EPiC Series in Computing

Volume 50, 2017, Pages 233–238

GCAI 2017. 3rd Global Conference on Artificial Intelligence

Abduction for Learning Smart City Rules

Nikolaj Bjørner[1], Maria-Cristina Marinescu[2], and Mooly Sagiv[3][4]

[1] Microsoft Research, Redmond, Washington, USA
nbjorner@microsoft.com
[2] Barcelona Supercomputing Center, Barcelona, Spain
maria.marinescu@bsc.es
[3] Tel Aviv University, Tel Aviv, Israel
msagiv@acm.org
[4] VMWare Research, Palo Alto, California, USA

Abstract

We propose using abduction for inferring implicit rules for Smart City ontologies. We show how we can use Z3 to extract candidate abducers from partial ontologies and leverage them in an iterative process of evolving an ontology by refining relations and restrictions, and populating relations. Our starting point is a Smart City initiative of the city of Barcelona, where a substantial ontology is being developed to support processes such as city planning, social services, or improving the quality of the data concerning (for instance) legal entities, whose incompleteness may sometimes hide fraudulent behavior. In our scenario we are supporting semantic queries over heterogeneous and noisy data. The approach we develop would allow evolving ontologies in an iterative fashion as new relations and restrictions are discovered.

1 Introduction

Cities are complex systems of interrelated domains that produce massive amounts of data from many sources and in many different formats. Some of this is open data, which is seldom cured. Data produced by citizens, sensors, or mobile devices, is likewise not always accurate, and may be inconsistent with other data sources. Ontologies are a flexible way to model a city and integrate the many heterogeneous data sources at a semantic level without modifying the data itself. Semantic integration via ontologies balances the ease of customization that domain specialists need with the definition of a stable, well-organized set of concepts that an inference engine can reason about. This approach makes it easier to integrate heterogeneous data sources, but these come at varying quality and granularity levels.

Arguably the main entry barrier for a semantic integration approach is the lengthy process of defining an ontology in the first place. A tool that can recommend extensions to an existing model would make ontology definition more palatable. At the same time, in our experience it is often unrealistic to define constraints over the data when this is of poor quality, as it may excessively restrict how much of it could be successfully integrated. Increasing the quality of the data would thus positively affect integration at a global level.

We propose an approach to generate new consistency rules based on an existing semantic model. To this end, we harness decision procedures and generate possible explanations for the current state of the model, via abduction. The process can support evolving the model by recommending useful constraints (the explanations) to the ontology developer and domain specialists. We can also view these as hypotheses for what can be reasonably expected from the data at the current time, which a data analyst can use as checks to explore and improve data quality. Finally, we envision this approach as a first step to satisfying queries over the city model by returning approximate answers when the ontology is not populated with the requested data.

2 Example

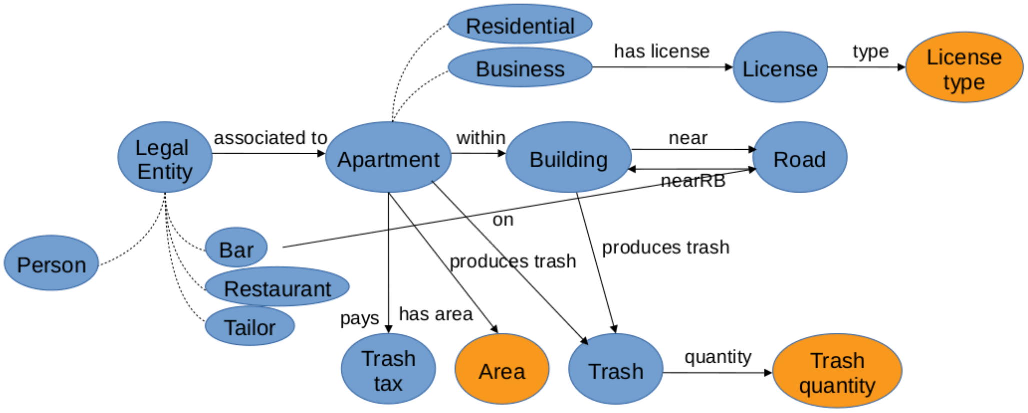

Figure 1: Partial city ontology

Figure 1 represents a partial ontology signature for the Barcelona Smart City initiative. It describes a set of sorts, represented as vertices, and binary relations between the sorts, represented as directed edges. The dashed lines show inheritance relationships, and the orange nodes (`License type`, `Area`, and `Trash quantity`) are attributes. The graph encodes information, such as, businesses and residential entities are a sub-type of apartments, and apartments relate to buildings using the `within` binary predicate. Both buildings and apartments produce trash,

$A_1 : \mathsf{Business} \subseteq \mathsf{Apartment}$
$A_2 : \forall X, Y . \mathsf{Business}(X) \wedge \mathsf{HasLicense}(X, Y) \wedge \mathsf{Restaurant}(Y) \implies \mathsf{TrashHog}(X)$
$A_3 : \forall X, Y . \mathsf{Apartment}(Y) \wedge \mathsf{TrashHog}(Y) \wedge \mathsf{Within}(Y, X) \wedge \mathsf{Building}(X) \implies \mathsf{TrashHog}(X)$

Figure 2: Ontologies for an entity to be a trash hog. A_1 defines that business as a sub-type of Apartment. A_2 indicates that restaurants produce large quantities of trash. A_3 states that a building that contains apartments that produce a lot of trash, produces a lot of trash.

which may be measured as a quantity that is usually only monitored per building. That means that although the relation `produces trash` exists between `Apartment` and `Trash` in the city model, it may not be populated with data. On the other hand, apartments (not buildings) pay trash tax and it is in the city's interest to collect trash tax fairly based on quantities that are produced from apartments. As it may well be, trash trucks collect from entire buildings and cannot enforce taxation. Our quest is to discover discrepancies, e.g., ways for the city of Barcelona to collect taxes fairly.

Suppose we observe that a building produces a substantial amount of trash. What are the possible causes of this observation? We may have created an ontology that already encodes this information directly, but more typically, this would not be the case and we would have to extract possible causes.

As an example, suppose we have encoded in the city model trash hog as shown in Fig. 2

We are given the observation $\mathsf{TrashHog}(b)$, for some entity b, for which we know $\mathsf{Building}(b)$ holds. For the given ontology, the explanation that b is a trash hog includes the case where b contains a restaurant business. The ontology supports the explanation (I) that there is some c, where $\mathsf{Apartment}(c)$, $\mathsf{TrashHog}(c)$, and $\mathsf{Within}(c, b)$ hold, but also (II) $\mathsf{Business}(c)$, $\mathsf{HasLicense}(c, d)$, $\mathsf{Restaurant}(d)$, $\mathsf{Within}(c, b)$. (II) is equivalent to saying that `Those buildings producing a lot of trash may contain restaurants`, an interesting — and more specific — hypothesis compared to (I). We can test this newly generated hypothesis and see whether the data statistically supports it. Those cases for which this rule does not hold may point to fraudulent (or absent) license registers, when apartments host businesses (that may produce a lot of trash) rather than being residential.

3 Background

3.1 Ontologies for Smart Cities

The complexity and similarity of city structures and mechanisms across the world makes them an ideal scenario for using semantic technologies to capture fundamental cross-domain models. Ontologies defined this way can offer a global view of the many heterogeneous sources without modifying them, can unify meaning and relationships across domains and applications, and can be more readily re-used, shared, and evolved. Cities such as A Croatia, Washington DC, Vienna, or neighborhoods of Winnipeg have built and used city ontologies to offer integrated information. The city of Barcelona is working on their own City Operating System, a platform that will allow the interconnected management of the different city services based on the integration of heterogeneous data sources via an ontology. City ontologies require reasoning tasks that are powerful, yet should be nonrestrictive enough to still be able to integrate and make good use of noisy data.

3.2 A bird's eye on abduction

Abductive reasoning has a rich history in Philosophy. Abductive reasoning (also called abduction[11], abductive inference, [11] is a form of logical inference which starts with an observation then seeks to find the simplest and most likely explanation. Abduction was studied in computer science logic [17] both for propositional logic e.g.,[16] and first order logic e.g., [15]. One of the most useful application of abduction in computer science is for logic programming [12]. Abduction also plays a crucial role in program verification, e.g., [2, 5]. Indeed our definition of abduction is inspired by [5].

Several kinds of abduction tasks are possible in Description Logics — the most commonly used formalism for ontologies and the Semantic Web: concept abduction, ABox abduction, TBox abduction, or Knowledge-base abduction — a generalization over TBox and ABox abductions. The ABox consists of assertions. The TBox contains statements about terminology. ABox abduction focuses on finding instances of entities or relations that would entail an ABox assertion. TBox generally focuses on repairing undesired non-subsumptions. In the context of ontologies, [18] restricts the notion of SAT in FOL to be able to use tableaux for finding solutions to abduction problems. [6] focuses on the query abduction problem to compute the minimal explanations that may be added to entail observations in the ABox. [1] emphasizes abduction as a feature of support tools for ontology quality control. [3] applies abduction to explain why a tuple is not an answer to a conjunctive query. All of these approaches, just like [7] and [8], focus on reasoning over the ABox. Some successful tools on abductive reasoning in logic programming are CIFF[14] and A — system[13]. These tools are built on modern Prolog engines, but they only allow the background theory to be a normal logic program, which corresponds to a plain datalog program extended with negation-as-failure. Our focus is on TBox abduction, which is less popular. An automata-based method for TBox abduction has been proposed in [10].

Our work utilizes SMT based tools for computing abducers and applying abduction for smart cities.

4 An Approach Based on Computing Abducers using Z3

We can formalize the problem of finding explanations as an abduction inference problem. That is, finding abducers Ab of implications $\mathit{Hyp} \implies \mathit{Obs}$, such that $Ab \wedge \mathit{Hyp}$ is consistent and implies Obs. In our example $\mathit{Hyp} := A_1, A_2, A_3, \mathsf{Business}(b)$ and $\mathit{Obs} := \mathsf{TrashHog}(b)$. Thus, we assume that b is a business and we would like to explain which buisinesses can be trashogs. We here provide a recipe using Z3 that allows enumerating alternative candidate abducers.

1. We first create a set of formulas, $\mathcal{CC}$, based on the Clark completion [4] our axioms. Besides A_1, A_2, A_3, we would also include assumptions $\mathsf{Building}(b), \mathsf{TrashHog}(b)$ and axioms $\mathsf{Apartment} \subseteq \mathsf{Business}$ (the contrapositive of A_1 required for the Clark completion) and $\mathsf{TrashHog}(X) \Rightarrow (\mathsf{Apartment}(a2(X)) \wedge \mathsf{TrashHog}(a2(X)) \wedge \mathsf{Within}(a2(X), X) \wedge \mathsf{Building}(X)) \vee (\mathsf{Business}(X) \wedge \mathsf{HasLicense}(X, a3(X)) \wedge \mathsf{Restaurant}(a3(X)))$, where $a2, a3$ are Skolem functions corresponding to the existential quantification over Y.

2. We invoke a model finder (Z3) to obtain an interpretation $\mathcal{M}$ for $\mathcal{CC}$ and $\mathcal{F} \subset \mathcal{M}$ a finite set of literals obtained from the Robinson diagram [9] for $\mathcal{M}$. The Clark completion ensures that $\mathcal{M}$ contains a trace that justifies the observable.

3. We obtain abducers by enumerating cores for the formula $\bigwedge(\mathcal{F} \setminus \{\mathsf{TrashHog}(b)\}) \wedge A_1 \wedge$

$A_2 \wedge A_3 \wedge \mathsf{Business}(b) \wedge \neg \mathsf{TrashHog}(b)$. The MARCO [1] procedure comes in handy for core enumeration. The extracted unsatisfiable cores over $\mathcal{F}$ can be used as abducers.

4. Step 2 is repeated after we add the negated cores. This exposes new alternative models that do not intersect with the previous cores.

5 Applying and Refining Abducers

The extracted abducers are validated against collected data. Indeed, abducer (II) will prove to be valid in several cases, but one can expect that restaurants are not the only sources of trash or that buildings contain unlicensed restaurants. An analyst may further check whether there are apartments in that building of type business. If there are, they could check whether it is these businesses which produce a lot of trash. If that is in fact the case, it may be the case that other types of businesses besides restaurants produce a lot of trash, or that the license type is erroneously registered. Our mini-ontology suggests that businesses of type Bar may be good candidates for trash generators. Conversely, we hypothesize, the extracted rules can be used to label data.

6 Acknowledgments

The work of Maria-Cristina Marinescu has been partially supported by the Spanish "Ministerio de Economia y Competitividad" under project grant SEMIOTIC (ref. TIN2016-78473-C3-2-R).

References

[1] Michael Bada, Chris Mungall, and Lawrence Hunter. A call for an abductive reasoning feature in owl-reasoning tools toward ontology quality control. In *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26-27, 2008*, 2008.

[2] Cristiano Calcagno, Dino Distefano, Peter W. O'Hearn, and Hongseok Yang. Compositional shape analysis by means of bi-abduction. *J. ACM*, 58(6):26:1–26:66, 2011.

[3] Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni. Reasoning about explanations for negative query answers in dl-lite. *J. Artif. Intell. Res.*, 48:635–669, 2013.

[4] Keith L. Clark. Negation as failure. In Hervé Gallaire and Jack Minker, editors, *Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, 1977.*, Advances in Data Base Theory, pages 293–322, New York, 1977. Plemum Press.

[5] Isil Dillig, Thomas Dillig, Boyang Li, Kenneth L. McMillan, and Mooly Sagiv. Synthesis of circular compositional program proofs via abduction. *STTT*, 19(5):535–547, 2017.

[6] Jianfeng Du, Kewen Wang, and Yi-Dong 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.*, pages 1034–1040, 2014.

[7] Jianfeng Du, Kewen Wang, and Yi-Dong Shen. Towards tractable and practical abox abduction over inconsistent description logic ontologies. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1489–1495, 2015.

[8] Jianfeng Du, Shuai Wang, Guilin Qi, Jeff Z. Pan, and Yong Hu. A new matchmaking approach based on abductive conjunctive query answering. In *The Semantic Web - Joint International*

[1] sun.iwu.edu/~mliffito/marco/

Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings, pages 144–159, 2011.

[9] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.

[10] Thomas Hubauer, Steffen Lamparter, and Michael Pirker. Automata-based abduction for tractable diagnosis. In *Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010*, 2010.

[11] John R. Josephson and Susan G. Josephson. *Abductive Inference: Computation, Philosophy, Technology*. Cambridge University Press, Cambridge, UK; New York, 1st edition, 1994.

[12] Antonis C. Kakas, Robert A. Kowalski, and Francesca Toni. Abductive logic programming. *J. Log. Comput.*, 2(6):719–770, 1992.

[13] Antonis C. Kakas, Bert Van Nuffelen, and Marc Denecker. A-system: Problem solving through abduction. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, pages 591–596, 2001.

[14] Paolo Mancarella, Giacomo Terreni, and Francesca Toni. Web sites repairing through abduction. *Electr. Notes Theor. Comput. Sci.*, 235:137–152, 2009.

[15] Marta Cialdea Mayer and Fiora Pirri. First order abduction via tableau and sequent calculi. *Logic Journal of the IGPL*, 1(1):99–117, 1993.

[16] Marta Cialdea Mayer and Fiora Pirri. Propositional abduction in modal logic. *Logic Journal of the IGPL*, 3(6):907–919, 1995.

[17] Marta Cialdea Mayer and Fiora Pirri. A study on the logic of abduction. In *12th European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings*, pages 18–22, 1996.

[18] A. Liliana Reyes-Cabello, Atocha Aliseda-Llera, and Ángel Nepomuceno-Fernández. Towards abductive reasoning in first-order logic. *Logic Journal of the IGPL*, 14(2):287–304, 2006.