Expressive Identification Constraints to Capture Functional Dependencies in Description Logics*

Diego Calvanese[1], Wolfgang Fischl[2], Reinhard Pichler[2], Emanuel Sallinger[2], and Mantas Šimkus[2]

[1] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
[2] Institute of Information Systems, Vienna University of Technology, Austria

Motivation and Main Results. Over the past years, we have been witnessing an enormous growth of the Semantic Web through initiatives like Open Linked Data [5] and Open Government Data [14, 25]. As was noted by He et al. [13] and Madhavan et al. [17], to a large extent, the data accessible on the web still originates from relational databases. The design of these databases often follows specific principles, called normal forms, developed in the beginnings of relational database research, see, e.g., [11, 12].

The goal of this work is to transfer these principles, in particular the Boyce-Codd Normal Form (BCNF), to RDF graphs enhanced with RDFS statements. To establish and justify this normal form for RDF graphs, we need the following: (1) A mapping of relational databases to RDF graphs and (2) identification constraints that capture functional dependencies (FDs) over RDF graphs.

For (1), W3C has recognized the importance of a standardized mapping of relational data to the Semantic Web data format RDF. To this end, the so-called *direct mapping* has been released as a W3C Recommendation [2]. Note that the direct mapping to RDF does not transfer the semantic information that may be present in the relational schema, e.g. functional or inclusion dependencies. We are going to study an enrichment of the direct mapping by transferring also important semantic information from relational to RDF data. Initial work on this includes the recent proposal to extend the direct mapping by the use of RDFS and OWL 2 vocabularies [21], achieving the transfer of primary and foreign keys. The mapping in [21] enjoys several important properties such as query-preservation. However, if the RDF graph resulting from such a mapping is later changed (through update, delete, or insert operations), then the correspondence between the relational and the RDF data may get lost. We therefore propose a further extension of the direct mapping that uses *DL-Lite*$_{RDFS}$ [3] — extended with disjointness — as basis. *DL-Lite*$_{RDFS}$ is a variant of *DL-Lite*$_{\mathcal{A}}$ [7] and captures the Description Logic (DL) fragment of RDFS [6]. While this DL is simple and allows for efficient reasoning, it naturally captures conceptual modeling constructs, and hence can express dependencies over RDF graphs. We introduce a mapping *d2r* that produces from a database instance an RDF graph together with a mapping

* This is an extended abstract of [10]. The first author has been partially supported by the Wolfgang Pauli Institute Vienna, and by the EU IP project Optique(grant agreement n. FP7-318338). The remaining authors have been partially supported by the Austrian Science Fund (FWF) project P25207-N23 and P25518-N23, and by the Vienna Science and Technology Fund (WWTF) project ICT12-15.

sm that outputs from a relational schema a DL TBox constraining RDF graphs. For this we use the well-known reification technique. We keep good properties of the mapping proposed in [21] – such as query preservation. In addition, we also introduce a mapping *r2d* that produces from an RDF graph (conforming to a DL TBox generated by *sm*) a database. This allows us to prove a desired one-to-one correspondence between relational databases and *legal* RDF graphs (i.e., RDF graphs satisfying the constraints of the TBox).

For (2), since *functional dependencies* (FDs) are a crucial building block in database design [18] and form the basis of BCNF, the focus of our work is on FDs. Intuitively, for a relation R, an FD $\{A_1, \ldots, A_n\} \rightarrow_R A_0$ expresses that if two tuples of R agree on the values of all attributes $A_1, \ldots, A_n$, they also have to agree on the value of attribute A_0. We will see how this notion can be extended to the DL and RDF setting through the use of paths. Sequeda, Arenas, and Miranker [21] have extended the direct mapping by constraints such as primary and foreign keys, while FDs have not been in the scope of their work. Several works consider DLs extended with FDs. Calvanese, De Giacomo, and Lenzerini [9] enrich DLs with a generalization of DL functionality assertions, called identification constraints (ids). The latter are extended by Calvanese et al. [8] to path-based ids (pids). Lutz, et al. [16] introduce *key assertions* as a possibility to use paths for identifying concepts. A different approach was taken by Khizder, Toman and Weddell [15, 22]. They have established a DL, called $\mathcal{CFD}$, which captures usual relational schema declarations. This DL includes uniqueness constructs, which capture FDs. Furthermore, they have extended uniqueness constructs to path-functional dependencies (PFDs) and have investigated their properties in more expressive DLs, such as $\mathcal{ALCN}$ [23]. In their most recent work [24] they have established PTime reasoning for the DL $\mathcal{CFD}$ extended with PFDs and disjointness constraints. However, altough the results of Toman and Wedell are a viable approach to reason about relational schemas, we are interested in capturing relational schemas and FDs in RDFS and OWL, thus following more closely the W3C standards. As *DL-Lite* is the logical underpinning of OWL 2 QL [20], we will focus on extensions for modelling FDs in *DL-Lite* , which is given by the earlier mentioned pids. We investigate their expressivness and show, that they fail to capture FDs for the direct mapping of relational data to RDF. We therefore introduce an extension of such ids, which we call *tree-based ids* (tids). With this new class of ids, we shall restore the desired one-to-one relationship between *legal* databases (i.e., satisfying a given set of FDs) and *legal* RDF graphs.

As mentioned above, our goal is to find BCNF-like conditions for RDF graphs. In relational schemas the purpose of using BCNF is to avoid update anomalies. A relational schema is in BCNF if the following holds: whenever a set Σ of given FDs implies an FD from a subset S of the attributes to some attribute $A \notin S$, we have that S is a super-key, i.e., Σ also implies an FD from S to *every* attribute of this schema. Our goal is to transfer the favorable properties of BCNF to the RDF world. To this end, we first analyze how update anomalies can arise in the presence of tids. We identify several paths (stemming from the same tid) identifying the same object as a crucial source of redundancy and hence of update

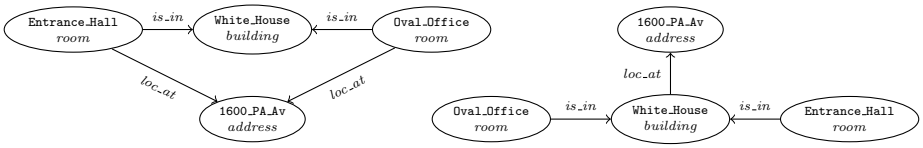

Fig. 1. An RDF graph with data about buildings.

anomalies. This observation inspires the definition of an *RDF Normal Form* (RNF). Returning to the direct mapping, we prove that a relational schema is in BCNF if and only if the corresponding TBox with its constraints guarantees RNF. As a kind of surprise, it turns out that — for relational schemas in BCNF — the additional expressive power of tree-based ids is not needed to capture FDs. Indeed, under the restriction to BCNF, the original form of ids introduced by Calvanese, De Giacomo, and Lenzerini [9] is expressive enough to transfer FDs from the relational schema to the DL TBox. Finally, we propose an algorithm, which decides in polynomial time whether a given set tids is in RNF. We will now give a short example, that illustrates of tids and RNF.

Example 1. Consider the two RDF graphs in Figure 1. Both RDF graphs store data about buildings. Each building has several rooms with an address. Clearly, all rooms in the same building have the same address. Such a restriction can be expressed using a tid. In the left RDF graph of Figure 1, the information that the `"White_House"` is located in `"16000_PA_Av"` is stored redundantly, due to the design, where rooms in a building have addresses rather than the building has the address itself. We can avoid such a redundancy by storing the address connected directly to the building concept, as it is in the right RDF graph of Figure 1. Our definition of RNF detects such problems in the design of DL TBoxes with tids.

Future Work On top of our agenda for future research is the extension of our work on RNF. So far, we have concentrated on preserving BCNF of a relational schema under the direct mapping of relational data to RDF. However, normal forms for eliminating redundancies in the data would be an interesting topic for the design of TBoxes in general. We thus see three main directions to continue our work. First, we would like to extend the definition of our RNF to other, maybe more expressive, DLs than *DL-Lite*$_{RDFS,tid}$, e.g. also to the DL $\mathcal{CFD}$ introduced by Toman and Wedell. Note that this raises highly non-trivial questions concerning the recognizability of the normal form, since our PTIME-membership result for this task crucially depends on the language restrictions of *DL-Lite*$_{RDFS,tid}$. Second, we also want to investigate relaxations of our definition of RNF. In our current definition, we request that a set of tids must be equivalent to a set of fully local ids. This allows us to capture BCNF in *DL-Lite*$_{RDFS,tid}$. However, for the definition of a normal form of more expressive DLs, the equivalence of tids to a richer class of ids – such as local pids considered by Calvanese et al. [8] may be more appropriate. And at last, we would like to investigate the relationship of RNF to other normal forms of non-relational data sources, e.g. XML [1], nested relations [19] or object-oriented data models, like F-Logic [4].

References

1. Arenas, M., Libkin, L.: A normal form for XML documents. ACM Trans. on Database Systems 29(1), 195–232 (2004)
2. Arenas, M., Bertails, A., Prud′hommeaux, E., Sequeda, J.: A direct mapping of relational data to RDF. W3C Recommendation, W3C (Sep 2012), available at `http://www.w3.org/TR/rdb-direct-mapping/`
3. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V., Sherkhonov, E.: Exchanging description logic knowledge bases. In: Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012). pp. 308–318. AAAI Press (2012)
4. Biskup, J., Menzel, R., Polle, T., Sagiv, Y.: Decomposition of relationships through pivoting. In: Conceptual Modeling ER'96, pp. 28–41. Springer (1996)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. on Semantic Web Information Systems 5(3), 1–22 (2009)
6. Brickley, D., Guha, R.V.: RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium (Feb 2004), available at `http://www.w3.org/TR/rdf-schema/`
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R.: Linking data to ontologies: The description logic DL-Lite$_A$. In: Proc. of the 2nd Int. Workshop on OWL: Experiences and Directions (OWLED 2006). CEUR Workshop Proceedings, vol. 216. CEUR-WS.org (2006)
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Path-based identification constraints in description logics. In: Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008). pp. 231–241. AAAI Press (2008)
9. Calvanese, D., De Giacomo, G., Lenzerini, M.: Identification constraints and functional dependencies in description logics. In: Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001). pp. 155–160. Morgan Kaufmann (2001)
10. Calvanese, D., Fischl, W., Pichler, R., Sallinger, E., Šimkus, M.: Capturing relational schemas and functional dependencies in RDFS. In: Proc. of the 28th AAAI Conf. on Artificial Intelligence (AAAI 2014). AAAI Press (2014), to appear
11. Codd, E.F.: Further normalization of the data base relational model. IBM Research Report RJ909, IBM, San Jose, California (1971)
12. Codd, E.F.: Normalized data structure: A brief tutorial. In: Proc. of the SIGFIDET Workshop. pp. 1–17. ACM (1971)
13. He, B., Patel, M., Zhang, Z., Chang, K.C.C.: Accessing the deep web. Communications of the ACM 50(5), 94–101 (2007)
14. HM Government: data.gov.uk. `http://data.gov.uk` (2014)
15. Khizder, V.L., Toman, D., Weddell, G.: Reasoning about duplicate elimination with description logic. In: Computational LogicCL 2000, pp. 1017–1032. Springer (2000)
16. Lutz, C., Areces, C., Horrocks, I., Sattler, U., et al.: Keys, nominals, and concrete domains. J. Artif. Intell. Res.(JAIR) 23, 667–726 (2005)
17. Madhavan, J., Afanasiev, L., Antova, L., Halevy, A.Y.: Harnessing the deep web: Present and future. In: Proc. of the 4th Biennial Conf. on Innovative Data Systems Research (CIDR 2009) (2009)
18. Mannila, H., Räihä, K.J.: The Design of Relational Databases. Addison Wesley Publ. Co. (1992)
19. Mok, W.Y., Ng, Y.K., Embley, D.W.: A normal form for precisely characterizing redundancy in nested relations. ACM Trans. Database Syst. 21(1), 77–106 (Mar 1996), `http://doi.acm.org/10.1145/227604.227612`

20. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 web ontology language: Profiles. http://www.w3.org/TR/owl2-profiles/ (2012)
21. Sequeda, J., Arenas, M., Miranker, D.P.: On directly mapping relational databases to RDF and OWL. In: Proc. of the 21st Int. World Wide Web Conf. (WWW 2012). pp. 649–658. ACM (2012)
22. Toman, D., Weddell, G.E.: On the interaction between inverse features and path-functional dependencies in description logics. pp. 603–608 (2005)
23. Toman, D., Weddell, G.E.: On keys and functional dependencies as first-class citizens in description logics. J. of Automated Reasoning 40(2–3), 117–132 (2008)
24. Toman, D., Weddell, G.E.: CFDnc: A PTIME description logic with functional constraints and disjointness. In: DL-13. pp. 451–463 (2013)
25. US Government: data.gov. http://www.data.gov (2014)