



Hypergraph-Based Inference Rules for Computing \mathcal{EL}^+ -Ontology Justifications

Hui Yang^(✉), Yue Ma, and Nicole Bidoit

LISN, CNRS, Université Paris-Saclay, Gif-sur-Yvette, France
{yang,ma,nicole.bidoit}@lisn.fr

Abstract. To give concise explanations for a conclusion obtained by reasoning over ontologies, *justifications* have been proposed as minimal subsets of an ontology that entail the given conclusion. Even though computing one justification can be done in polynomial time for tractable Description Logics such as \mathcal{EL}^+ , computing all justifications is complicated and often challenging for real-world ontologies. In this paper, based on a graph representation of \mathcal{EL}^+ -ontologies, we propose a new set of *inference rules* (called H-rules) and take advantage of them for providing a new method of computing all justifications for a given conclusion. The advantage of our setting is that most of the time, it reduces the number of *inferences* (generated by H-rules) required to derive a given conclusion. This accelerates the enumeration of justifications relying on these inferences. We validate our approach by running real-world ontology experiments. Our graph-based approach outperforms PULi [14], the state-of-the-art algorithm, in most of cases.

1 Introduction

Ontologies provide structured representations of domain knowledge that are suitable for AI reasoning. They are used in various domains, including medicine, biology, and finance. In the domain of ontologies, one of the interesting topics is to provide explanations of reasoning conclusions. To this end, *justifications* have been proposed to offer users a brief explanation for a given conclusion. Computing justifications has been widely explored for different tasks, for instance for debugging ontologies [1, 9, 11] and computing ontology modules [6]. Extracting just one justification can be easy for tractable ontologies, such as \mathcal{EL}^+ [17]. For instance, we can find one justification by deleting unnecessary axioms one by one. However, there may exist more than one justification for a given conclusion. Computing all such justifications is computationally complex and reveals itself to be a challenging problem [18].

There are mainly two different approaches [17] to compute all justifications for a given conclusion, the *black-box* approach and the *glass-box* approach. The *black-box* approach [11] relies only on a reasoner and, as such, can be

This work is funded by the BPI-France (PSPC AIDA: 2019-PSPC-09).

© The Author(s) 2022

J. Blanchette et al. (Eds.): IJCAR 2022, LNAI 13385, pp. 310–328, 2022.

https://doi.org/10.1007/978-3-031-10769-6_19

used for ontologies in any existing Description Logics. For example, a simple (naive) *black-box* approach would check all the subsets of the ontology using an existing reasoner and then filter the subset-minimal ones (i.e., justifications). Many advanced and optimized black-box algorithms have been proposed since 2007 [10]. Meanwhile, the glass-box approaches have achieved better performances over certain specific ontology languages (such as \mathcal{EL}^+ -ontology) by going deep into the reasoning process. Among them, the class of SAT-based methods [1–3, 14, 16] performs the best. The main idea developed by SAT-based methods is to trace, in a first step, a *complete set of inferences* (*complete set* for short) that contribute to the derivation of a given conclusion, and then, in a second step, to use SAT-tools or resolution to extract all justifications from these inferences. A detailed example is provided in Sect. 4.1.

In the real world, ontologies are always huge. For instance, the SnomedCT ontology contains more than 300,000 axioms. Thus, the traced *complete set* can be large, which could make it challenging to extract the justifications over them. Several techniques could be applied to reduce the size of the traced *complete set*, like the *locality-based modules* [8] and the *goal-directed tracing algorithm* [12]. One of their shared ideas is to identify, for a given conclusion, a particular part of the ontology relevant for the extraction of justifications. For example, the state-of-the-art algorithm, PULi [14], uses a *goal-directed tracing algorithm*. However, even for PULi, a simple ontology $\mathcal{O} = \{A_i \sqsubseteq A_{i+1} \mid 1 \leq i \leq n-1\}$ with the conclusion $A_0 \sqsubseteq A_n$ leads to a *complete set* containing $n-1$ inferences. This set can not be reduced further even with the previously mentioned optimizations. From this observation, we decided to explore a new SAT-based glass-box method to handle such situations better.

Now, let us look carefully at the ontology \mathcal{O} above, and let us regard each A_i as a graph node N_{A_i} . Then we are able to construct, for \mathcal{O} , a directed graph whose edges are of the form $N_{A_i} \rightarrow N_{A_{i+1}}$. It turns out that all the justifications for the conclusion $A_0 \sqsubseteq A_n$ are extracted from all the paths from N_{A_0} to N_{A_n} , and here we have only one such path. We can easily extend this idea on \mathcal{EL}^+ -ontology because most of the \mathcal{EL}^+ -axioms can be interpreted as direct edges except one case (i.e., $A \equiv B_1 \sqcap \dots \sqcap B_n$), for which we need a hyperedge (for more details see Definition 3). However, for more expressive ontologies, this translation becomes more complicated. For example, it is hard to map \mathcal{ALC} -axioms to edges as those axioms may contain negation or disjunction of concepts.

This example inspired us to explore a hypergraph representation of the ontology and reformulate inferences and justifications. Roughly, our inferences are built from elementary paths of the hypergraph and lead to particular paths called H-paths. Then, computing all the justifications for a given conclusion is made using such H-paths. For the previous ontology \mathcal{O} and the conclusion $A_0 \sqsubseteq A_n$, our *complete set* is reduced to only two inferences (no matter the value of n) corresponding to the unique path from N_{A_0} to N_{A_n} . The source of improvement provided by our method is twofold. On the one hand, it comes from the fact that elementary paths are pre-computed while extracting the inferences and that existing algorithms like depth-first search can efficiently compute such paths. On the other hand, yet as a consequence, decreasing the size of the *complete sets* of inferences leads to smaller inputs for the SAT-based algorithm

extracting justifications from the *complete set* (recall here that our method is a SAT-based glass-box method).

The paper is organized as follows. Section 2 introduces preliminary definitions and notions. In Sect. 3, we associate a hypergraph representation to \mathcal{EL}^+ -ontology and introduce a new set of inference rules, called H-rules, that generate our inferences. In Sect. 4, we develop the algorithm minH , which compute justifications based on our inferences. Section 5 shows experimental results and Sect. 6 summarizes our work.

2 Preliminaries

2.1 \mathcal{EL}^+ -Ontology

Given sets of atomic concepts $\mathbf{N}_C = \{A, B, \dots\}$ and atomic roles $\mathbf{N}_R = \{r, s, t, \dots\}$, the set of \mathcal{EL}^+ concepts C and axioms α are built by the following grammar rules:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C, \quad a ::= C \sqsubseteq C \mid C \equiv C \mid r \sqsubseteq s \mid r_1 \circ \dots \circ r_n \sqsubseteq s.$$

A \mathcal{EL}^+ -ontology \mathcal{O} is a finite set of \mathcal{EL}^+ -axioms. An **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of \mathcal{O} consists of a non-empty set $\Delta^{\mathcal{I}}$ and a mapping from atomic concepts $A \in \mathbf{N}_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and from roles $r \in \mathbf{N}_R$ to a subset $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For a concept C built from the grammar rules, we define $C^{\mathcal{I}}$ inductively by: $(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(\exists r.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b, (a, b) \in r^{\mathcal{I}}, b \in C^{\mathcal{I}}\}$, $(r \circ s)^{\mathcal{I}} = \{(a, b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists c, (a, c) \in r^{\mathcal{I}}, (c, b) \in s^{\mathcal{I}}\}$. An interpretation is a **model** of \mathcal{O} if it is compatible with all axioms in \mathcal{O} , i.e., for all $C \sqsubseteq D, C \equiv D, r \sqsubseteq s, r_1 \circ \dots \circ r_n \sqsubseteq s \in \mathcal{O}$, we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}, C^{\mathcal{I}} = D^{\mathcal{I}}, r^{\mathcal{I}} \subseteq s^{\mathcal{I}}, (r_1 \circ \dots \circ r_n)^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, respectively. We say $\mathcal{O} \models a$ where α is an axiom iff each model of \mathcal{O} is compatible with α . A concept A is **subsumed** by B w.r.t. \mathcal{O} if $\mathcal{O} \models A \sqsubseteq B$.

Next, we use A, B, \dots, G (possibly with subscripts) to denote atomic concepts and we use X, Y, Z (possibly with subscripts) to denote atomic concepts A, \dots, G , or complex concepts $\exists r.A, \dots, \exists r.G$.

We assume that ontologies are normalized. A \mathcal{EL}^+ -ontology \mathcal{O} is **normalized** if all its axioms are of the form $A \equiv B_1 \sqcap \dots \sqcap B_m, A \sqsubseteq B_1 \sqcap \dots \sqcap B_m, A \equiv \exists r.B, A \sqsubseteq \exists r.B, r \sqsubseteq s$, or $r \circ s \sqsubseteq t$, where $A, B, B_i \in \mathbf{N}_C$, and $r, s, t \in \mathbf{N}_R$. Every \mathcal{EL}^+ -ontology can be normalised in polynomial time by introducing new atomic concepts and atomic roles.

Example 1. *The following set of axioms is a \mathcal{EL}^+ -ontology:*

$$\mathcal{O} = \{ a_1: A \sqsubseteq D, a_2: D \sqsubseteq \exists r.E, a_3: E \sqsubseteq F, a_4: B \equiv \exists t.F, a_5: r \sqsubseteq t, a_6: G \equiv C \sqcap B, a_7: C \sqsubseteq A \}.$$

It is clear that $\mathcal{O} \models A \sqsubseteq \exists r.E$ as for all models \mathcal{I} , we have $A^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ by the axiom a_1 and $D^{\mathcal{I}} \subseteq (\exists r.E)^{\mathcal{I}}$ by a_2 .

Table 1. Inference rules over \mathcal{EL}^+ -ontology.

$$\begin{aligned}
\mathcal{R}_1 &: \frac{A \sqsubseteq A_1, \dots, A \sqsubseteq A_n, \quad A_1 \sqcap A_2 \sqcap \dots \sqcap A_n \sqsubseteq B}{A \sqsubseteq B} \\
\mathcal{R}_2 &: \frac{A \sqsubseteq A_1, \quad A_1 \sqsubseteq \exists r.B}{A \sqsubseteq \exists r.B} \quad \mathcal{R}_3 : \frac{A \sqsubseteq \exists r.B_1, \quad B_1 \sqsubseteq B_2, \quad \exists r.B_2 \sqsubseteq B}{A \sqsubseteq B} \\
\mathcal{R}_4 &: \frac{A_0 \sqsubseteq \exists r_1.A_1, \quad \dots, \quad A_{n-1} \sqsubseteq \exists r_n.A_n, \quad r_1 \circ \dots \circ r_n \sqsubseteq r}{A_0 \sqsubseteq \exists r.A_n}
\end{aligned}$$

2.2 Inference, Support and Justification

Given a \mathcal{EL}^+ -ontology \mathcal{O} , a major reasoning task over \mathcal{O} is *classification*, which aims at finding all subsumptions $\mathcal{O} \models A \sqsubseteq B$ for atomic concepts A, B occurring in \mathcal{O} . Generally, it can be solved by applying *inferences* recursively over \mathcal{O} [5].

An **inference** ρ is a pair $\langle \rho_{pre}, \rho_{con} \rangle$ whose *premise* set ρ_{pre} consists of \mathcal{EL}^+ -axioms and *conclusion* ρ_{con} is a single \mathcal{EL}^+ -axiom. As usual, a sequence of inferences ρ^1, \dots, ρ^n is a **derivation** of an axiom α from \mathcal{O} if $\rho^n_{con} = \alpha$ and for any $\beta \in \rho^i_{pre}, 1 \leq i \leq n$, we have $\beta \in \mathcal{O}$ or $\beta = \rho^j_{con}$ for some $j < i$.

As usual, **inference rules** are used to generate inferences. For instance, Table 1 [1,5] shows a set of inference rules for \mathcal{EL}^+ -ontologies. Next, we use $\mathcal{O} \vdash A \sqsubseteq B$ to denote that $A \sqsubseteq B$ is derivable from \mathcal{O} using inferences generated by the rules in Table 1. The set of inference rules in Table 1 is *sound* and *complete* for classification [5], i.e., $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{O} \vdash A \sqsubseteq B$ for any $A, B \in \mathbf{N}_C$.

A **support** of $A \sqsubseteq B$ over \mathcal{O} is a sub-ontology $\mathcal{O}' \subseteq \mathcal{O}$ such that $\mathcal{O}' \models A \sqsubseteq B$. The **justifications** for $A \sqsubseteq B$ are subset-minimal supports of $A \sqsubseteq B$. We denote the collection of all justifications for $A \sqsubseteq B$ w.r.t. \mathcal{O} by $J_{\mathcal{O}}(A \sqsubseteq B)$.

We say S is a **complete set** (of inferences) for $A \sqsubseteq B$ if for any justifications \mathcal{O}' of $A \sqsubseteq B$, we can derive $A \sqsubseteq B$ from \mathcal{O}' using only the inferences in S .

Example 2 (Example 1 cont'd). *Before applying inference rules, axioms in \mathcal{O} are preprocessed in order to be compatible with Table 1. For example, a_4 is replaced by $B \sqsubseteq \exists t.F$ and $\exists t.F \sqsubseteq B$. Then, according to the inference rules of Table 1, we may produce the following inferences: $\rho = \langle \{A \sqsubseteq D, D \sqsubseteq \exists r.E\}, A \sqsubseteq \exists r.E \rangle$, $\rho' = \langle \{A \sqsubseteq \exists r.E, r \sqsubseteq t\}, A \sqsubseteq \exists t.E \rangle$ and $\rho'' = \langle \{A \sqsubseteq \exists t.E, E \sqsubseteq F, \exists t.F \sqsubseteq B\}, A \sqsubseteq B \rangle$ generated by rule \mathcal{R}_2 , \mathcal{R}_4 and \mathcal{R}_3 respectively. Then $\mathcal{O} \vdash A \sqsubseteq B$ since $A \sqsubseteq B$ is derivable from \mathcal{O} by the sequence ρ, ρ', ρ'' .*

Notice that $\mathcal{O}' = \{a_1, a_2, a_3, a_4, a_5\}$ is a support for $A \sqsubseteq B$, and thus, any superset \mathcal{O}'' of \mathcal{O}' is a support of $A \sqsubseteq B$. \mathcal{O}' is also one of the justifications for $A \sqsubseteq B$ as for any $\mathcal{O}''' \subset \mathcal{O}'$, we have $\mathcal{O}''' \not\models A \sqsubseteq B$. Moreover, here the three inferences ρ, ρ', ρ'' provide a complete set for $A \sqsubseteq B$.

3 Hypergraph-Based Inference Rules

3.1 H-Inferences

In general, a (directed) hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of nodes \mathcal{V} and a set of hyperedges \mathcal{E} [4,7]. A hyperedge is of the form $e = (S_1, S_2), S_1, S_2 \subseteq \mathcal{V}$. In this paper, a hypergraph is associated to an ontology as follows:

Definition 3. For a given \mathcal{EL}^+ -ontology \mathcal{O} , the associated hypergraph is $\mathcal{G}_{\mathcal{O}} = (\mathcal{V}_{\mathcal{O}}, \mathcal{E}_{\mathcal{O}})$ where (i) the set of nodes $\mathcal{V}_{\mathcal{O}} = \{N_A, N_r, N_{\exists r.A} \mid A \in \mathbf{N}_C, r \in \mathbf{N}_R\}$ and (ii) the set of edges $\mathcal{E}_{\mathcal{O}}$ is defined by $f(\mathcal{O})$ where f is the multi-valued mapping shown in Fig. 1. Given a hyperedge e of $\mathcal{E}_{\mathcal{O}}$, the inverse image of e , $f^{-1}(e)$, is defined in the obvious manner. For a set E of hyperedges, $f^{-1}(E) = \cup_{e \in E} f^{-1}(e)$.

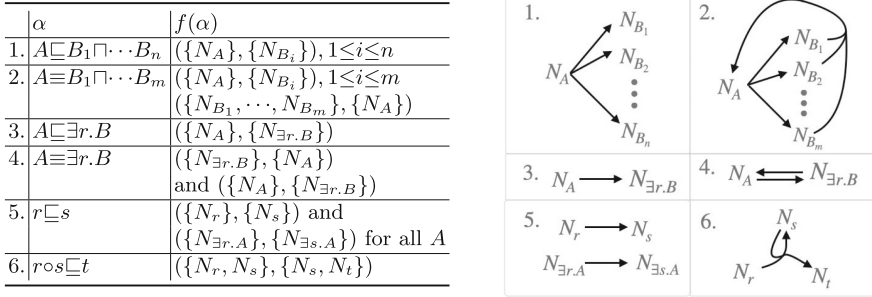


Fig. 1. Definition of f (left) and graphical illustrations of $f(\alpha)$ (right)

Notice that, the hyperedges associated with $A \equiv B_1 \sqcap \dots \sqcap B_m$ are (i) the hyperedge $(\{N_{B_1}, \dots, N_{B_m}\}, \{N_A\})$ and (2) of course, the edges corresponding to $A \sqsubseteq B_1 \sqcap \dots \sqcap B_m$.

Example 4 (Example 1 cont'd). The hypergraph $\mathcal{G}_{\mathcal{O}}$ for \mathcal{O} is shown in Fig. 2, where $e_0 = (\{N_C\}, \{N_A\})$, $e_1 = (\{N_A\}, \{N_D\})$, $e_2 = (\{N_D\}, \{N_{\exists r.E}\})$, etc. Also, $f^{-1}(e_0) = C \sqsubseteq A$, $f^{-1}(e_1) = A \sqsubseteq D$, and $f^{-1}(e_2) = D \sqsubseteq \exists r.E$, etc.

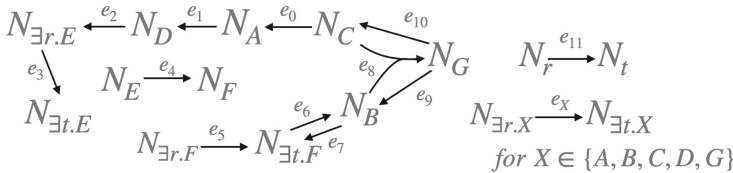


Fig. 2. The hypergraph associated with the ontology \mathcal{O} .

As for graphs, a path (next called **regular path**) from nodes N_1 to N_2 in a hypergraph is a sequence of edges:

$$e_0 = (S_1^0, S_2^0), e_1 = (S_1^1, S_2^1), \dots, e_n = (S_1^n, S_2^n) \tag{1}$$

where $N_1 \in S_1^0, N_2 \in S_2^n$ and $S_2^{i-1} = S_1^i, 1 \leq i \leq n$. Next, the **existence** of a regular path from N_X to N_Y in a hypergraph $\mathcal{G}_{\mathcal{O}}$ is denoted $N_X \rightsquigarrow N_Y$. Now, we introduce hypergraph-based inferences which are based on the existence of regular paths as follows:

Table 2. H-rules over $\mathcal{G}_{\mathcal{O}} = (\mathcal{V}_{\mathcal{O}}, \mathcal{E}_{\mathcal{O}})$.

$$\begin{aligned}
\mathcal{H}_0 : & \frac{N_X \rightsquigarrow N_Y}{N_X \overset{h}{\rightsquigarrow} N_Y} & \mathcal{H}_2 : & \frac{N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_1}, N_{B_1} \overset{h}{\rightsquigarrow} N_{B_2}, N_{\exists r.B_2} \rightsquigarrow N_Y}{N_X \overset{h}{\rightsquigarrow} N_Y} \\
\mathcal{H}_1 : & \frac{N_X \overset{h}{\rightsquigarrow} N_{B_1}, \dots, N_X \overset{h}{\rightsquigarrow} N_{B_m}, N_A \rightsquigarrow N_Y, e}{N_X \overset{h}{\rightsquigarrow} N_Y} : e = (\{N_{B_1}, \dots, N_{B_m}\}, \{N_A\}) \in \mathcal{E}_{\mathcal{O}} \\
\mathcal{H}_3 : & \frac{N_X \overset{h}{\rightsquigarrow} N_{\exists r.A_1}, N_{A_1} \overset{h}{\rightsquigarrow} N_{\exists s.A_2}, N_{\exists t.A_2} \rightsquigarrow N_Y, e}{N_X \overset{h}{\rightsquigarrow} N_Y} : e = (\{N_r, N_s\}, \{N_s, N_t\}) \in \mathcal{E}_{\mathcal{O}}
\end{aligned}$$

Definition 5. Given a hypergraph $\mathcal{G}_{\mathcal{O}}$, Table 2 gives a set of inference rules called **H-rules**. Inferences based on H-rules are called **H-inferences**. Next, we denote by $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$ (or simply $N_X \overset{h}{\rightsquigarrow} N_Y$) the fact that $N_X \overset{h}{\rightsquigarrow} N_Y$ can be derived from $\mathcal{G}_{\mathcal{O}}$ using the H-inferences.

Example 6 (Example 4 cont'd). As shown in Fig. 2, we have $N_A \rightsquigarrow N_{\exists r.E}$, $N_E \rightsquigarrow N_F$, $N_{\exists r.F} \rightsquigarrow N_B$ from the existence of regular paths. Then we can derive $N_A \overset{h}{\rightsquigarrow} N_B$ from $\mathcal{G}_{\mathcal{O}}$ by the H-rules \mathcal{H}_0 , \mathcal{H}_1 and \mathcal{H}_2 which generate the H-inferences ρ^1, ρ^2, ρ^3 , where $\rho^1 = \langle \{N_A \rightsquigarrow N_{\exists r.E}\}, N_A \overset{h}{\rightsquigarrow} N_{\exists r.E} \rangle$, $\rho^2 = \langle \{N_E \rightsquigarrow N_F\}, N_E \overset{h}{\rightsquigarrow} N_F \rangle$ and $\rho^3 = \langle \{N_A \overset{h}{\rightsquigarrow} N_{\exists r.E}, N_E \overset{h}{\rightsquigarrow} N_F, N_{\exists r.F} \rightsquigarrow N_B\}, N_A \overset{h}{\rightsquigarrow} N_B \rangle$, respectively.

Note that the first rule \mathcal{H}_0 , the initialization rule, makes regular paths the elementary components of H-rules. Moreover, Proposition 7 formally states that, in our H-inference system, we do not need to add the transitive inference rule:

$$\frac{N_X \overset{h}{\rightsquigarrow} N_Z, N_Z \overset{h}{\rightsquigarrow} N_Y}{N_X \overset{h}{\rightsquigarrow} N_Y}.$$

Proposition 7. If $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Z$ and $\mathcal{O} \vdash_h N_Z \overset{h}{\rightsquigarrow} N_Y$ then $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$.

3.2 Completeness and Soundness of H-Inferences

The following result is the main result of this section. It states the equivalence of $N_X \overset{h}{\rightsquigarrow} N_Y$ derivation (by Table 2) and ontology entailment for $X \sqsubseteq Y$, and thus states that our H-rules are sound and complete for \mathcal{EL}^+ -ontology.

Theorem 8. If \mathcal{O} is an \mathcal{EL}^+ -ontology, then $\mathcal{O} \models X \sqsubseteq Y$ iff $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$, where X, Y are concepts of either form A or $\exists r.B$.

Proof. “ \Leftarrow ” is obvious by induction over Table 2 and the fact that $N_X \rightsquigarrow N_Y$ implies $\mathcal{O} \models X \sqsubseteq Y$, so we only need to prove the direction “ \Rightarrow ”.

Assume that $\mathcal{O} \models X \sqsubseteq Y$. We consider two cases:

Case 1. We assume $\mathcal{O} \vdash X \sqsubseteq Y^1$. Let $d(X, Y)$ be the length of one shortest derivation of $X \sqsubseteq Y$ from \mathcal{O} using Table 1. We prove “ \Rightarrow ” by induction on $d(X, Y)$.

- **Assume** $d(X, Y) = 0$. In this case \mathcal{O} must contain axioms of the form $X \equiv Y \sqcap \dots$ or $X \sqsubseteq Y \sqcap \dots$. Clearly we have $N_X \rightsquigarrow N_Y$ thus $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$.
- **Assuming** “ \Rightarrow ” holds when $d(X, Y) < k$, let us prove “ \Rightarrow ” holds for $d(X, Y) = k$. Suppose ρ^{last} is the last inference in one shortest derivation of $X \sqsubseteq Y$ using Table 1. Two cases arise:

1. Assume ρ^{last} is generated by $\mathcal{R}_1(n > 1)$, \mathcal{R}_3 or $\mathcal{R}_4(n = 2)$. For example, assume $\rho^{last} = \langle \{X \sqsubseteq \exists r.B_1, B_1 \sqsubseteq B_2, \exists r.B_2 \sqsubseteq Y\}, X \sqsubseteq Y \rangle$ comes from \mathcal{R}_3 . We have $d(X, \exists r.B_1), d(B_1, B_2), d(\exists r.B_2, Y) < k$ because their corresponding subsumptions can be derived without ρ^{last} . By the assumption $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_1}, N_{B_1} \overset{h}{\rightsquigarrow} N_{B_2}, N_{\exists r.B_2} \overset{h}{\rightsquigarrow} N_Y$. Then we have $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_2}$ by first deriving $N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_1}, N_{B_1} \overset{h}{\rightsquigarrow} N_{B_2}$, and then applying H-inference:

$$\rho^{new} = \langle \{N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_1}, N_{B_1} \overset{h}{\rightsquigarrow} N_{B_2}, N_{\exists r.B_2} \rightsquigarrow N_{\exists r.B_2}\}, N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_2} \rangle.$$

Then $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$ by Proposition 7 since $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_{\exists r.B_2}, N_{\exists r.B_2} \overset{h}{\rightsquigarrow} N_Y$. The argument also holds for $\mathcal{R}_1(n > 1)$ (or $\mathcal{R}_4(n = 2)$) by applying \mathcal{H}_1 (or \mathcal{H}_3) instead of \mathcal{H}_2 .

2. Assume ρ^{last} is generated by $\mathcal{R}_1(n = 1)$, \mathcal{R}_2 or $\mathcal{R}_4(n = 1)$. Then, in each case, we have ρ^{last} has the form $\langle \{X \sqsubseteq Z, Z \sqsubseteq Y\}, X \sqsubseteq Y \rangle$. As in case 1, we have $d(X, Z), d(Z, Y) < k$. By the assumption, $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Z, N_Z \overset{h}{\rightsquigarrow} N_Y$, then $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$ by Proposition 7.

Case 2. If $\mathcal{O} \vdash X \sqsubseteq Y$ does not hold, then X or Y is not atomic. In this case, we introduce new axioms $A \equiv X, B \equiv Y$ with new atomic concepts A, B and denote the extended ontology by \mathcal{O}' . Clearly, $\mathcal{O}' \models A \sqsubseteq B$ and thus $\mathcal{O}' \vdash A \sqsubseteq B$ since Table 1 is sound and complete. Therefore, we have $\mathcal{O}' \vdash_h N_A \overset{h}{\rightsquigarrow} N_B$ by the same arguments as above. Now, notice that $\mathcal{G}_{\mathcal{O}'}$ is obtained from $\mathcal{G}_{\mathcal{O}}$ by adding 4 edges: $(\{N_A\}, \{N_X\}), (\{N_X\}, \{N_A\}), (\{N_B\}, \{N_Y\})$ and $(\{N_Y\}, \{N_B\})$, thus we have $\mathcal{O}' \vdash_h N_A \overset{h}{\rightsquigarrow} N_B$ iff $\mathcal{O} \vdash_h N_X \overset{h}{\rightsquigarrow} N_Y$.

3.3 Extracting Justifications from $\mathcal{G}_{\mathcal{O}}$

Now, we formally define H-paths as a hypergraph representation of classical derivations based on H-rules. The reader should pay attention to the fact that H-paths are not classical hyperpaths [7]. Next, for the sake of homogeneity, we consider a regular path from N_X to N_Y as the set of its edges and denote it as $P_{X,Y}$.

¹ The reader should recall that the equivalence ($\mathcal{O} \models X \sqsubseteq Y$ iff $\mathcal{O} \vdash X \sqsubseteq Y$) only holds when X and Y are atomic concepts wrt. the inference system presented in Table 1.

Definition 9 (H-paths). In the hypergraph $\mathcal{G}_{\mathcal{O}}$, an H-path $H_{X,Y}$ from N_X to N_Y is a set of edges recursively generated by the following composition rules:

0. A regular path $P_{X,Y}$ is an H-path from N_X to N_Y ;
1. If $e = (\{N_{B_1}, \dots, N_{B_m}\}, \{N_A\}) \in \mathcal{V}_{\mathcal{O}}$, if H_{X,B_i} are H-paths for $i = 1..m$, and if $P_{A,Y}$ is a regular path, then $H_{X,B_1} \cup \dots \cup H_{X,B_m} \cup P_{A,Y} \cup \{e\}$ is an H-path from N_X to N_Y ;
2. If $H_{X,\exists r.B_1}, H_{B_1,B_2}$ are H-paths and $P_{\exists r.B_2,Y}$ is a regular path, then $H_{X,\exists r.B_1} \cup H_{B_1,B_2} \cup P_{\exists r.B_2,Y}$ is an H-path from N_X to N_Y ;
3. If $e = (\{N_r, N_s\}, \{N_s, N_t\}) \in \mathcal{V}_{\mathcal{O}}$, if $H_{X,\exists r.A_1}, H_{A_1,\exists s.A_2}$ are H-paths and if $P_{\exists t.A_2,B}$ is a regular path, then $H_{X,\exists r.A_1} \cup H_{A_1,\exists s.A_2} \cup P_{\exists t.A_2,B} \cup \{e\}$ is an H-path from N_X to N_Y .

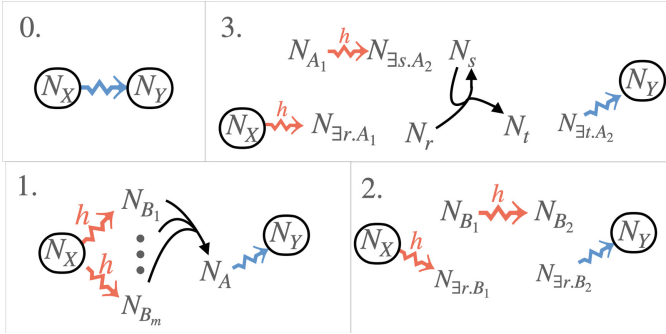


Fig. 3. Structure of H-paths from N_X to N_Y

Figure 3 gives an illustration of H-paths: the blue arrows \rightsquigarrow correspond to regular paths, and the red ones $\overset{h}{\rightsquigarrow}$ to H-paths. It is straightforward to compare composition rules building H-paths with H-rules building derivations in Table 2. One may also consider H-paths as deviation-trees with leaves corresponding to the edges in $\mathcal{G}_{\mathcal{O}}$. However, our approach provides a more direct characterization of justifications as shown in Theorem 10.

We say that an H-path $H_{X,Y}$ is **minimal** if there is no H-path $H'_{X,Y}$ such that $H'_{X,Y} \subset H_{X,Y}$.

Now, we are ready to explain how H-paths and justifications are related. We can compute justifications from minimal H-paths as stated below:

Theorem 10. Given X, Y of either form A or $\exists r.B$. Let

$$\mathcal{S} = \{f^{-1}(H_{X,Y}) \mid H_{X,Y} \text{ is a minimal H-path from } N_X \text{ to } N_Y\}.$$

Then $\mathcal{J}_{\mathcal{O}}(X \sqsubseteq Y) = \{s \in \mathcal{S} \mid s' \not\subseteq s, \forall s' \in \mathcal{S}\}$. That is, all justifications for $X \sqsubseteq Y$ are the minimal subsets in \mathcal{S} .

Proof. For any justification \mathcal{O}' of $X \sqsubseteq Y$, there exists a minimal H-path $H_{X,Y}$ such that $\mathcal{O}' = f^{-1}(H_{X,Y})$. The reason is that, since $\mathcal{O}' \models X \sqsubseteq Y$, there exists an H-path $H_{X,Y}$ from N_X to N_Y on $\mathcal{G}_{\mathcal{O}'}$ by Theorem 8. Without loss of generality, we can assume $H_{X,Y}$ is minimal on $\mathcal{G}_{\mathcal{O}'}$, then it is also minimal on $\mathcal{G}_{\mathcal{O}}$ since $\mathcal{G}_{\mathcal{O}'}$ is a sub-graph of $\mathcal{G}_{\mathcal{O}}$. We have $\mathcal{O}' = f^{-1}(H_{X,Y})$ because otherwise there exists $\mathcal{O}'' \subsetneq \mathcal{O}'$ such that $\mathcal{O}'' = f^{-1}(H_{X,Y})$, and thus $\mathcal{O}'' \models X \sqsubseteq Y$ by Theorem 8 again. Therefore, \mathcal{O}' is not a justification. Contradiction.

Now, we know \mathcal{S} contains all justifications for $X \sqsubseteq Y$. Moreover, $f^{-1}(H_{X,Y}) \models X \sqsubseteq Y$ for any H-path $H_{X,Y}$. Therefore, we have $J_{\mathcal{O}}(X \sqsubseteq Y) = \{s \in \mathcal{S} \mid s' \not\subseteq s, \forall s' \in \mathcal{S}\}$ by the definition of justifications.

Example 11. (Example 4 cont'd). *The regular paths from N_A to $N_{\exists r.E}$ and from N_E to N_F produce two H-paths $H_{A,\exists r.E} = \{e_1, e_2, e_3\}$ and $H_{E,F} = \{e_4\}$. Then, applying the third composition rule with $H_{A,\exists r.E}, H_{E,F}$ and $P_{\exists r.F,B} = \{e_6\}$, we get $H_{A,B} = \{e_1, e_2, e_3, e_4, e_6\}$, which is the unique H-path from N_A to N_B . Thus, by Theorem 10, we have $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$, the only justification for $A \sqsubseteq B$.*

4 Implementation: Computing Justifications

4.1 SAT-Based Method

In this section, we describe briefly how PULi [14], the state-of-the-art *glass-box* algorithm, proceeds. Given an ontology \mathcal{O} , computing $J_{\mathcal{O}}(X \sqsubseteq Y)$ is done through 2 steps: (1) tracing a *complete set* for $X \sqsubseteq Y$, (2) using resolution to extract the justifications from *the complete set*. The following example illustrates both steps:

Example 12 (Example 1 cont'd). *Let us compute $J_{\mathcal{O}}(G \sqsubseteq D)$ using PULi's method.*

1. *Using the goal-directed tracing algorithm in [12], the first step produces a complete set of inferences² $\{\rho_1, \rho_2\}$ for $G \sqsubseteq D$, where $\rho_1 = \langle \{G \sqsubseteq C, C \sqsubseteq A\}, G \sqsubseteq A \rangle$, $\rho_2 = \langle \{G \sqsubseteq A, A \sqsubseteq D\}, G \sqsubseteq D \rangle$.*
2. *This step is again composed of two parts:*
 - (a) *The first part proceeds to the translation of the inferences into clauses. Let us denote $\bar{p}_1: G \sqsubseteq C$, $\bar{p}_2: C \sqsubseteq A$, $\bar{p}_3: A \sqsubseteq D$, $p_4: G \sqsubseteq A$, $p_5: G \sqsubseteq D$. Here the literals $\bar{p}_1, \bar{p}_2, \bar{p}_3$ (with a bar) are called **answer literals** as they correspond to the axioms a_6, a_7, a_1 in \mathcal{O} . Thus, we obtain $\mathcal{C} = \{\neg \bar{p}_1 \vee \neg \bar{p}_2 \vee p_4, \neg p_4 \vee \neg \bar{p}_3 \vee p_5\}$ by rewriting the inferences ρ_1, ρ_2 as clauses.*
 - (b) *Secondly, a new clause $\neg p_5$ is added to \mathcal{C} , where p_5 corresponds to the conclusion $G \sqsubseteq D$, and resolution is applied over \mathcal{C} . The set of all justifications $J_{\mathcal{O}}(G \sqsubseteq D)$ is obtained by considering (i) the clauses formed of*

² For the sake of simplicity, we use the inference rules in Table 1 although PULi uses a slightly different set of inference rules [13].

Algorithm 1: minH

```

input :  $X \sqsubseteq Y$ 
output:  $J: \mathcal{J}_{\mathcal{O}}(X \sqsubseteq Y)$ .
1  $J \leftarrow \emptyset$ ;
2  $\mathcal{U} \leftarrow \text{CompleteH}(N_X \overset{h}{\rightsquigarrow} N_Y)$ ;
3  $\text{min\_hpaths} \leftarrow \text{resolution}(\text{clauses}(\mathcal{U}))$ ;
4 for  $h \in \text{min\_hpaths}$  do
5   if  $f^{-1}(h') \not\subseteq f^{-1}(h)$  for any  $h' \in \text{min\_hpaths}$  then
6      $J.add(f^{-1}(h))$ 
7   end
8 end

```

answer literals only and (ii) among them keeping the minimal ones³. In this example, after the resolution phase, the only clause that consists of merely answer literals is $\neg\bar{p}_1 \vee \neg\bar{p}_2 \vee \neg\bar{p}_3$. Thus, the set of all justifications is $J_{\mathcal{O}}(G \sqsubseteq D) = \{\{a_1, a_6, a_7\}\}$.

Our method for computing justifications follows the same steps as PULi although here the major difference is that the first step computes a complete set of H-inferences instead of a complete set of inferences wrt. Table 1.

4.2 Computing Justification by Minimal H-Paths

In this section, given an ontology \mathcal{O} and its associated hypergraph $\mathcal{G}_{\mathcal{O}}$, we present minH (Algorithm 1) that computes all justifications for $X_0 \sqsubseteq Y_0$ using the minimal H-paths from N_{X_0} to N_{Y_0} over $\mathcal{G}_{\mathcal{O}}$. The algorithm minH proceeds in two steps described below.

Step 1. First, at Line 2, minH computes a *complete set* of inferences \mathcal{U} for $N_{X_0} \overset{h}{\rightsquigarrow} N_{Y_0}$ using CompleteH (See Algorithm 2). Here, \mathcal{U} is complete in the sense that for any H-path $H_{X,Y}$, we can derive $N_X \overset{h}{\rightsquigarrow} N_Y$ using inferences in \mathcal{U} from the edge set $H_{X,Y}$. CompleteH computes \mathcal{U} as follows:

- **Line 3–12 of Algorithm 2:** The recursive application of `trace_one_turn` (See Algorithm 3) outputs the set of all H-inferences whose conclusion is the given input $N_{X_1} \overset{h}{\rightsquigarrow} N_{Y_1}$;
- **Line 13–17 of Algorithm 2:** Let `path` be the depth-first search algorithm that computes all regular paths from N_X to N_Y in $\mathcal{G}_{\mathcal{O}}$ with input (N_X, N_Y) . Intuitively, the purpose is to shift inferences from regular paths to edges.

Step 2. Then Algorithm minH computes all justifications for $X_0 \sqsubseteq Y_0$ as follows:

³ Here a clause c is smaller than c_1 if all the literals of c are in c_1 .

Algorithm 2: CompleteH

```

input :  $N_X \overset{h}{\rightsquigarrow} N_Y$ 
output:  $\mathcal{U}$ : a complete set of inferences for  $N_X \overset{h}{\rightsquigarrow} N_Y$ .
1  $\mathcal{U}$ , history,  $Q \leftarrow \emptyset$ ; //  $Q$  is a queue
2  $Q.add(N_X \overset{h}{\rightsquigarrow} N_Y)$ ;
3 while  $Q \neq \emptyset$  do
4    $N_{X_1} \overset{h}{\rightsquigarrow} N_{Y_1} \leftarrow Q.takeNext()$ ;
5   history.add( $N_{X_1} \overset{h}{\rightsquigarrow} N_{Y_1}$ );
6    $\mathcal{U} \leftarrow \mathcal{U} \cup \text{trace\_one\_turn}(N_{X_1} \overset{h}{\rightsquigarrow} N_{Y_1})$ ;
7   for  $N_{X_2} \overset{h}{\rightsquigarrow} N_{Y_2}$  appearing in  $\text{trace\_one\_turn}(N_{X_1} \overset{h}{\rightsquigarrow} N_{Y_1})$  do
8     if  $N_{X_2} \overset{h}{\rightsquigarrow} N_{Y_2} \notin \text{history}$  and  $N_{X_2} \overset{h}{\rightsquigarrow} N_{Y_2} \notin Q$  then
9        $Q.add(N_{X_2} \overset{h}{\rightsquigarrow} N_{Y_2})$ 
10    end
11  end
12 end
13 for  $N_{X_2} \rightsquigarrow N_{Y_2}$  appearing in  $\mathcal{U}$  do
14   for  $p = \{e_1, e_2, \dots, e_n\} \in \text{path}(N_{X_2}, N_{Y_2})$  do
15      $\mathcal{U}.add(\langle \{e_1, e_2, \dots, e_n\}, N_{X_2} \rightsquigarrow N_{Y_2} \rangle)$ ;
16   end
17 end

```

- **Line 3 of Algorithm 1:** It computes all minimal H-paths from N_{X_0} to N_{Y_0} using resolution, which is developed by PULi⁴, over the clauses generated from \mathcal{U} as illustrated in Sect. 4.1. Here, a literal p is associated with each edge e , each $N_X \overset{h}{\rightsquigarrow} N_Y$, and each $N_X \rightsquigarrow N_Y$ in \mathcal{U} . The answer literals are those associated with edges.
- **Line 4–8 of Algorithm 1:** It computes justifications by mapping back all the minimal H-paths and select the subset-minimal sets as stated in Theorem 10.

Example 13 (Example 4 cont’d). Assume $X_0 = G$ and $Y_0 = D$ are the input of *minH*. Then at line 2 of *minH*, we have $\mathcal{U} = \{\rho^1, \rho^2\}$, where $\rho^1 = \langle \{N_G \rightsquigarrow N_D\}, N_G \overset{h}{\rightsquigarrow} N_D \rangle$ is H-inference obtained by *CompleteH* (line 3–12) and $\rho^2 = \langle \{e_0, e_1, e_8\}, N_G \rightsquigarrow N_D \rangle$ is produced from regular paths obtained by *CompleteH* (line 13–17). Let us denote $\bar{p}_0:e_0$, $\bar{p}_1:e_1$, $\bar{p}_2:e_8$ as answer literals and $p_3:N_G \rightsquigarrow N_D$, $p_4:N_G \overset{h}{\rightsquigarrow} N_D$. Then $\text{clauses}(\mathcal{U}) = \{\neg p_3 \vee p_4, \neg \bar{p}_0 \vee \neg \bar{p}_1 \vee \neg \bar{p}_2 \vee p_3\}$.

By resolution over $\text{clauses}(\mathcal{U})$, we obtain $\text{min_hpaths} = \{\{e_0, e_1, e_8\}\}$ at line 3 of *minH*. Then the output of *minH* is $\mathcal{J} = \{\{a_1, a_6, a_7\}\}$, which is the set of all justifications for $G \sqsubseteq D$.

⁴ Available at <https://github.com/liveontologies/pinpointing-experiments>.

Algorithm 3: trace_one_turn

```

input :  $N_X \overset{h}{\rightsquigarrow} N_Y$ 
output: the set result of all H-inferences whose conclusion is  $N_X \overset{h}{\rightsquigarrow} N_Y$ .

1 result  $\leftarrow \emptyset$ ;
2  $\mathcal{P}_1(X, Y) \leftarrow \{(\{N_{B_1}, \dots, N_{B_m}\}, \{N_A\}) \in \mathcal{E}^{\mathcal{O}} \mid \mathcal{O} \models X \sqsubseteq A \sqsubseteq Y\}$ ;
3 for  $(\{N_{B_1}, \dots, N_{B_m}\}, \{N_A\}) \in \mathcal{P}_1(X, Y)$  do
4   if  $\text{path}(N_A, N_Y) \neq \emptyset$  or  $Y = A$  then
5     result.add $((\{N_X \overset{h}{\rightsquigarrow} N_{B_1}, \dots, N_X \overset{h}{\rightsquigarrow} N_{B_m}, N_A \rightsquigarrow N_Y\}, N_X \overset{h}{\rightsquigarrow} N_Y))$ ;
6   end
7 end

8  $\mathcal{P}_2(X, Y) \leftarrow \{(r, B_1, B_2) \mid \mathcal{O} \models X \sqsubseteq \exists r. B_1, B_1 \sqsubseteq B_2, \exists r. B_2 \sqsubseteq Y\}$ ;
9 for  $(r, B_1, B_2) \in \mathcal{P}_2(X, Y)$  do
10  if  $\text{path}(N_{\exists r. B_2}, N_Y) \neq \emptyset$  or  $Y = \exists r. B_2$  then
11    result.add $((\{N_X \overset{h}{\rightsquigarrow} N_{\exists r. B_1}, N_{B_1} \overset{h}{\rightsquigarrow} N_{B_2}, N_{\exists r. B_2} \rightsquigarrow N_Y\}, N_X \overset{h}{\rightsquigarrow} N_Y))$ ;
12  end
13 end

14  $\mathcal{P}_3(X, Y) \leftarrow \{(r, s, t, A_1, A_2) \mid r \circ s \sqsubseteq t \in \mathcal{O}, \mathcal{O} \models X \sqsubseteq \exists r. A_1, A_1 \sqsubseteq \exists s. A_2, \exists t. A_2 \sqsubseteq Y\}$ ;
15 for  $(r, s, t, A_1, A_2) \in \mathcal{P}_3(X, Y)$  do
16  if  $\text{path}(N_{\exists t. A_2}, N_Y) \neq \emptyset$  or  $Y = \exists t. A_2$  then
17    result.add $((\{N_X \overset{h}{\rightsquigarrow} N_{\exists r. A_1}, N_{A_1} \overset{h}{\rightsquigarrow} N_{\exists s. A_2}, N_{\exists t. A_2} \rightsquigarrow N_Y, (\{N_r, N_t\}, \{N_s, N_t\}), \{N_s, N_t\}\}, N_X \overset{h}{\rightsquigarrow} N_Y))$ ;
18  end
19 end

```

4.3 Optimization

Below we present two optimizations that have been implemented in order to accelerate the computation of all justifications.

1. In Algorithm 3, for the H-inference added at Line 5, we require that there exists at least one regular path from N_A to N_Y that does not contain an edge $e_i = (\{N_A\}, \{N_{B_i}\})$ for some $1 \leq i \leq m$. Otherwise, as shown in Fig. 4, H-paths corresponding to this H-inference are not minimal, as they all contain one H-path from N_X to N_Y of the form $H_{X, B_i} \cup (P_{A, Y} - \{e_i\})$. In the same spirit, we require that the H-path from N_X to N_{B_i} does not pass by N_A .
2. If we have an H-path $H_{A, B} = H_{A, \exists r. B_1} \cup H_{B_1, B_2} \cup P_{\exists r. B_2, B}$ where

$$H_{A, \exists r. B_1} = H_{A, \exists r. C} \cup H_{C, B_1}. \quad (2)$$

then $H_{C, B_2} = H_{C, B_1} \cup H_{B_1, B_2}$ is also an H-path and $H_{A, B} = H_{A, \exists r. C} \cup H_{C, B_2} \cup P_{\exists r. B_2, B}$. The two different ways to decompose $H_{A, B}$ above are already considered in Line 8 when executing Algorithm 3 with the input $N_A \overset{h}{\rightsquigarrow} N_B$. It means that the decomposition (2) is redundant. We can avoid such redundancy by requiring $\exists r. B_2 \neq Y$ at Line 11.

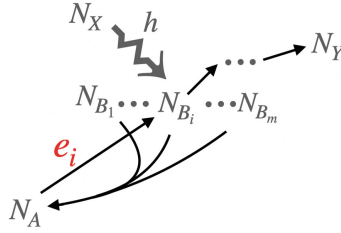


Fig. 4. Illustration of Optimization 1

5 Experiments

To evaluate and validate our approach, we compare minH^5 with PULi [14], the state-of-the-art algorithm for computing justifications at this moment. Both methods compute all justifications based on resolution but with different inference rules generated in different ways. PULi uses a complete set (next denoted by elk) generated by the ELK reasoner [13], which uses inference rules slightly different from those in Table 1. Our method uses the complete set \mathcal{U} generated by Step 1 of minH , described in Sect. 4.2. To analyze the performance of our setting, we make the following two measures: (1) we compare the size of elk with that of \mathcal{U} , (2) we compare the time cost of PULi with that of minH . All the experiments were conducted on a machine with an INTEL Xeon 2.6 GHz and 128 GiB of RAM.

The experiments were processed with four different ontologies⁶: go-plus, galen7, SnomedCT (version Jan. 2015 and Jan. 2021). All the non- \mathcal{EL}^+ axioms are deleted. Here, go-plus, galen7 are the same ontologies used in [14]. We denote the four ontologies above by go-plus, galen7, snt2015 and snt2021. The number of axioms, concepts, relations, and queries for each ontology are shown in Table 3.

Next a **query** refers to a *direct subsumption*⁷ $A \sqsubseteq B$. In our experiments, for the four ontologies, the set of all justifications $J_{\mathcal{O}}(A \sqsubseteq B)$ is computed for each query $A \sqsubseteq B$. A query $A \sqsubseteq B$ is called **trivial** iff all minimal H-paths from N_A to N_B are regular paths, otherwise, the query is **non-trivial**.

Comparing Complete Sets: \mathcal{U} vs. elk . We summarize our results in Table 4 and Fig. 5. Table 4 shows that on all four ontologies, \mathcal{U} is much smaller than elk on average. Especially on galen7, the difference between elk and \mathcal{U} is even up to 50 times. The gap is even more significant for the median value since a large part of the queries is trivial. However, the gap is much smaller for the maximal number. On snt2021, the largest \mathcal{U} in size is three times larger than that of elk .

⁵ A prototype is available at <https://gitlab.lisn.upsaclay.fr/yang/minH>.

⁶ Available at <https://osf.io/9sj8n/>, <https://www.snomed.org/>.

⁷ i.e., $\mathcal{O} \models A \sqsubseteq B$ and there is no other atomic concept A' such that $\mathcal{O} \models A \sqsubseteq A'$, $A' \sqsubseteq B$. Direct subsumptions can be computed by a reasoner supporting ontology classification.

Table 3. Summary of sizes of the input ontologies.

	go-plus	galen7	snt2015	snt2021
#axioms	105557	44475	311466	362638
#concepts	57173	28482	311480	361226
#roles	157	964	58	132
#queries	90443	91332	461854	566797

Table 4. Summary of size of elk , \mathcal{U} .

		go-plus	galen7	snt2015	snt2021
elk	average	166.9	3602.0	114.7	67.3
	median	43.0	3648.0	10.0	31.0
	max	7919.0	81501.0	2357	2226
\mathcal{U}	average	34.2	74.6	29.4	19.4
	median	4.0	5.0	1.0	3.0
	max	7772	24103	2002	6452
#non-trivial query		50272	62470	195082	304321

In Fig. 5, for a given query, if the complete set elk contains fewer inference rules than \mathcal{U} , the corresponding blue point is below the red line. The percentage of such cases are: 0.34% for go-plus, 0.066% for galen7, 0.79% for snt2015, and 1.01% for snt2021. It means that for most of the queries, the corresponding \mathcal{U} is smaller than elk .

As shown in Table 4 and in Fig. 5, sometimes minH generates bigger complete set \mathcal{U} than PULi. It may happen when, for example, there might be exponentially many different regular paths occurring in the computation process of minH . Therefore, minH could produce a huge complete set. Also, \mathcal{U} can be bigger than elk when all the regular paths involved are simple. For example, if all regular paths contain only one edge, then the complete set \mathcal{U} includes many clauses of the form $\neg p_e \vee p_{N_A \rightsquigarrow N_B}$, which happens because H-rules use regular paths. Indeed, the clause $\neg p_e \vee p_{N_A \rightsquigarrow N_B}$ is redundant since we can omit this clause by replacing $p_{N_A \rightsquigarrow N_B}$ by p_e . For elk , this does not happen.

Comparing Time Cost: minH vs. PULi. In the following, we only compare the time cost on non-trivial queries. For trivial queries, all H-path are regular paths. Thus all the justifications have already been enumerated by path in minH . It is also easy to compute all the justifications for trivial queries for PULi.

We set a limit of 60s for each query. The timed-out queries contribute of 60s to the total time cost. To compare minH with PULi, we test all three different strategies, *threshold*, *top down*, *bottom up* of the resolution algorithm proposed in [14]. We summarize in Table 5 the total time cost (top) and the timed-out queries (bottom). Figure 6 gives the comparisons over queries that are successful for both minH and PULi.

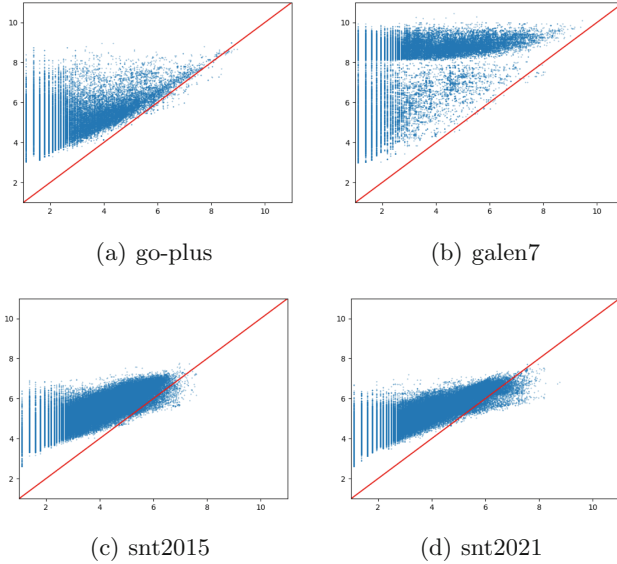


Fig. 5. Each blue point has coordinate $(\log(\#\mathcal{U}), \log(\#\text{elk}))$, where \mathcal{U}, elk are generated from a non-trivial query, the red line is $x = y$. (Color figure online)

As shown in Table 5, when using the threshold strategy, **minH** is more time consuming in total (+5%) on **snt2021**, and **minH** has more timed-out queries than **PULi** on **snt2015** and **snt2021**. This is in part due to the fact that \mathcal{U} is larger than elk for relatively many queries on **snt2015** and **snt2021** as shown in Fig. 5. For the remaining 11 cases, **minH** performs better than **PULi** in terms of total time cost and the number of timed-out queries. Especially on **galen7**, the gap between the two methods is even up to ten times for the total time cost. We can see from Table 5 that the threshold strategy performs the best for **PULi** on all four ontologies. This strategy is also the best strategy for **minH** except for **galen7**, for which the *bottom up* strategy is the best with **minH**.

For each strategy detailed in Fig. 6, the black curve (the ordered time costs of **minH** on successful queries) is always below the red curve (the ordered time costs of **PULi** on successful queries) for all the ontologies. This suggests that **minH** spends less time over successful queries. Also, most of the green points are below the red lines, which suggests that **minH** performs better than **PULi** most of the time for a given query. In some cases, we can see that **PULi** is more efficient than **minH**. One of the reasons might be as follows. Note that when computing justifications by resolution, we have to compare two different clauses and delete the redundant one (i.e., the non-minimal one). When regular paths are big, **minH** might be time consuming because of these comparisons.

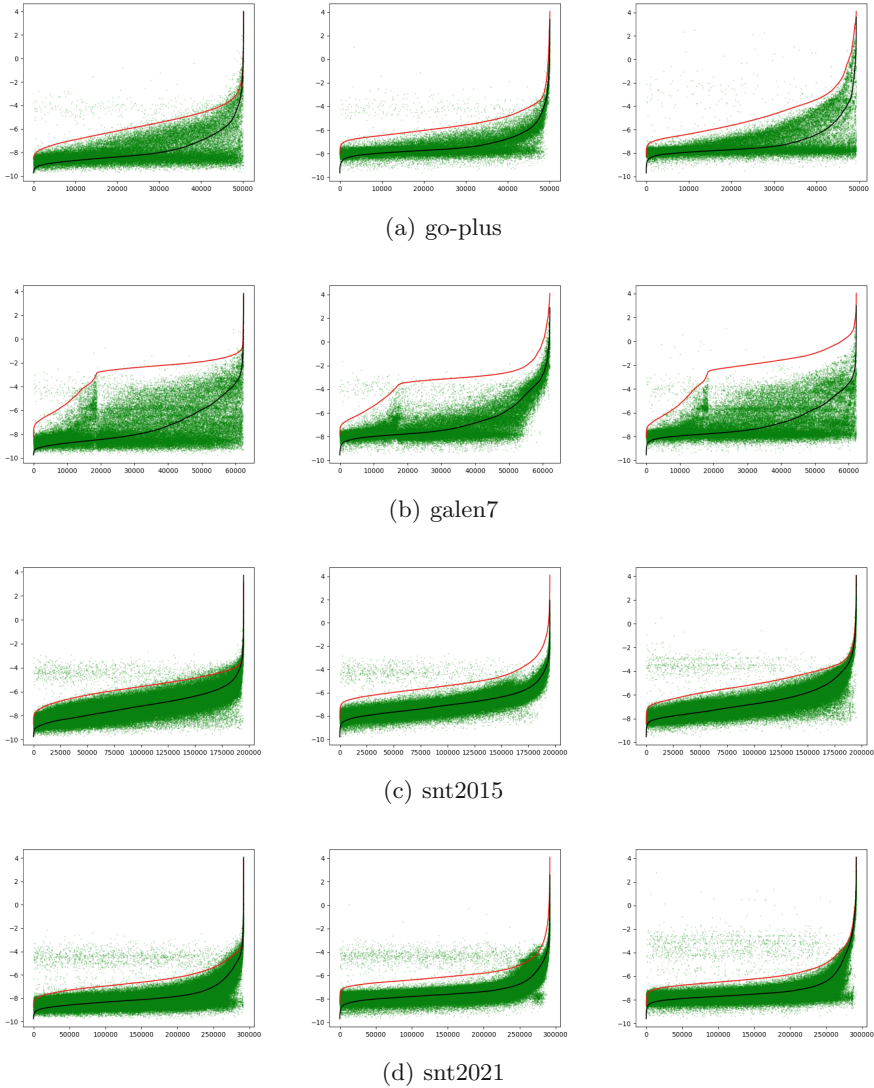


Fig. 6. For each line, the left, middle and right charts correspond to *threshold*, *top down*, *bottom up* strategies respectively. The y-axis is the log value of time(s). The red (resp. black) curve presents the ascending ordered (log value of) time cost of PULi (resp. minH). For a green point (x, y) , e^y is the time cost of minH for the query corresponding to the red line point (x, y') . (Color figure online)

Table 5. Total time cost and number of timed-out queries.

		<i>threshold</i>	<i>top down</i>	<i>bottom up</i>
total times(s) (PULi/minH)	go-plus	8482.7/ 7350.3	16352.3/ 8935.6	73629.1/ 17950.9
	galen7	10796.2/ 3681.4	43372.9/ 10607.9	36300.9/ 3156.3
	snt2015	1956.8/ 973.5	13650.7/ 1107.6	15058.3/ 11392.2
	snt2021	2116.1 /2222.6	11573.9/ 2361.6	19402.1/ 17154.9
timed-out queries (PULi/minH/both)	go-plus	116/ 103 /93	202/ 117 /114	935/ 223 /223
	galen7	48/ 43 /43	370/ 123 /120	228/ 38 /38
	snt2015	0 /3/0	49/ 3 /3	96/ 88 /83
	snt2021	2 /8/1	39/ 9 /9	144/ 133 /128

6 Conclusion

In this paper, we introduce and investigate a new set of sound and complete inference rules based on a hypergraph representation of ontologies. We design the algorithm `minH` that leverages these inference rules to compute all justifications for a given conclusion. The key of the performance of our method is that regular paths are used as elementary components of H-paths and this leads to reducing the size of complete sets because (1) rules are more compact than standard ones, (2) redundant inferences are captured and eliminated by regular paths (see Sect. 4.3). The efficiency of the algorithm `minH` has been validated by our experiments showing that it outperforms PULi in most of the cases.

There are still many possible extensions and applications of the hypergraph approach. For instance, to get even more compact inference rules, we could extend the notion of regular path to a more general one that will encapsulate the inference rule \mathcal{H}_2 in the same way as regular paths are encapsulated in H-rules. Moreover, we will try to apply our approach for other tasks like classification and to compute logical differences [15].

References

1. Arif, M.F., Mencía, C., Ignatiev, A., Manthey, N., Peñaloza, R., Marques-Silva, J.: BEACON: an efficient SAT-based tool for debugging \mathcal{EL}^+ ontologies. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 521–530. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_32
2. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient axiom pinpointing with EL2MCS. In: Hölldobler, S., Krötzsch, M., Peñaloza, R., Rudolph, S. (eds.) KI 2015. LNCS (LNAI), vol. 9324, pp. 225–233. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24489-1_17
3. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient MUS enumeration of horn formulae with applications to axiom pinpointing. In: Heule, M., Weaver, S. (eds.) SAT 2015. LNCS, vol. 9340, pp. 324–342. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24318-4_24
4. Ausiello, G., Laura, L.: Directed hypergraphs: introduction and fundamental algorithms—a survey. *Theor. Comput. Sci.* **658**, 293–306 (2017)

5. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: IJCAI, vol. 5, pp. 364–369 (2005)
6. Chen, J., Ludwig, M., Ma, Y., Walther, D.: Zooming in on ontologies: minimal modules and best excerpts. In: d’Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 173–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_11
7. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. *Discret. Appl. Math.* **42**(2–3), 177–201 (1993)
8. Grau, B.C., Horrocks, L., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: theory and practice. *J. Artif. Intell. Res.* **31**, 273–318 (2008)
9. Ignatiev, A., Marques-Silva, J., Mencía, C., Peñaloza, R.: Debugging EL+ ontologies through horn MUS enumeration. In: Artale, A., Glimm, B., Kontchakov, R. (eds.) Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, 18–21 July 2017. CEUR Workshop Proceedings, vol. 1879. CEUR-WS.org (2017). <http://ceur-ws.org/Vol-1879/paper54.pdf>
10. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 267–280. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_20
11. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging unsatisfiable classes in OWL ontologies. *J. Web Semant.* **3**(4), 268–293 (2005)
12. Kazakov, Y., Klinov, P.: Goal-directed tracing of inferences in EL ontologies. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 196–211. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_13
13. Kazakov, Y., Krötzsch, M., Simancik, F.: ELK reasoner: architecture and evaluation. In: ORE (2012)
14. Kazakov, Y., Skočovský, P.: Enumerating justifications using resolution. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) IJCAR 2018. LNCS (LNAI), vol. 10900, pp. 609–626. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94205-6_40
15. Ludwig, M., Walther, D.: The logical difference for \mathcal{ELH}^r -terminologies using hypergraphs. In: Schaub, T., Friedrich, G., O’Sullivan, B. (eds.) 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, 18–22 August 2014 - Including Prestigious Applications of Intelligent Systems (PAIS 2014). *Frontiers in Artificial Intelligence and Applications*, vol. 263, pp. 555–560. IOS Press (2014). <https://doi.org/10.3233/978-1-61499-419-0-555>
16. Manthey, N., Peñaloza, R., Rudolph, S.: Efficient axiom pinpointing in EL using sat technology. In: *Description Logics* (2016)
17. Peñaloza, R.: Axiom pinpointing. In: Cota, G., Daquino, M., Pozzato, G.L. (eds.) *Applications and Practices in Ontology Design, Extraction, and Reasoning, Studies on the Semantic Web*, vol. 49, pp. 162–177. IOS Press (2020). <https://doi.org/10.3233/SSW200042>
18. Penaloza, R., Sertkaya, B.: Understanding the complexity of axiom pinpointing in lightweight description logics. *Artif. Intell.* **250**, 80–104 (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

