Merge, Explain, Iterate: A Combination of MHS and MXP in an ABox Abduction Solver*

Extended Abstract

Janka Boborová, Iveta Balintová, Júlia Pukancová and Martin Homola

Comenius University in Bratislava, Mlynská dolina, 842 41 Bratislava, Slovakia

Abstract

MHS-MXP is an ABox abduction algorithm for DL, that leverages the divide-and-conquer strategy of the effective but incomplete MergeXplain and jointly uses the Minimal Hitting Set algorithm (MHS) to track the search-space exploration in order to ensure completeness. In this extended abstract we focus on our updated implementation which achieved a black-box combination of the abduction component and the JFact DL reasoner via the `OWLKnowledgeExplorerReasoner` interface of OWL API. We then present the results of our empirical evaluation showing that, compared to MHS, MHS-MXP has a significant advantage on favourable inputs, however, it does not perform as well as MHS on unfavourable inputs.

Keywords

abduction, description logics, ontologies

1. Introduction

ABox abduction [2] starts from a background ontology and an observation in the form of an ABox assertion, that is not entailed. The task is to find explanations – sets of ABox assertions, such that the background ontology and the explanation together entail the observation. Optionally, abducibles are a set of ABox assertions from which the explanations may be drawn.

MHS-MXP is a "hybrid" abduction algorithm, that leverages the divide-and-conquer strategy of the effective but yet incomplete MergeXplain [3] and jointly uses the Minimal Hitting Set algorithm (MHS) [4] to track the search-space exploration in order to ensure completeness. It was developed as an ABox abduction approach and implemented into a solver [5].

We report on an updated implementation which fixes a serious flaw of the previous version, namely in the integration of the external DL reasoner. The flaw greatly limited the inputs the solver could correctly handle. MHS and MHS-MXP call an external DL reasoner for consistency checking and obtaining certain information about the model (specifically, the set of all atomic concept and role assertions that are true in the model); this task is referred to as "model

DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece

*The full version [1] of this paper will appear at the *18th European Conference on Logics in Artificial Intelligence (JELIA 2023)*, Dresden, Germany, Sep 20–22, 2023

boborova3@uniba.sk (J. Boborová); ivbalintova@gmail.com (I. Balintová); pukancova@fmph.uniba.sk (J. Pukancová); homola@fmph.uniba.sk (M. Homola)

https://dai.fmph.uniba.sk/~pukancova/ (J. Pukancová); https://dai.fmph.uniba.sk/~homola/ (M. Homola)

0000-0001-6384-9771 (M. Homola)

extraction". It is, however, highly non-standard for DL reasoners to output model information (some of then may not even construct a model, e.g. consequence-based reasoners such as Elk [6]).

In our current implementation, model extraction from the JFact reasoner is achieved via the experimental `OWLKnowledgeExplorerReasoner` interface of OWL API. We also report on an empirical evaluation that we were able to run on this implementation. It is conjectured that cases with a lower number of contradictory abducibles and those with smaller sizes of explanations and a lower overall count of explanations are favourable for MHS-MXP. It was also confirmed by our evaluation. On the other hand, on cases where the search space was clogged by large numbers of contradictory abducibles, MHS-MXP did not perform as well as MHS.

2. Implementation

An implementation[1] of the abduction solver was developed in Java. It implements the DL Abducion API [7] for integration into applications. The solver allows to choose between the MHS or MHS-MXP abduction algorithm. It supports any OWL 2 ontology as background knowledge and observations as (possibly multiple) concept or role assertions, including complex concepts and negated roles. The solver is capable to find all explanations in the class of sets of atomic and negated atomic concept and role assertions involving the named individuals from the observation or the knowledge base's ABox.

The solver relies on OWL API. DL reasoning, in the form of consistency checking and model extraction, is provided by an external DL reasoner. In the new version, we access the completion graph through an OWL API `OWLKnowledgeExplorerReasoner`[2] interface implemented by the JFact reasoner[3]. In fact, it required some adjustments to make the interface work in JFact[4].

Apart from the input ontology and the observation, the solver allows additional settings: the choice of abduction algorithm; a timeout; a depth limit for HS-tree exploration; and abducibles, either as a set of symbols or assertions. In the new version, it is also possible to toggle role assertions and role looping (role assertions of form $R(a, a)$ s.t. $R \in N_R$ and $a \in N_I$) in explanations, and different ways how relevant explanation is defined for a multiple observation.

An example input file follows, in which the ontology is loaded from `familyX.owl` and the observation is set to Father(jack). In addition, the timeout is set to 14,400 seconds, and negations and role assertions in explanations are suppressed:

```
-f: files/familyX.owl
-o: Prefix: fam: <http://www.semanticweb.org/julia/ontologies/2017/8/family\#>
    Class: fam:Father Individual: fam:jack Types: fam:Father
-t: 14400
-n: false
-r: false
```

[1]Available at https://github.com/boborova3/MHS-MXP-algorithm.
[2]https://owlcs.github.io/owlapi/apidocs_4/org/semanticweb/owlapi/reasoner/knowledgeexploration/OWLKnowledgeExplorerReasoner.html
[3]https://github.com/owlcs/jfact
[4]The version we actually used is avalable at https://github.com/boborova3/jfact/tree/test4.

Table 1
Basic characteristics of groups: #: number of observations; C_m, C_a, C_M: min, average, and max count of explanations; S_m, S_a, S_M: min, average, and max size of the largest explanation. All generated observations have explanations consisting of only atomic concept assertions, so for any observation, the set of explanations is equal in the unfavourable and the favourable case.

Set	#	C_m	C_a	C_M	S_m	S_a	S_M
S1	10	1	7	20	1	1	1
S2	10	8	69.5	159	2	2	2
S3	10	47	212.4	479	3	3	3
S4	10	251	417.8	839	4	4	4
S5	10	503	2627	6719	5	5	5

Set	#	C_m	C_a	C_M	S_m	S_a	S_M
C1	9	1	4.8	9	1	1.11	2
C2	11	14	51	99	1	2	3
C3	13	111	212.92	299	2	3.15	4
C4	8	359	524.75	839	3	4	5
C5	9	1175	2863	6719	5	5	5

The default algorithm is MHS-MXP, MHS is toggled by adding `-mhs: true` into the input file. On the given input, the solver returns 3 explanations: {Grandfather(jack)}, {GreatGrandfather(jack)} and {Male(jack), Parent(jack)}.

3. Evaluation

We evaluated[5] the implementation of MHS-MXP against MHS. We conjecture that MHS-MXP will perform better when the maximum size or the total count of explanations is lower. We used the LUBM ontology [8] to generate 50 observations with the explanations of various sizes (from 1 to 5). The observations were divided into groups S1–5 according to the size of the largest explanation and C1–5 according to the explanation count (Table 1).

The theoretical properties of MHS-MXP [5] show that the allowed abducible space may have a significant impact on its performance. We, therefore, distinguish between (a) the unfavourable case: MHS-MXP has a disadvantage if the abducibles (i.e. the search space) are clogged by a large number of mutually conflicting facts – the case was constructed by allowing for each atomic assertion also its complement as abducibles); and (b) the favourable case: when mutually conflicting abducibles are not present (the case allowed only atomic assertions as abducibles). The size of abducibles in the unfavourable and the favourable case was 86 and 43, respectively.

We observe that in the unfavourable case (Figure 1 (a,b)), both algorithms computed explanations up to size 3, but MHS-MXP achieved slightly worse results than MHS. This verifies our hypothesis that MHS-MXP is not able to handle large numbers of conflicting abducibles well.

On the other hand, in the favourable case (Figure 1 (c,d)), MHS-MXP achieved significantly better results than MHS, which performs similarly in both cases. All MHS-MXP groups except for S5 and C5 terminated successfully within the timeout. MHS-MXP managed to find all explanations. We also observe a correlation in the increase of computation time and the size of the largest explanation (groups S) and likewise for the explanation count (groups C).

[5] The evaluation was executed on a virtual machine with 8 cores (16 threads) of Intel Xeon CPU E5-2695 v4, 2.10GHz, with 32GB RAM, running Ubuntu 20.04 and Oracle Java SE Runtime Environment v1.8.0_201. Execution times were measured using `ThreadMXBean` from the `java.lang.management` package. We measured user time –- the actual time without system overhead. The maximum Java heap size to 4GB. Each input was run 5 times and the results were averaged for each group. The timeout was set to 4 hours (=14,440 seconds).

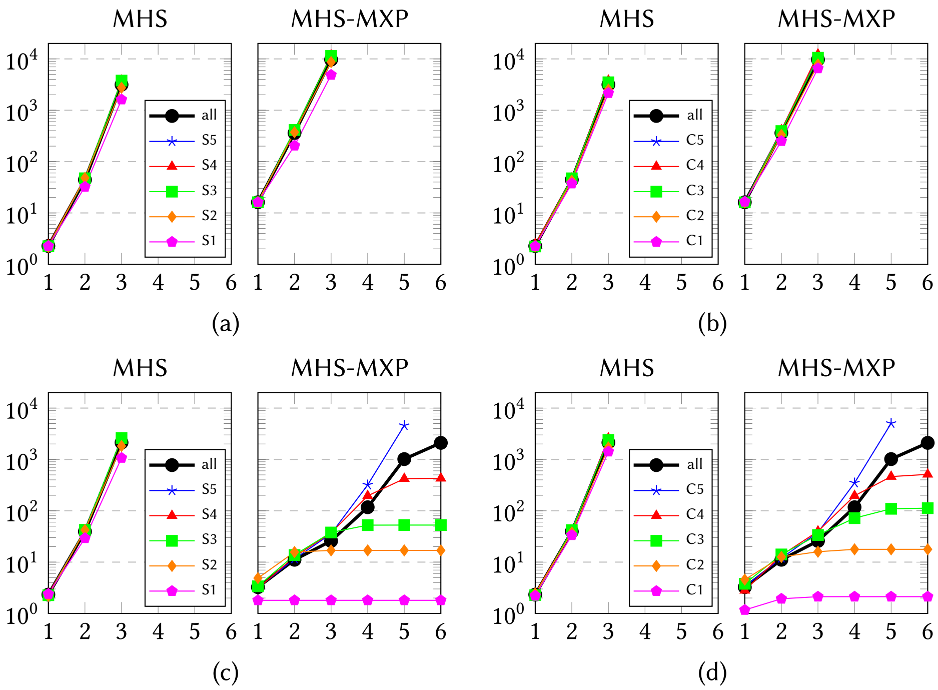

Figure 1: Evaluation results: Unfavourable case, groups (a) S1–5 and (b) C1–5; favourable case, groups (c) S1–5 and (d) C1–5. Average time in seconds (y-axis) for fully exploring the search space up to the particular explanation size (x-axis) in the input group. If more than one-third of the inputs in the group reached the timeout before reaching the x-value, the y-value was omitted from the plot.

4. Conclusions

We have reported on the current implementation of the MHS-MXP ABox abduction solver and its evaluation which demonstrated an advantage of MHS-MXP over MHS on a significant class of inputs, particularly those where abducibles can be carefully configured by a knowledgeable user (as not uncommon in real-world applications [9]). We hope that further optimization will help to bring the algorithm on par with MHS also in the unfavourable cases of inputs.

While there are currently approaches in ABox abduction [10, 11, 12, 13] which are more tractable, they are also limited in the supported DL expressivity. Notably, in our solver, we were able to achieve black-box integration with JFact which supports the full $\mathcal{SROIQ}$ DL [14], i.e. OWL 2 [15]. The implementation of the `OWLKnowledgeExplorerReasoner` interface in other DL reasoners will make them accessible for such integration with MHS and derived algorithms.

Acknowledgments

This research was sponsored by the Slovak Republic under the grant APVV-19-0220 (ORBIS) and by the EU under the H2020 grant no. 952215 (TAILOR). J. Boborová was supported by an extraordinary scholarship awarded by Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics and by a DL student grant.

References

[1] M. Homola, J. Pukancová, J. Boborová, I. Balintová, Merge, explain, iterate: A combination of MHS and MXP in an ABox abduction solver, in: Logics in Artificial Intelligence – 18th European Conference, JELIA 2023, Dresden, Germany, Sep 20–22, 2023, Proceedings, volume 14281 of *LNCS*, Springer, 2023.

[2] C. Elsenbroich, O. Kutz, U. Sattler, A case for abductive reasoning over ontologies, in: Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, Athens, GA, US, volume 216 of *CEUR-WS*, 2006.

[3] K. M. Shchekotykhin, D. Jannach, T. Schmitz, MergeXplain: Fast computation of multiple conflicts for diagnosis, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, AAAI Press, 2015.

[4] R. Reiter, A theory of diagnosis from first principles, Artificial intelligence 32 (1987) 57–95.

[5] M. Homola, J. Pukancová, I. Balintová, J. Boborová, Hybrid MHS-MXP abox abduction solver: First empirical results, in: Proceedings of the 35th International Workshop on Description Logics (DL 2022), Haifa, Israel, volume 3263 of *CEUR-WS*, 2022.

[6] Y. Kazakov, M. Krötzsch, F. Simančík, The incredible ELK, Journal of Automated Reasoning 53 (2014) 1–61.

[7] J. Kloc, M. Homola, J. Pukancová, DL abduction API v2 and GUI interface, in: Proceedings of the 36th International Workshop on Description Logics (DL 2023), Rhodes, Greece, To appear in CEUR-WS, 2023.

[8] Y. Guo, Z. Pan, J. Heflin, LUBM: A benchmark for OWL knowledge base systems, Journal of Web Semantics 3 (2005) 158–182.

[9] S. M. Rashid, J. P. McCusker, D. M. Gruen, O. Seneviratne, D. L. McGuinness, A concise ontology to support research on complex, multimodal clinical reasoning, in: The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings, volume 13870 of *LNCS*, Springer, 2023, pp. 390–407.

[10] J. Du, G. Qi, Y. Shen, J. Z. Pan, Towards practical ABox abduction in large description logic ontologies, Int. J. Semantic Web Inf. Syst. 8 (2012) 1–33.

[11] J. Du, K. Wang, Y. Shen, A tractable approach to ABox abduction over description logic ontologies, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27-31, 2014, Québec City, Québec, Canada., 2014, pp. 1034–1040.

[12] W. Del-Pinto, R. A. Schmidt, Abox abduction via forgetting in ALC, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, AAAI Press, 2019, pp. 2768–2775.

[13] P. Koopmann, W. Del-Pinto, S. Tourret, R. A. Schmidt, Signature-based abduction for expressive description logics, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, 2020, pp. 592–602.

[14] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible $\mathcal{SROIQ}$, in: Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, AAAI, 2006, pp. 57–67.

[15] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, U. Sattler, OWL 2: The next step for OWL, J. Web Semant. 6 (2008) 309–322.