdot H).C \quad \text{by } \mathbf{C}_o(A \sqsubseteq \exists R.B, R \sqsubseteq R, B \sqsubseteq \exists S.C, S \sqsubseteq S \cdot H), \quad (17)$$

$$\begin{array}{ll}
\mathbf{R}_0 \frac{}{\rho \sqsubseteq \rho} & \mathbf{R}_1^\epsilon \frac{\epsilon \sqsubseteq R \quad \rho_1 \sqsubseteq \rho_2}{\rho_1 \sqsubseteq R \cdot \rho_2} \\
\mathbf{R}_\sqsubseteq \frac{\rho_1 \sqsubseteq \rho_2}{\rho_1 \sqsubseteq R} : \rho_2 \sqsubseteq R \in \mathcal{O} & \mathbf{R}_r^\epsilon \frac{\rho_1 \sqsubseteq R \quad \epsilon \sqsubseteq \rho_2}{\rho_1 \sqsubseteq R \cdot \rho_2} \\
\mathbf{C}_0 \frac{}{C \sqsubseteq C} & \mathbf{C}_\perp \frac{C \sqsubseteq \exists \rho. D \quad D \sqsubseteq \perp}{C \sqsubseteq \perp} \\
\mathbf{C}_\top \frac{}{C \sqsubseteq \top} & \mathbf{C}_\exists^\epsilon \frac{C \sqsubseteq D \quad \epsilon \sqsubseteq R}{C \sqsubseteq \exists R. D} \\
\mathbf{C}_\sqsubseteq \frac{C \sqsubseteq D}{C \sqsubseteq E} : D \sqsubseteq E \in \mathcal{O} & \mathbf{C}_\exists \frac{C \sqsubseteq \exists \rho. D \quad \rho \sqsubseteq R \quad D \sqsubseteq E}{C \sqsubseteq \exists R. E} \\
\mathbf{C}_\sqcap^- \frac{C \sqsubseteq D_1 \sqcap D_2}{C \sqsubseteq D_1 \quad C \sqsubseteq D_2} & \mathbf{C}_o \frac{C \sqsubseteq \exists \rho_1. D \quad \rho_1 \sqsubseteq R \quad D \sqsubseteq \exists \rho_2. E \quad \rho_2 \sqsubseteq \rho}{C \sqsubseteq \exists (R \cdot \rho). E} \\
\mathbf{C}_\sqcap^+ \frac{C \sqsubseteq D_1 \quad C \sqsubseteq D_2}{C \sqsubseteq D_1 \sqcap D_2} &
\end{array}$$

**Fig. 2.** Optimized inference rules for reasoning in  $\mathcal{EL}_\perp^+$  implemented in ELK

$$R \cdot S \cdot H \sqsubseteq R \cdot S \cdot H \quad \text{by } \mathbf{R}_0(), \quad (18)$$

$$R \cdot S \cdot H \sqsubseteq V \quad \text{by } \mathbf{R}_\sqsubseteq(R \cdot S \cdot H \sqsubseteq R \cdot S \cdot H) : R \cdot S \cdot H \sqsubseteq V \in \mathcal{O}, \quad (19)$$

$$A \sqsubseteq \exists V. C \quad \text{by } \mathbf{C}_\exists(A \sqsubseteq \exists (R \cdot S \cdot H). C, R \cdot S \cdot H \sqsubseteq V, C \sqsubseteq C). \quad (20)$$

As can be seen from Examples 1 and 2, derivations using the rules in Figure 2 can be more difficult to understand because they use relatively complex rules such as  $\mathbf{R}_r^\epsilon$  and  $\mathbf{C}_o$  and manipulate with extended  $\mathcal{EL}_\perp^+$  axioms such as  $A \sqsubseteq \exists (R \cdot S \cdot H). C$  and  $S \sqsubseteq S \cdot H$ , the last of which is not even expressible in  $\mathcal{EL}^+$ . Fortunately, it is always possible to rewrite any derivation by the rules in Figure 2 into the one by the rules in Figure 1 using a simple recursive procedure (with an unavoidable quadratic blowup):

**Theorem 1.** *Let  $\mathcal{O}$  be an  $\mathcal{EL}_\perp^+$  ontology,  $d$  a derivation by the rules in Figure 2 for  $\mathcal{O}$ , and  $F \sqsubseteq G \in d$  an (ordinary)  $\mathcal{EL}_\perp^+$  axiom. Then one can construct a derivation  $e$  by the rules in Figure 1 for  $\mathcal{O}$  with  $F \sqsubseteq G \in e$  such that  $\|e\| = O(\|d\|^2)$ .*

The proof of Theorem 1 can be found in the technical report [11], which also contains an overview of the Protégé plug-in for displaying proofs based on this result.

### 2.3 Composed Conclusions, Redundancy and Completeness

From now on, we focus in the rules in Figure 2, so when we talk about axioms derived by these rules, we mean extended  $\mathcal{EL}_\perp^+$  axioms. It is easy to see that the rules in Figure 2 are *sound*, that is, the conclusions of the rules are logical consequences of the premises and the side conditions, if there are any. Therefore, every derivation contains only axioms entailed by the ontology. It turns out that the converse property also holds:

**Theorem 2.** *Let  $\mathcal{O}$  be an  $\mathcal{EL}_\perp^+$  ontology and  $F \sqsubseteq G$  an  $\mathcal{EL}_\perp^+$  concept inclusion such that  $\mathcal{O} \models F \sqsubseteq G$ . Then there exists a derivation  $d$  using the rules in Figure 2 such that either  $F \sqsubseteq G \in d$  or  $F \sqsubseteq \perp \in d$ .*

Similarly to existing results [12, 1], Theorem 2 is proved by constructing a canonical model using the set of all derivable axioms. One can actually prove a stronger version of this theorem, namely that every entailed subsumption is derivable by an optimized derivation—a derivation which does not use a certain kind of redundant inferences [11].

**Definition 1.** We say that an axiom  $\alpha$  in a derivation  $d$  is composed if can be obtained by rules  $\mathbf{C}_{\sqsupset}^+$ ,  $\mathbf{C}_{\sqsupset}^\varepsilon$  or  $\mathbf{C}_{\sqsupset}$  from the previous axioms. An application of a rule in Figure 2 to premises in  $d$  is redundant if it is an application by rule  $\mathbf{C}_{\sqsupset}^-$  in which the premise is composed, by rule  $\mathbf{C}_{\sqsupset}$  in which the first premise is composed, or by rule  $\mathbf{C}_o$  in which the first or the third premise is composed. The derivation  $d$  is optimized if every axiom  $\alpha$  in  $d$  is obtained from the previous axioms using a non-redundant rule application.

*Example 3.* Consider the ontology  $\mathcal{O} = \{A \sqsubseteq \exists R.B, B \sqsubseteq C, C \sqsubseteq D\}$ . Then the following derivation using the rules in Figure 2 is possible:

$$\begin{array}{lll}
A \sqsubseteq A & \text{by } \mathbf{C}_0(), & (21) \\
A \sqsubseteq \exists R.B & \text{by } \mathbf{C}_{\sqsubseteq}(A \sqsubseteq A) : A \sqsubseteq \exists R.B \in \mathcal{O}, & (22) \\
B \sqsubseteq B & \text{by } \mathbf{C}_0(), & (23) \\
B \sqsubseteq C & \text{by } \mathbf{C}_{\sqsubseteq}(B \sqsubseteq B) : B \sqsubseteq C \in \mathcal{O}, & (24) \\
B \sqsubseteq D & \text{by } \mathbf{C}_{\sqsubseteq}(B \sqsubseteq C) : C \sqsubseteq D \in \mathcal{O}, & (25) \\
R \sqsubseteq R & \text{by } \mathbf{R}_0(), & (26) \\
(+)\ A \sqsubseteq \exists R.C & \text{by } \mathbf{C}_{\sqsupset}(A \sqsubseteq \exists R.B, R \sqsubseteq R, B \sqsubseteq C), & (27) \\
C \sqsubseteq C & \text{by } \mathbf{C}_0(), & (28) \\
C \sqsubseteq D & \text{by } \mathbf{C}_{\sqsubseteq}(C \sqsubseteq C) : C \sqsubseteq D \in \mathcal{O}, & (29) \\
(+)\ A \sqsubseteq \exists R.D & \text{by } \mathbf{C}_{\sqsupset}(A \sqsubseteq \exists R.C, R \sqsubseteq R, C \sqsubseteq D). & (30)
\end{array}$$

In this derivation, axioms (27) and (30) (labeled with +) are composed because they were obtained by rule  $\mathbf{C}_{\sqsupset}$ . Hence the inference that has produced (30) is redundant because it is made by an application of  $\mathbf{C}_{\sqsupset}$  to a composed first premise (27). Still, the derivation (21)–(30) is optimized because (30) *can* be obtained from the previous axioms by another (non-redundant) application of  $\mathbf{C}_{\sqsupset}$  to a non-composed premise (22):

$$(+)\ A \sqsubseteq \exists R.D \quad \text{by } \mathbf{C}_{\sqsupset}(A \sqsubseteq \exists R.B, R \sqsubseteq R, B \sqsubseteq D). \quad (31)$$

In other words, since a derivation is a sequence of axioms and not a sequence of rule applications, it matters by which inferences axioms *can* be obtained from the previous axioms. Note that if (25) is removed from the derivation, then the inference (31) is no longer possible and the derivation becomes non-optimized.

In practice, the optimization above means that when applying the rules in Figure 2 to check entailment of concept inclusion, it is not necessary to apply  $\mathbf{C}_{\sqsupset}^-$  to premises derived by  $\mathbf{C}_{\sqsupset}^+$  or apply  $\mathbf{C}_{\sqsupset}$  and  $\mathbf{C}_o$  to premises derived by  $\mathbf{C}_{\sqsupset}$  [1].

## 2.4 The Subformula Property and Goal-Directed Rule Application

So far Theorem 2 cannot be used for effectively checking if a given subsumption  $C \sqsubseteq D$  is entailed by the ontology  $\mathcal{O}$  since there are infinitely many axioms that can be

derived by the rules in Figure 2—already  $\mathbf{C}_0$  can produce infinitely many conclusions. It turns out, it is sufficient to derive only axioms of the form  $\rho_1 \sqsubseteq \rho_2$ ,  $C_1 \sqsubseteq C_2$ , and  $C_1 \sqsubseteq \exists \rho_2.C_2$  such that all concepts  $C_i$  and role chains  $\rho_i$  ( $i = 1, 2$ ) occur (possibly as sub-expressions) either in the ontology  $\mathcal{O}$  or in the given subsumption  $F \sqsubseteq G$  tested for entailment [1]. In other words, if  $\mathcal{O} \models F \sqsubseteq G$  then  $F \sqsubseteq G$  or  $F \sqsubseteq \perp$  is derivable by the rules in Figure 2 without creating new concepts or role chains. We refer to this property as *subformula property*. The subformula property implies that checking entailment  $\mathcal{O} \models F \sqsubseteq G$  can be done in polynomial time since there are at most polynomially-many different axioms of the above forms, and one can compute a derivation  $d$  containing all such axioms by repeatedly applying the rules in Figure 2.

The rules, however, can be restricted even further. Specifically, an axiom  $C_1 \sqsubseteq C_2$  needs to be derived in  $d$  only if  $C_1 = F$  for the tested subsumption  $F \sqsubseteq G$  or if some non-composed axiom of the form  $C \sqsubseteq \exists \rho.C_1$  is already derived in  $d$ . Indeed, it is easy to observe from the rules in Figure 2 that an axiom  $C_1 \sqsubseteq C_2$  can be used in a rule application only if this rule derives a concept inclusion axiom with the same left-hand side  $C_1$  or it uses another non-composed axiom of the form  $C \sqsubseteq \exists \rho.C_1$  (rules  $\mathbf{C}_\exists$  and  $\mathbf{C}_\circ$ ). Similarly, an axiom  $\rho_1 \sqsubseteq \rho_2$  needs to be derived only if  $\rho_1 = \epsilon$  or if some other non-composed axiom of the form  $C \sqsubseteq \exists \rho_1.D$  is already derived. We call this optimization *goal-directed rule application*.

*Example 4.* Consider the ontology  $\mathcal{O} = \{A \sqsubseteq \exists R.B, A \sqsubseteq \exists R.C, B \sqsubseteq C, C \sqsubseteq B\}$ . Suppose we want to check whether  $\mathcal{O} \models A \sqsubseteq B$ . Then the following goal-directed non-redundant rule applications can be performed:

$$A \sqsubseteq A \quad \text{by } \mathbf{C}_0(), \quad (32)$$

$$A \sqsubseteq \exists R.B \quad \text{by } \mathbf{C}_\sqsubseteq(A \sqsubseteq A) : A \sqsubseteq \exists R.B \in \mathcal{O}, \quad (33)$$

$$B \sqsubseteq B \quad \text{by } \mathbf{R}_0(), \quad (34)$$

$$B \sqsubseteq C \quad \text{by } \mathbf{R}_\sqsubseteq(B \sqsubseteq B) : B \sqsubseteq C \in \mathcal{O}, \quad (35)$$

$$R \sqsubseteq R \quad \text{by } \mathbf{R}_0(), \quad (36)$$

$$(+)\ A \sqsubseteq \exists R.C \quad \text{by } \mathbf{C}_\exists(A \sqsubseteq \exists R.B, R \sqsubseteq R, B \sqsubseteq C), \quad (37)$$

$$(+)\ A \sqsubseteq \exists R.C \quad \text{by } \mathbf{C}_\sqsubseteq(A \sqsubseteq A) : A \sqsubseteq \exists R.C \in \mathcal{O}. \quad (38)$$

Note that deriving axioms with the left-hand side  $C$  (e.g.,  $C \sqsubseteq C$  by rule  $\mathbf{C}_0$ ) is not necessary since the axiom  $A \sqsubseteq \exists R.C$  is composed (and thus, e.g., cannot be used in rule  $\mathbf{C}_\exists$  like axiom  $A \sqsubseteq \exists R.B$ ). Since no further rules need to be applied and the axiom  $A \sqsubseteq B$  is not derived, we can conclude that  $\mathcal{O} \not\models A \sqsubseteq B$ . Note that if we swap (33) with (38), then the axiom  $A \sqsubseteq \exists R.C$  would not be composed and we would need to derive subsumptions  $C \sqsubseteq C$  and  $C \sqsubseteq B$  by rules  $\mathbf{C}_0$  and  $\mathbf{C}_\sqsubseteq$ . Thus the set of derived axioms depends on the order in which the rules are applied.

Note that if a derivation  $d$  contains  $C \sqsubseteq D$  or  $\rho_1 \sqsubseteq \rho_2$  then  $C \sqsubseteq C$  or  $\rho_1 \sqsubseteq \rho_1$  must be derived in  $d$  respectively by  $\mathbf{C}_0$  and  $\mathbf{R}_0$  before that. Therefore, to save space, from now on we skip applications of the rules  $\mathbf{C}_0$  and  $\mathbf{R}_0$  (e.g., like (32), (34), and (36) in Example 4). We will also skip application of the rules  $\mathbf{C}_\sqsubseteq$  and  $\mathbf{R}_\sqsubseteq$  producing axioms in the ontology from the conclusions of  $\mathbf{C}_0$  and  $\mathbf{R}_0$  (e.g., like (33) and (35) in Example 4).

### 3 Avoiding Duplicate Role Compositions

In this section, we present an optimization, using which one can avoid some duplicate conclusions of rule  $\mathbf{C}_o$ . Intuitively, this optimization is designed to deal more efficiently with specific role chain axioms such as transitivity  $T \cdot T \sqsubseteq T$ . It is closely related to a similar optimization for role chain axioms presented previously for a fragment of  $\mathcal{EL}_\perp^+$  [13]. To illustrate the problem addressed, consider the following example.

*Example 5.* Consider the ontology  $\mathcal{O} = \{A \sqsubseteq \exists L.B, B \sqsubseteq \exists P.C, C \sqsubseteq \exists P.D, L \cdot P \sqsubseteq L, P \cdot P \sqsubseteq P\}$ . The roles  $L$  and  $P$  can be thought of as expressing the ‘located-in’ and ‘part-of’ relations. So the last axiom of  $\mathcal{O}$ , in particular, expresses that if  $x$  is located in  $y$  and  $y$  is a part of  $z$  then  $x$  is located in  $z$ . Let us try to determine whether  $\mathcal{O} \models A \sqsubseteq B$  using goal-direct application of rules (skipping applications of  $\mathbf{C}_o$  and  $\mathbf{R}_o$ , and applications of  $\mathbf{C}_\sqsubseteq$ ,  $\mathbf{R}_\sqsubseteq$  producing axioms in  $\mathcal{O}$  as noted before):

$$A \sqsubseteq \exists(L \cdot P).C \quad \text{by } \mathbf{C}_o(A \sqsubseteq \exists L.B, L \sqsubseteq L, B \sqsubseteq \exists P.C, P \sqsubseteq P), \quad (39)$$

$$A \sqsubseteq \exists(L \cdot P).D \quad \text{by } \mathbf{C}_o(A \sqsubseteq \exists(L \cdot P).C, L \cdot P \sqsubseteq L, C \sqsubseteq \exists P.D, P \sqsubseteq P), \quad (40)$$

$$B \sqsubseteq \exists(P \cdot P).D \quad \text{by } \mathbf{C}_o(B \sqsubseteq \exists P.C, P \sqsubseteq P, C \sqsubseteq \exists P.D, P \sqsubseteq P), \quad (41)$$

$$A \sqsubseteq \exists(L \cdot P).D \quad \text{by } \mathbf{C}_o(A \sqsubseteq \exists L.B, L \sqsubseteq L, B \sqsubseteq \exists(P \cdot P).D, P \cdot P \sqsubseteq P). \quad (42)$$

Note that the axiom  $A \sqsubseteq \exists(L \cdot P).D$  was derived two times by  $\mathbf{C}_o$  in (40) and (42). Intuitively, this is because the role chain inclusion  $L \cdot P \cdot P \sqsubseteq L \cdot P$  can be proved in two ways: either as  $(L \cdot P) \cdot P \sqsubseteq L \cdot P$  using  $L \cdot P \sqsubseteq L$  or as  $L \cdot (P \cdot P) \sqsubseteq L \cdot P$  using  $P \cdot P \sqsubseteq P$ .

In general, suppose that we have a derivation where some axiom is derived by  $\mathbf{C}_o$ :

$$C \sqsubseteq \exists(R \cdot \rho).F \quad \text{by } \mathbf{C}_o(C \sqsubseteq \exists S.D, S \sqsubseteq R, D \sqsubseteq \exists(T \cdot \mu).F, (T \cdot \mu) \sqsubseteq \rho). \quad (43)$$

Let us try to determine when the same conclusion can be derived differently. Suppose that  $\mu \neq \epsilon$ . Then  $D \sqsubseteq \exists(T \cdot \mu).F$  can be only derived by  $\mathbf{C}_o$ :

$$D \sqsubseteq \exists(T \cdot \mu).F \quad \text{by } \mathbf{C}_o(D \sqsubseteq \exists \tau.E, \tau \sqsubseteq T, E \sqsubseteq \exists \eta.F, \eta \sqsubseteq \mu). \quad (44)$$

Now suppose that we also have  $S \sqsubseteq S$ ,  $S \cdot T \sqsubseteq R$  and  $\eta \sqsubseteq \rho$  in the derivation. Then  $C \sqsubseteq \exists(R \cdot \rho).F$  can be derived as follows:

$$C \sqsubseteq \exists(S \cdot T).E \quad \text{by } \mathbf{C}_o(C \sqsubseteq \exists S.D, S \sqsubseteq S, D \sqsubseteq \exists \tau.E, \tau \sqsubseteq T), \quad (45)$$

$$C \sqsubseteq \exists(R \cdot \rho).F \quad \text{by } \mathbf{C}_o(C \sqsubseteq \exists(S \cdot T).E, S \cdot T \sqsubseteq R, E \sqsubseteq \exists \eta.F, \eta \sqsubseteq \rho). \quad (46)$$

Note that when we replace the rule application (43) with the two rule applications (45) and (46), the third premises  $D \sqsubseteq \exists \tau.E$  and  $E \sqsubseteq \exists \eta.F$  used in these applications appear in the derivation before the third premise  $D \sqsubseteq \exists(T \cdot \mu).F$  of (43) since the latter was obtained from the former by (44). Therefore, if we apply this transformation repeatedly to all inferences by  $\mathbf{C}_o$ , it will always terminate.

To summarize, we can always avoid applying  $\mathbf{C}_o$  with  $S \sqsubseteq R$  and  $T \cdot \mu \sqsubseteq \rho$  as the second and the last premises if  $\mu \neq \epsilon$  and we can derive  $S \cdot T \sqsubseteq R$  ( $S \sqsubseteq S$  is always derivable by  $\mathbf{R}_o$ ) and  $\tau \sqsubseteq \rho$  for every derivable  $\tau \sqsubseteq \mu$ . Due to the subformula

property, we can precompute all subsumptions on role chains occurring in the ontology and compute all pairs  $\langle \rho_1 \sqsubseteq R, \rho_2 \sqsubseteq \rho \rangle$  of such role subsumptions with  $\rho_1, \rho_2$  and  $R \cdot \rho$  occurring in the ontology, excluding the pairs  $\langle S \sqsubseteq R, T \cdot \mu \sqsubseteq \rho \rangle$  for which the above condition holds. Only the remaining pairs of subsumptions should then be used in  $\mathbf{C}_\mathcal{O}$ .

*Example 6.* Continuing Example 5, we can show that the above conditions hold for the pair  $\langle S \sqsubseteq R, T \cdot \mu \sqsubseteq \rho \rangle = \langle L \sqsubseteq L, P \cdot P \sqsubseteq P \rangle$ . Indeed,  $S \cdot T \sqsubseteq R = L \cdot P \sqsubseteq L$  can be derived, and since  $\mu = \rho = P$ ,  $\tau \sqsubseteq \rho$  is derivable if  $\tau \sqsubseteq \mu$  is. Thus, the pair  $\langle L \sqsubseteq L, P \cdot P \sqsubseteq P \rangle$  should not be used in  $\mathbf{C}_\mathcal{O}$ , and thus inference (42) is not necessary. One can show that the pair  $\langle P \sqsubseteq P, P \cdot P \sqsubseteq P \rangle$  should not be used in  $\mathbf{C}_\mathcal{O}$  as well.

As mentioned, there is a close relation of the optimization above with a similar optimization for rules in Figure 1 [13]. The main idea is to identify the role chain axioms  $\rho \sqsubseteq R$  for which rule  $\mathbf{E}_\mathcal{O}$  can be applied in a *left-linear way*, that is, only if all premises starting from the second one are derived by rules other than  $\mathbf{E}_\mathcal{O}$  with  $k \geq 2$ . This is the case, e.g., for both axioms  $L \cdot P \sqsubseteq L$  and  $P \cdot P \sqsubseteq P$  from ontology  $\mathcal{O}$  in Example 5. For example, rule  $\mathbf{E}_\mathcal{O}$  using  $L \cdot P \sqsubseteq L \in \mathcal{O}$  would be applied for  $A \sqsubseteq \exists L.B$  and  $B \sqsubseteq \exists P.C$  (derived by  $\mathbf{E}_\sqsubseteq$ ), but would not be applied for  $A \sqsubseteq \exists L.B$  and  $B \sqsubseteq \exists P.D$  if  $B \sqsubseteq \exists P.D$  is derived by  $\mathbf{E}_\mathcal{O}$  from  $B \sqsubseteq \exists P.C$  and  $C \sqsubseteq \exists P.D$  using  $P \cdot P \sqsubseteq P \in \mathcal{O}$ . The reason is that the same conclusion  $A \sqsubseteq \exists L.D$  can be derived in a left-linear way from  $A \sqsubseteq \exists L.C$  (derived by  $\mathbf{E}_\mathcal{O}$  using  $L \cdot P \sqsubseteq L \in \mathcal{O}$ ) and  $C \sqsubseteq \exists P.D$  (derived by  $\mathbf{E}_\sqsubseteq$ ). The main difference between the two optimizations, is that, due to the differences in the rules in Figure 1 and Figure 2, instead of classifying which *stated* role chain axioms can be used in a left-linear way, we determine which *derived* role chain axioms should be ‘concatenated’ in rule  $\mathbf{C}_\mathcal{O}$ . The latter is algorithmically easier to determine using the condition formulated above.

## 4 Deterministic Saturation

Recall from Example 4 that the set of axioms obtained by applying the rules in a goal-directed way may depend on the order in which the rules are applied. Although this side effect has no impact on reasoning results, it introduces some difficulties when extending ELK reasoning services beyond checking of logical entailment. Specifically, the procedures for incremental reasoning [8] and proof generation [9] implemented in ELK, require repeating some rule applications performed in the derivation, and if the rules are applied in a different order than originally, the procedures may result in incorrect results. In this section we describe a modification of our rule application procedure for which the derived axioms do not depend on the order of rule application.

Recall from Section 2.3 that to determine whether a rule such as  $\mathbf{R}_\exists$  should be applied to some axioms in the derivation, i.e., it is not redundant, one has to check which of these axioms are composed, i.e., *can be derived* by certain rules from the *previous* axioms in the derivation. This property ensures that only necessary rules are applied, but it can make rule application dependent on when (i.e. after which axioms) the premises were derived. Instead, we may slightly relax this requirement and decide whether an axiom is composed or not only based on the rules by which it was actually derived, which would make rule applications not to depend on other axioms in the derivation.



$$\mathbf{C}_{\equiv}^- \frac{C \sqsubseteq A}{C \sqsubseteq D} : A \equiv D \in \mathcal{O} \qquad \mathbf{C}_{\equiv}^+ \frac{C \sqsubseteq D}{C \sqsubseteq A} : A \equiv D \in \mathcal{O}$$

**Fig. 3.** The new inference rules for concept definitions

For example, axiom  $A \sqsubseteq \exists R.C$  in Example 4 was derived by both rules  $\mathbf{C}_{\exists}$  and  $\mathbf{C}_{\sqsubseteq}^-$ . We would then not apply  $\mathbf{C}_{\exists}$  to the first conclusion of a ‘composition’ inference by  $\mathbf{C}_{\exists}$ , but apply it to the second conclusion of ‘non-composition’ inference by  $\mathbf{C}_{\sqsubseteq}^-$ . Clearly, the advantage here is that it does not matter which of the two inferences was made first. It may seem that axioms are rarely derived by multiple rules, so the relaxed rule application strategy might not result in too many unnecessary inferences. The following example illustrates that this may not be really the case in practice.

*Example 7.* Consider the ontology  $\mathcal{O} = \{A \sqsubseteq B \sqcap \exists R.A, C \equiv \exists R.B\}$ . Recall, that concept equivalence  $C \equiv \exists R.B$  represents two axioms  $C \sqsubseteq \exists R.B$  and  $\exists R.B \sqsubseteq C$ . To test  $\mathcal{O} \models A \sqsubseteq C$ , we apply the rules in Figure 2 in a goal-directed way:

$$A \sqsubseteq B \qquad \text{by } \mathbf{C}_{\sqcap}^-(A \sqsubseteq B \sqcap \exists R.A), \qquad (47)$$

$$A \sqsubseteq \exists R.A \qquad \text{by } \mathbf{C}_{\sqcap}^-(A \sqsubseteq B \sqcap \exists R.A), \qquad (48)$$

$$(+)\ A \sqsubseteq \exists R.B \qquad \text{by } \mathbf{C}_{\exists}(A \sqsubseteq \exists R.A, R \sqsubseteq R, A \sqsubseteq B), \qquad (49)$$

$$A \sqsubseteq C \qquad \text{by } \mathbf{C}_{\sqsubseteq}^-(A \sqsubseteq \exists R.B) : \exists R.B \sqsubseteq C \in \mathcal{O}, \qquad (50)$$

$$A \sqsubseteq \exists R.B \qquad \text{by } \mathbf{C}_{\sqsubseteq}^-(A \sqsubseteq C) : C \sqsubseteq \exists R.B. \qquad (51)$$

Note that axiom  $A \sqsubseteq \exists R.B$  is derived by rules  $\mathbf{C}_{\exists}$  and  $\mathbf{C}_{\sqsubseteq}^-$ , so we would need to consider the second conclusion (51) for applications of  $\mathbf{C}_{\exists}$ . Note that the second rule application is a direct result of the equivalence axiom  $C \equiv \exists R.B$ , which was used to replace the subsumer  $\exists R.B$  in (49) with  $C$  and back with  $\exists R.B$ . So it is actually not possible to have application (51) before (49).

The scenario illustrated in Example 7 is rather common: whenever an axiom  $C \sqsubseteq D$  is derived and  $D$  occurs in some concept equivalence in the ontology, the same axiom  $C \sqsubseteq D$  will be derived again. To avoid such duplicate inferences, we introduce new rules in Figure 3 to deal specifically with *concept definitions*—concept equivalences  $A \equiv D$  where  $A$  is a (*defined*) atomic concept. Most concept equivalences in existing ontologies are of this form. We assume that all concept equivalences in  $\mathcal{O}$  are concept definitions and each atomic concept  $A$  is defined in at most one of them; remaining concept equivalences can be always replaced with concept inclusions. Finally, we extend Definition 1 by allowing composed axioms to be obtained by rule  $\mathbf{C}_{\sqsubseteq}^+$  and redundant rule applications to include applications by rule  $\mathbf{C}_{\equiv}^-$  in which the premise is composed. Note that if the premise of  $\mathbf{C}_{\equiv}^-$  is composed, then it can only be obtained by  $\mathbf{C}_{\sqsubseteq}^+$  using the same concept definition  $A \equiv D$ , and hence the conclusion of this rule must be already derived.

*Example 8.* Continuing Example 7, with the new rules in Figure 3 we will have just rule application (52) instead of (50)–(51); the application of rule  $\mathbf{C}_{\equiv}^-$  to (52) is redundant.

$$(+)\ A \sqsubseteq C \qquad \text{by } \mathbf{C}_{\equiv}^+(A \sqsubseteq \exists R.B) : C \equiv \exists R.B \in \mathcal{O}. \qquad (52)$$

**Table 1.** Summary information for the test ontologies (numbers of the different kinds of axioms)

Ontology	$C \sqsubseteq D$	$C \equiv D$	$R \sqsubseteq S$	$R \cdot R \sqsubseteq R$	$R_1 \cdot R_2 \sqsubseteq S$
EL-GALEN	25,563	9,968	958	58	0
GALEN7	27,820	15,270	1000	0	385
GALEN8	53,449	113,622	1024	0	385
SNOMED CT (Jan 2014)	229,330	69,908	11	0	1
ANATOMY	17,551	21,831	4	3	2

## 5 Preliminary Experimental Evaluation

In this section we present the preliminary results of empirical evaluation of the two new reasoning techniques described in Sections 3 and 4: the role composition optimization and the deterministic saturation using new rules for concept definitions. For both experiments we use a set of the well-known biomedical ontologies:<sup>1</sup> the July 2014 version of SNOMED CT,<sup>2</sup> three versions of OpenGALEN<sup>3</sup> (EL-GALEN, GALEN7, and GALEN8), and ANATOMY (an experimental version of SNOMED CT which uses role chain axioms to model the body structure). These ontologies have been frequently used in the past for evaluation of  $\mathcal{EL}$  reasoners [14, 15, 1]. The summary information about these ontologies is presented in Table 1.

For experiments we used a development version of ELK 0.5. The experimental setup is the same for all experiments: each ontology is classified 20 times, 10 warm-up runs to exclude the effects of JIT compilation and 10 measured runs, for which the results are averaged. The combined loading + classification wall clock time (in ms.) is used as the main performance metric. We used a PC with Intel Core i5-2520M 2.50GHz CPU, Java 1.6 and 4GB RAM available to JVM.

The first experiment evaluates effectiveness of the role composition optimization described in Section 3. All ontologies (except for SNOMED CT in which the only sub-role chain axiom has no effect) are classified with the optimization being turned on and off. The results are shown in Table 2. It can be seen that in most cases the optimization considerably reduces the number of inferences as well the number of derived subsumptions (many subsumptions are derived by several inferences). The difference translates into time savings. The ANATOMY ontology stands out as the case where the optimization is critically important since it cuts down the number of inferences by rule  $\mathbf{C}_o$  by nearly an order of magnitude.

The aim of the the second experiment is to evaluate the effectiveness of the deterministic saturation optimization described in Section 4. We compare the classification time and the number of derived axioms in three cases: a) with the ELK' current non-deterministic saturation [1], b) with deterministic saturation using the rules in Figure 2, and c) with deterministic saturation using the additional rules for concept definitions (see Figure 3). The results are shown in Table 3.

<sup>1</sup>Unless specified otherwise, the ontologies can be downloaded from the ELK project page [elk.semanticweb.org](http://elk.semanticweb.org)

<sup>2</sup><http://www.ihtsdo.org/licensing/>

<sup>3</sup><http://www.opengalen.org/sources/sources.html>

**Table 2.** Evaluation of the role composition optimization from Section 3

Ontology Optimization	Classification time		Inferences		Derived subsumptions	
	On	Off	On	Off	On	Off
EL-GALEN	964	1,039	2,080,194	2,207,823	1,485,247	1,550,695
GALEN7	1,632	1,998	5,707,082	7,001,796	2,787,261	2,952,952
GALEN8	15,587	16,981	40,239,327	49,719,156	19,377,172	20,136,178
ANATOMY	8,957	27,766	45,466,892	176,130,924	7,105,923	10,223,484

**Table 3.** Evaluation of deterministic saturation (see Section 3). The shortcuts “non-det”, “det”, and “det+defn” stand for non-deterministic saturation, unoptimized deterministic saturation, deterministic saturation with the new optimized rules for handling of concept definitions.

Ontology	Classification time			Inferences			Derived subsumptions		
	non-det	det	det+defn	non-det	det	det+defn	non-det	det	det+defn
EL-GALEN	959	1,513	1,032	2.1M	4.2M	2.3M	1.5M	2.7M	1.9M
GALEN7	1,984	3,408	2,699	5.7M	11.6M	9.9M	2.8M	5.3M	4.7M
GALEN8	16,750	27,339	19,866	40.2M	93.7M	65.3M	19.4M	37.2M	28.4M
SNOMED CT	14,441	21,791	15,431	25.5M	54.1M	31.2M	17.1M	30.3M	23.6M
ANATOMY	9,183	15,546	9,875	45.5M	73.0M	47.2M	7.1M	12.1M	8.3M

One can see that deterministic saturation without further optimizations is significantly slower and often makes nearly twice as many inferences as non-deterministic saturation. This is largely because many axioms are derived by multiple inferences due to equivalence axioms (as illustrated in Example 3). The special rules to deal with concept definitions reduce such redundant derivations and improve performance so that it is close to that of non-deterministic saturation. Still in some cases, e.g., for GALEN7 and GALEN8, the difference between non-deterministic and optimized deterministic strategies is visible and it remains our goal to investigate how it can be reduced even further.

## 6 Summary

The paper presented several recent developments in ELK which range from novel reasoning optimizations, such as efficient handling of role chain axioms, to modifications aimed at supporting additional reasoning services, such as proof-based explanations. Our experiments show that the latter changes *may* result in minor performance setbacks, and it remains our future goal to investigate how to avoid even such minor performance compromises.

## References

1. 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) (2014) 1–61
2. Harris, M.A., Lock, A., Bühler, J., Oliver, S.G., Wood, V.: FYPO: the fission yeast phenotype ontology. *Bioinformatics* **29**(13) (2013) 1671–1678

3. Hoehndorf, R., Dumontier, M., Gkoutos, G.V.: Identifying aberrant pathways through integrated analysis of knowledge in pharmacogenomics. *Bioinformatics* **28**(16) (2012) 2169–2175
4. Hoehndorf, R., Harris, M.A., Herre, H., Rustici, G., Gkoutos, G.V.: Semantic integration of physiology phenotypes with an application to the cellular phenotype ontology. *Bioinformatics* **28**(13) (2012) 1783–1789
5. Jupp, S., Stevens, R., Hoehndorf, R.: Logical gene ontology annotations (GOAL): exploring gene ontology annotations with OWL. *J. of Biomedical Semantics* **3**(Suppl 1)(S3) (2012) 1–16
6. Osumi-Sutherland, D., Reeve, S., Mungall, C.J., Neuhaus, F., Rutenberg, A., Jefferis, G.S.X.E., Armstrong, J.D.: A strategy for building neuroanatomy ontologies. *Bioinformatics* **28**(9) (2012) 1262–1269
7. The Gene Ontology Consortium: Gene ontology annotations and resources. *Nucleic Acids Res* (2012)
8. Kazakov, Y., Klinov, P.: Incremental reasoning in OWL EL without bookkeeping. In: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*. (2013) 232–247
9. Kazakov, Y., Klinov, P.: Goal-directed tracing of inferences in  $\mathcal{EL}$  ontologies. In: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014, Proceedings, Part II*. (2014) 196–211
10. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P., eds.: *Proc. 6th Int. Semantic Web Conf. (ISWC'07)*. Volume 4825 of LNCS., Springer (2007) 267–280
11. Kazakov, Y., Klinov, P.: Advancing ELK: Not only performance matters. Technical report, University of Ulm (2015) available from <http://http://elk.semanticweb.org/publications/elk-advancing-trdl-2015.pdf>.
12. 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)*, Professional Book Center (2005) 364–369
13. Kazakov, Y., Krötzsch, M., Simančík, F.: Unchain my  $\mathcal{EL}$  reasoner. In Rosati, R., Rudolph, S., Zakharyashev, M., eds.: *Proc. 24th Int. Workshop on Description Logics (DL'11)*. Volume 745 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 202–212
14. Baader, F., Lutz, C., Suntisrivaraporn, B.: Efficient reasoning in  $\mathcal{EL}^+$ . In Parsia, B., Sattler, U., Toman, D., eds.: *Proc. 19th Int. Workshop on Description Logics (DL'06)*. Volume 189 of CEUR Workshop Proceedings., CEUR-WS.org (2006)
15. Lawley, M.J., Bousquet, C.: Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner. In: *Proc. 6th Australasian Ontology Workshop (IAOA'10)*. (2010) 45–49