A Hypergraph Approach for Logic-based Abduction

Qiancheng Ouyang
LISN, CNRS
Université Paris-Saclay
email: oyqc@lisn.fr

Tianjiao Dai
LISN, CNRS
Université Paris-Saclay
email: dai@lisn.fr

Yue Ma
LISN, CNRS
Université Paris-Saclay
email: yue.ma@lisn.fr

***Abstract*—Abduction reasoning, which finds possible hypotheses from existing observations, has been studied in many different areas. We consider an abduction problem that takes into account a user's interest. We propose a new approach to solving such an abduction problem based on a hypergraph representation of an ontology and obtain a linear algorithm for a description logic.**

***Index Terms*—Abduction; Hypergraph; Description Logic**

I. Introduction

Abduction reasoning aims to generate a possible hypothesis for a given observation. Abduction has been applied in many artificial intelligence (AI) areas, such as machine learning, logical programming, and statistical relational AI [7].

We focus on *logical-based abduction* [4] over *description logic ontologies*. Here, ontologies consist of *axioms* that state the relationship of different *concepts* and *relationships* over a specific domain. Then, our abduction problem consists of three parts: (i) a given background knowledge (i.e., an existing ontology $\mathcal{O}$); (ii) a set of hypotheses (i.e., a set of axioms $\mathcal{H}$) and (iii) a given conclusion (i.e., a single axiom). There have been many studies of abduction over different ontologies, such as the complexity of abduction over $\mathcal{EL}$ [2] and their application to repairing ontologies [8], abduction over $\mathcal{EL}$ by translation to first-order logic [5], forgetting-based abductive reasoning over expressive ontology $\mathcal{ALC}$ [3], and signature-based abduction over more expressive $\mathcal{ALCOI}\mu$ [6].

We propose a new solution (section IV) of abduction over a special $\mathcal{EL}$-ontology (free of role restrictions) in Section III, based on a hypergraph representation of ontologies.

II. Preliminaries

An *ontology* $\mathcal{O}$ is a set of *axioms* of the form $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$, where A_i, B are called *concepts*. An interpretation $\mathcal{I} = \langle \Delta^I, \cdot^I \rangle$ consists of a non-empty domain Δ^I and a mapping $\cdot^I$ that maps each concept to a subset $A^I \subseteq \Delta^I$. A model of $\mathcal{O}$ is an interpretation that for each $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{O}$, we have $A_1^{\mathcal{I}} \cap \cdots \cap A_n^{\mathcal{I}} \subseteq A^{\mathcal{I}}$. We say $\mathcal{O} \models A'_1 \sqcap \cdots \sqcap A'_n \sqsubseteq B'$ iff for any models $\mathcal{I}$ of $\mathcal{O}$, we have $(A'_1)^{\mathcal{I}} \cap \cdots \cap (A'_n)^{\mathcal{I}} \subseteq (B')^{\mathcal{I}}$.

A *(directed) hypergraph* $\mathcal{H} = \{\mathcal{V}, \mathcal{E}\}$ consists of a node set $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ and a hyperedge set $\mathcal{E} = \{e_1, e_2 \cdots, e_m\}$, where $e_i = \langle T(e_i), f(e_i) \rangle$ with $T(e_i) \subseteq \mathcal{V}$ being a subset and $f(e_i) \in \mathcal{V}$ being a node. Note that a classical hyperedge can have multiple nodes in its head, which we require to be a singleton for computing abduction.

Definition 1 ([1])**.** Given a hypergraph $\mathcal{H} = \{\mathcal{V}, \mathcal{E}\}$, assume $S \subseteq \mathcal{V}$ and $v \in \mathcal{V}$. A *hyperpath* from S to v is a sequence $h = [e_1, e_2, \cdots, e_n]$ of hyperedges such that (i) $f(e_n) = \{v\}$; (ii) for $i = 1, \cdots, n$, $T(e_i) \subseteq S \cup \{f(e_1); \cdots, f(e_{i-1})\}$; (iii) for $i = 1, \cdots, n$, $f(e_i) \in \bigcup_{i<j\leq n} T(e_j)$.

III. Abduction Problem

We consider an abduction problem that takes into account a user's interests represented by a set of concepts Σ.

Definition 2. An abduction problem is a tuple

$$\langle \mathcal{O},\ \Sigma,\ A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \rangle,$$

where $\Sigma = \{A', B', \cdots\}$ is a set of concept names. A *solution* of this problem is a (minimal) ontology

$$\mathcal{H} = \{A'_1 \sqcap \cdots \sqcap A'_n \sqsubseteq B' \mid A'_i, B' \in \Sigma, n \geq 0\}$$

such that $\mathcal{O} \cup \mathcal{H} \models A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$. A solution $\mathcal{H}$ is called a *hypothesis* with respect to Σ.

Example 1. Let an ontology $\mathcal{O}_0$ be:

$$peopleWithDiploma \sqsubseteq doctor$$
$$peopleHasPaper \sqsubseteq researcher$$
$$doctor \sqcap employeeWithUniversityChair \sqsubseteq professor$$

$\mathcal{O}_0$ can not derive the following axiom α_0:

$$\alpha_0 : doctor \sqcap employeeWithUniversityChair \sqsubseteq researcher$$

although it should be true. Consider $\Sigma_0 = \{professor,\ peopleHasPaper\}$. If we add a hypothesis $\mathcal{H}_0 = \{professor \sqsubseteq peopleHasPaper\}$, we have $\mathcal{O}_0 \cup \mathcal{H}_0 \models \alpha_0$. Therefore, $\mathcal{H}_0$ is a solution of the abduction problem $\mathcal{A}_0 = \langle \mathcal{O}_0,\ \Sigma_0,\ \alpha_0 \rangle$. It is clear that $\mathcal{H}_0$ is also a minimal solution to the abduction problem. But there is no solution to $\mathcal{A}_0$ if $\Sigma_0 = \{professor,\ peopleWithDiploma\}$.

IV. A Hypergraph-based Algorithm

We now present a method of finding a (minimal) solution to the abduction problem using hypergraphs.

Definition 3. For each set $\mathcal{O}$ of axioms, we define a hypergraph $H_{\mathcal{O}} = (\mathcal{N}_h, \mathcal{E}_h)$, where $\mathcal{N}_h := \{N_{A'} \mid A' \in \mathsf{N_C}\}$ and

$$\mathcal{E}_h := \{\{N_{A'_1}, \cdots, N_{A'_n}\} \rightarrow N_{A'} \mid A'_1 \sqcap \cdots \sqcap A'_n \sqsubseteq A' \in \mathcal{O}\}$$

Example 2 (Example 1 cont'd)**.** By definition, the hypergraph $H_{\mathcal{O}_0}$ of $\mathcal{O}_0$ is shown in Figure. 1. Now, we add an edge

 ISBN: 978-1-68558-056-8

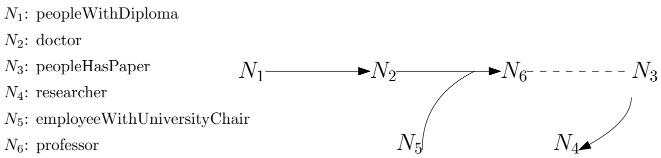

Fig. 1: The hypergraph representation $H_{\mathcal{O}_0}$ of $\mathcal{O}_0$ in Example 1

$\{N_6\}\to N_3$ to the hypergraph $H_{\mathcal{O}_0}$. Then, we can find a hyperpath h from $\{N_2,\ N_5\}$ to N_4:

$$h = [\{N_2, N_5\}\to N_6,\ \{N_6\}\to N_3,\ \{N_3\}\to N_4]$$

Theorem 1. Given an ontology $\mathcal{O}$ and its associated hypergraph $H_{\mathcal{O}}$, an ontology $\mathcal{H}$ is a (minimal) solution to the abduction problem $\langle \mathcal{O}, \Sigma, A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B\rangle$ iff $H_{\mathcal{H}}$ is a (minimal) hypergraph such that (i) All nodes in $H_{\mathcal{H}}$ are of the form $N_A,\ A \in \Sigma$, and (ii) There exists a hyperpath from $N_{A_1}, \cdots, N_{A_n}$ to N_B in $H_{\mathcal{O}} \cup H_{\mathcal{H}}$.

Example 3 (Example 1 cont'd)**.** By Theorem 1, to solve the abduction problem $\mathcal{A}_0$, it is enough to find an $H_{\mathcal{H}}$ such that there exists a hyperpath from $\{N_2, N_5\}$ to N_4 in $H_{\mathcal{O}_0} \cup H_{\mathcal{H}}$. The hypergraph $H_{\mathcal{H}}$ consists of a single edge $\{N_6\}\to N_3$ satisfying the requirement, leading to the hyperpath given in Example 2 as the minimal solution of the problem.

Before stating our main Algorithm 2, we define a property of *saturation* for a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and $V \subset \mathcal{V}$. We define $U \subset V$ to be *saturated* (under V) if there exists $e \in \mathcal{E}$ such that $T(e) = U$ and $f(e) \notin V$. For example, in Fig. 1, if we have $V = \{N_1, N_2, N_5\}$, then $\{N_1\}$ and $\{N_2, N_5\}$ are saturated under V, while other subsets of V are not. Algorithm 1 finds all vertices approachable from V in run-time $O(|\mathcal{E}|)$.

Proposition 1. For a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and $V \subset \mathcal{V}$, $v \in Span(\mathcal{V}, \mathcal{E}, V)$ iff. there is a hyperpath from V to v.

Algorithm 1: $Span(\mathcal{V}, \mathcal{E}, V)$

input : hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, set $V \subset \mathcal{V}$
output: $W \subset \mathcal{V}$ of all vertices spanned from V
$W = V$.
$\mathcal{U} = \{U,\ U$ is saturated under $V\}$.
while $\mathcal{U} \neq \emptyset$ **do**
  choose $U \in \mathcal{U}$,
  while *there exists* $v \in \mathcal{V} \setminus W$ *such that* $(W, v) \in \mathcal{E}$ **do**
    put v into W;
    put all saturated sets containing v into $\mathcal{U}$.
  end
  remove U.
end
return W

In Algorithm 2, we first check if v can be directly reached by V (Line 1-3). Then, we check if the aiming hypergraph exists (Line 4-11). These 2 steps have run-time $O(|\mathcal{E}|)$. In the minimizing step, for each $e \in \mathcal{E}'$, we check only once if e can be deleted. Hence, the total run-time is $O(|\Sigma||\mathcal{E}|)$.

Algorithm 2:

input : hypergraph $H = (\mathcal{V}, \mathcal{E})$, $\Sigma \subset \mathcal{V}$, $S \subset \mathcal{V}$, $v \in \mathcal{V}$
output: hypergraph $\mathcal{H}$ on Σ
1 $V = Span(\mathcal{V}, \mathcal{E}, S)$.
2 **if** $v \in V$ **then**
3   **return** empty graph.
4 **else**
5   **if** $\Sigma \subset V$ or $\Sigma \cup V = \emptyset$ **return** *non-existence.*
6   $\Sigma \setminus V = \{v_1, \ldots, v_m\}$,
7   choose m hyper-edges $\mathcal{E}' = \{e_1, \ldots, e_m\}$ where $T(e_i) \subset \Sigma \cap V$ and $f(e_i) = v_i$ for $1 \leq i \leq m$.
8   $V = V \cup \Sigma$.
9 **end**
10 **if** $v \notin Span(\mathcal{V}, \mathcal{E} \cup \mathcal{E}', V)$ **then**
11   **return** non-existence.
12 **else**
13   minimize $\mathcal{E}'$ (check if there exists $e \in \mathcal{E}'$ such that $\mathcal{E}' - e$ satisfies until we get a minimal size).
14   **return** $\mathcal{H} = (\Sigma, \mathcal{E}')$.
15 **end**

We explain Algorithm 2 via the following example.

Solution of Example 3. (via Algorithm 2)

1) $H = H_{\mathcal{O}}, \Sigma = \{N_3, N_6\}, S = \{N_2, N_5\}, v = N_4$.
2) Line 1: $V = Span(\mathcal{V}, \mathcal{E}, S) = \{N_2, N_5, N_6\}$.
3) Line 7: $\mathcal{E}' = \{\{N_6\} \to N_3\}$.
4) Line 10: $v \in Span(\mathcal{V}, \mathcal{E} \cup \mathcal{E}', V = \mathcal{V})$.
5) Line 13: we see $\{\{N_6\} \to N_3\}$ cannot be deleted. It returns $\mathcal{H} = (\Sigma, \mathcal{E}')$.

Theorem 2. For Algorithm 2, the output $\mathcal{H}$ is a minimal hypergraph satisfying the conditions (i) and (ii) in Theorem 1.

V. Conclusion

In this work, we introduce a hypergraph-based algorithm for solving abduction problems over $\mathcal{EL}$-ontologies that do not have role restrictions, which have a linear time complexity w.r.t. the size of the input ontology. As for future work, we plan to implement our algorithm and extend it to handle general $\mathcal{EL}$-ontologies with role restrictions, as well as more expressive ontologies such as $\mathcal{ALC}$.

Acknowledgment. We thank Hui Yang for bringing our attention to the topic and the discussion with us.

References

[1] A. Giorgio and L. Luigi, "Directed hypergraphs: Introduction and fundamental algorithms-a survey", Theoretical Computer Science, vol. 658, pp. 293–306, 2017.

[2] M. Bienvenu, "Complexity of abduction in the EL family of lightweight description logics", Proc. of KR'08, 2008, pp. 220–230.

[3] W. Del-Pinto and R. A. Schmidt, "Abox abduction via forgetting in ALC", Proc. of AAAI'19, 2019, pp. 2768–2775.

[4] T. Eiter and G. Gottlob, "The complexity of logic-based abduction", J. ACM, vol. 42 (1), 1995, pp. 3–42.

[5] F. Haifani, P. Koopmann, S. Tourret, and C. Weidenbach, "Connection-minimal abduction in *EL* via translation to FOL", Proc. of IJCAR'22, 2022, pp. 188–207.

[6] P. Koopmann, W. Del-Pinto, S. Tourret, and R. A. Schmidt "Signature-based abduction for expressive description logics", Proc. of KR'20, 2020, pp. 592–602.
[7] Sindhu V. Raghavan, "Bayesian Abductive Logic Programs: A Probabilistic Logic for Abductive Reasoning", Statistical Relational Artificial Intelligence, Proc. of IJCAI'11, 2011, pp. 2840–2841.
[8] F. Wei-Kleiner, Z. Dragisic, and P. Lambrix, "Abduction framework for repairing incomplete EL ontologies: Complexity results and algorithms", Proc. of AAAI'14, 2014, pp. 1120–1127.