Modular Provenance in Multi-Context Systems (Extended Abstract)*

Matthias Knorr, Carlos Viegas Damásio, Ricardo Gonçalves, and João Leite

NOVA LINCS, Departamento de Informática, FCT NOVA, Portugal
`{mkn, cd, rjrg, jleite}@fct.unl.pt`

Abstract. A rapidly increasing amount of data, information and knowledge is becoming available on the Web, often written in different formats and languages, adhering to standardizations driven by the World Wide Web Consortium initiative. Taking advantage of all this heterogeneous knowledge requires its integration for more sophisticated reasoning services and applications. To fully leverage the potential of such systems, their inferences should be accompanied by justifications that allow a user to understand a proposed decision/recommendation, in particular for critical systems (healthcare, law, finances, etc.). However, determining such justifications has commonly only been considered for a single formalism, such as relational databases, description logic ontologies, or declarative rule languages. In this paper, we give an overview on the first approach for providing provenance for heterogeneous knowledge bases building on the general framework of multi-context systems, as an abstract, but very expressive formalism to represent knowledge bases written in different formalisms and the flow of information between them.

Keywords: Provenance; Heterogeneous knowledge bases; Multi-context systems

1 Introduction

A rapidly increasing amount of data, information and knowledge is becoming available on the Web, driven by the Semantic Web initiative led by the World Wide Web Consortium (W3C).[1] A number of language standards have been established in this initiative and to take advantage of all this available knowledge often requires their integration. This is particularly true for (but not limited to) integrations of rule languages, e.g., under answer set semantics [4] and ontology languages based on description logics [1], that are both highly expressive, but with orthogonal/complementary characteristics and modelling features (see, e.g., [11, 22, 18, 12, 21] and references therein).

However, in the course of the integration of such heterogeneous knowledge, it becomes increasingly difficult to trace the causes for a certain inference, or find

---

[*] This paper is an extended abstract of a conference publication [19].

[1] https://www.w3.org/

the justification for some proposed decision, in particular, if the pieces of knowledge originate from different authors. It would thus be important to provide methods that accompany inferences/decisions with explanations/justifications in a way a user can understand to allow for the validation of reasoning results, in particular for critical systems (healthcare, law, finances, etc.).

This has been recognized in different areas of Artificial Intelligence, and for several Knowledge Representation and Reasoning formalisms, the problem of finding justifications has been considered. In particular, a lot of work has focussed on tracing the origins of derivations, commonly under the name of provenance [5], e.g., in relational databases and Datalog [15, 16], Logic Programming [8], Answer Set Programming [13], Description Logics ontology languages [17, 6, 2], as well as in SPARQL [7] and data streams [14]. Yet, provenance for heterogeneous knowledge bases has mostly been ignored, with the exception of [9], though limited to two very restricted settings.

In this paper, we overview research results on justifications in terms of provenance for heterogeneous knowledge bases, utilising multi-context systems (MCSs) [3] as our formalism of choice. MCSs allow for the integration of a large variety of logic-based formalisms, and model the flow of information between them. They cover very general approaches for integrating ontologies and rules [20], thus allowing to study provenance in a more general manner, which then paves the way towards provenance in related approaches in the literature. We focus on providing justifications of inferences (the only question that has been handled in the literature are explanations of inconsistencies when repairing inconsistent multi-context systems [10], which is inherently different).

The contributions of our research can be summarized as follows:

- We develop the first general approach for provenance in heterogeneous knowledge bases, and in multi-context systems in particular, annotating inferences with their justifying provenance information.
- We provide means to compute this provenance information annotating models, so-called equilibria, in MCSs.
- We establish under which conditions this provenance information can indeed be computed, showing its applicability to a wide class of formalisms.

Here, we overview the main ideas of the approach, leaving the details to the full paper [19].

2 Provenance Multi-Context Systems

Multi-context systems (MCSs) [3] are defined as a collection of components, so-called contexts, each of which allows one to represent knowledge in some logic-based formalism. Each such logic is associated with a set of well-formed knowledge bases in the logic (its admitted syntax), possible belief sets, indicating how models are defined in this logic (its admitted semantics), and a function assigning to each possible knowledge base a set of acceptable such belief sets.

MCSs use so-called bridge rules that allow one to model the flow of information between these contexts, in the sense that they admit the incorporation of knowledge in one context based on the beliefs considered true in other contexts. The semantics of MCSs is then assigned using equilibria that take the acceptable belief sets and the interaction between contexts into account.

Provenance multi-context systems extend MCSs with the means to explain inferences obtained from the modular integration of its contexts. In the course of this presentation, we assume that the reader is familiar with notions of provenance semirings.

First, to be able to represent provenance, corresponding annotations need to be defined. In our approach, the essential idea is to provide annotation languages for each context, together with means to construct annotations that take the interaction between contexts into account. Based on a number of annotation names N_i and binary operators Σ_i, this results in a number of annotation languages V_i and one particular language V_*, contained in all the others, which is meant to correspond to the integration of information in bridge rules between contexts. For the technical details, we refer to the extended paper.

Based on this, we can introduce provenance logics as a means to capture a large variety of formalisms that allow tracing the reasons for inferences.

Definition 1. *A* provenance logic L *is a tuple* $(\mathcal{K}, \mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ *where*

(1) $\mathcal{K}$ *is a commutative semiring over polynomials over some* V_i *with* $\oplus_\mathcal{K}, \otimes_\mathcal{K} \in \Sigma_i$*, and a natural order* $\preceq_\mathcal{K}$*;*

(2) $\mathbf{KB}$ *is the set of well-formed knowledge bases of* L *such that each* $kb \in \mathbf{KB}$ *is a set composed of formulas distinctly annotated with elements from* V_i*;*

(3) $\mathbf{BS}$ *is the set of possible annotated belief sets, i.e., functions that map beliefs from the set of possible beliefs* B_L *of* L *to* V_i*, such that false beliefs are mapped to* $0_\mathcal{K}$*;*

(4) $\mathbf{ACC} : \mathbf{KB} \to 2^{\mathbf{BS}}$ *is a function describing the semantics of* L *by assigning to each knowledge base a set of acceptable annotated belief sets.*

Note that the idea of possible belief sets from MCSs is extended in that sets of annotated beliefs are used. I.e., rather than using sets of beliefs which are meant to be true, sets of beliefs with their corresponding annotations are considered. The function **ACC** then assigns semantics to knowledge bases by associating them with acceptable annotated belief sets. Here, in the spirit of MCSs, we focus on determining the provenance of true elements.

Example 1. We present some example provenance logics.

- L_{db} – Databases with provenance under bag semantics [15]:
  - $\mathcal{K}_{db}$: $\mathbb{N}[X]$;
  - $\mathbf{KB}_{db}$: the set of annotated databases together with queries expressed in an appropriate query language, such as Datalog;
  - $\mathbf{BS}_{db}$: the set of sets of atoms with annotations;
  - $\mathbf{ACC}_{db}(kb)$: the set of tuples in kb and query results over kb with their annotation according to $\mathcal{K}_{db}$;
- L_{dl} – Description Logic $\mathcal{ELH}^r$ [2]:

  - $\mathcal{K}_{dl}$: $Trio[X]$, i.e., $\mathbb{N}[X]$ with idempotent $\times$;
  - $\mathbf{KB}_{dl}$: set of well-formed annotated $\mathcal{ELH}^r$ ontologies;
  - $\mathbf{BS}_{dl}$: the set of sets of annotated atomic inferences;
  - $\mathbf{ACC}_{dl}(kb)$: the set of atomic inferences from kb with their annotation according to $\mathcal{K}_{dl}$;
- L_{lp} – Normal logic programs under answer set semantics (adapted from [8]):
  - $\mathcal{K}_{lp}$: $PosBool[X]$, i.e., $\mathbb{N}[X]$ with idempotent $+$ and $\times$ and absorption on $+$, over positive atoms;
  - $\mathbf{KB}_{lp}$: the set of annotated normal logic programs;
  - $\mathbf{BS}_{lp}$: the set of sets of atoms with annotations;
  - $\mathbf{ACC}_{lp}(kb)$: the answer sets of kb with annotations according to $\mathcal{K}_{lp}$;

Similar to MCSs, bridge rules are used to specify how knowledge is transferred between the different components.

Definition 2. *Given a collection of provenance logics $L = \langle L_1, \ldots, L_n \rangle$, an L_i-*bridge rule *over L, $1 \leq i \leq n$, is of the form:*

$$\begin{aligned} \pi@s \leftarrow (r_1 : p_1), \ldots, (r_j : p_j), \\ \mathbf{not}\,(r_{j+1} : p_{j+1}), \ldots, \mathbf{not}\,(r_m : p_m) \end{aligned} \tag{1}$$

where $\pi \in N_*$ and, for $1 \leq k \leq m$, $1 \leq r_k \leq n$ and $p_k \in B_{L_{r_k}}$, and, for each $kb \in \mathbf{KB}_i$, $kb \cup \{v@s\} \in \mathbf{KB}_i$ for every $v \in V_*$.

Note that each of the r_k refer to one of the logics and the beliefs p_k belong to the corresponding set of possible beliefs $B_{L_{r_k}}$ of logic L_{r_k}. Also note that π is the annotation name of the bridge rule itself, whereas v is an annotation variable associated to the bridge rule head s, intended to be incorporated into the knowledge base kb.

With this in place, we can introduce provenance multi-context systems.

Definition 3. *A* provenance multi-context system (pMCS) *is a collection of contexts $M = \langle C_1, \ldots, C_n \rangle$ where $C_i = (L_i, kb_i, br_i)$, $L_i = (\mathcal{K}_i, \mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a provenance logic, $kb_i \in \mathbf{KB}_i$ a knowledge base, and br_i is a set of L_i-bridge rules over $\langle L_1, \ldots, L_n \rangle$.*

We assume that the annotations used for the elements occurring in the individual kb_i are unique elements from N_i, and that each context uses a different set of annotations V_i. Also, while different forms of specifying the annotations of formulas can be found in the literature, here we use uniformly the notation introduced for bridge rules, i.e., the annotation is given in front of a formula with @ as separator.

Example 2. Consider $M = \langle C_1, C_2, C_3 \rangle$ such that:
- C_1 is a database context with L_{db}, $kb = \{d_1@p(1,1), d_2@p(1,2)\}$ with a single relation $p(e,f)$ with two tuples, $br_1 = \emptyset$, and query q defined by $q(x,y) \leftarrow p(e,x), p(e,y)$;

- C_2 a DL context with L_{dl}, $kb_2 = \{o_1@A \sqsubseteq B\}$, and $br_2 = \{b_1@A(w) \leftarrow \mathbf{not}\,(3 : l)\}$;
- C_3 an ASP context with L_{lp}, $kb_3 = \{r_1@l \leftarrow \mathit{not}\ m, n\}$, and $br_3 = \{c_1@n \leftarrow (1 : q(1, 1)), c_2@m \leftarrow (2 : B(w))\}$.

As C_1 has no bridge rules, we obtain $\mathbf{ACC}_{db}(kb_1) = \{S_1\}$ with $S_1(p(1, 1)) = d_1$, $S_1(p(1, 2)) = d_2$, $S_1(q(1, 1)) = d_1^2$, $S_1(q(1, 2)) = S_1(q(2, 1)) = d_1 \times d_2$, and $S_1(q(2, 2)) = d_2^2$. For both kb_2 and kb_3, $\mathbf{ACC}_i(kb_i) = \{S_i\}$ with S_i mapping every atomic inference/atom to 0 (as the bridge rules are not considered for the semantics of individual contexts).

Regarding the semantics of pMCSs, belief states are used, i.e., collections $S = \langle S_1, \dots, S_n\rangle$ such that each S_i is an element of $\mathbf{BS}_i$. Among them, specific belief states, called equilibria, exist that take bridge rules into account for determining acceptable belief states, similar to MCSs. We adapt this with annotations building on the algebraic approach for non-monotonic rules [8] to pass annotation information via bridge rules. The main idea is to use annotations from V_* assuming the existence of distinct negative names (using **not**) in the respective N_i, one per negated p_k with $j + 1 \leq k \leq m$ for bridge rules of the form (1). This is necessary as we assume that false beliefs are annotated with $0_{\mathcal{K}}$, thus no annotations exist for such negations.

Definition 4. *The* commutative semiring for bridge rules $\mathcal{BR}$ *is defined as PosBool[V_*], for $\wedge, \vee \in \Sigma_*$, with idempotent meet ($\wedge$) and join ($\vee$), absorption on $\vee$, and logical consequence as natural order, i.e., $k_1 \preceq_{\mathcal{BR}} k_2$ iff $k_1 \models k_2$.*

Based on this, we can formally define when bridge rules are applicable (when their positive body elements are true and their negative body elements are false), and how this can be used to determine equilibria, associating the corresponding annotations to true beliefs. We refer for the details to the technical paper.

Example 3. Consider M from Ex. 2. Since C_1 does not contain bridge rules, S_1 is fully determined in Ex. 2. Then, by the first rule in br_3, we have that $S_3(n) = c_1 \wedge d_1^2$. If the other rule in br_3 is not applicable, then $S_3(l) = r_1 \times_3 (c_1 \wedge d_1^2)$ holds. In this case, the only bridge rule in br_2 is not applicable, thus $B(w)$ cannot be inferred from C_2 which ensures that the second rule in br_3 is not applicable. In fact, together with S_2 mapping every atomic inference to 0, we obtain an equilibrium.

In the extended paper, we then also show under which conditions and how equilibria can be computed.

Acknowledgments We thank the anonymous reviewers for their helpful comments and we acknowledge partial support of this work by FCT projects RIVER (PTDC/CCI-COM/30952/2017), FORGET (PTDC/CCI-INF/32219/2017), and NOVA LINCS (UIDB/04516/2020).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
2. Bourgaux, C., Ozaki, A., Peñaloza, R., Predoiu, L.: Provenance for the description logic $\mathcal{ELH}^r$. In: IJCAI. pp. 1862–1869. ijcai.org (2020)
3. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Procs. of AAAI. pp. 385–390. AAAI Press (2007)
4. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Commun. ACM **54**(12), 92–103 (2011)
5. Buneman, P.: The providence of provenance. In: BNCOD. LNCS, vol. 7968, pp. 7–12. Springer (2013)
6. Calvanese, D., Lanti, D., Ozaki, A., Peñaloza, R., Xiao, G.: Enriching ontology-based data access with provenance. In: Procs. of IJCAI. pp. 1616–1623. ijcai.org (2019)
7. Damásio, C.V., Analyti, A., Antoniou, G.: Provenance for SPARQL queries. In: Procs. of ISWC. LNCS, vol. 7649, pp. 625–640. Springer (2012)
8. Damásio, C.V., Analyti, A., Antoniou, G.: Justifications for logic programming. In: Procs. of LPNMR. LNCS, vol. 8148, pp. 530–542. Springer (2013)
9. Dividino, R.Q., Schenk, S., Sizov, S., Staab, S.: Provenance, trust, explanations - and all that other meta knowledge. KI **23**(2), 24–30 (2009)
10. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. Artif. Intell. **216**, 233–274 (2014)
11. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artif. Intell. **172**(12-13), 1495–1539 (2008)
12. Eiter, T., Simkus, M.: Linking open-world knowledge bases using nonmonotonic rules. In: Procs. of LPNMR. LNCS, vol. 9345, pp. 294–308. Springer (2015)
13. Fandinno, J., Schulz, C.: Answering the "why" in answer set programming - A survey of explanation approaches. Theory Pract. Log. Program. **19**(2), 114–203 (2019)
14. Glavic, B., Esmaili, K.S., Fischer, P.M., Tatbul, N.: Ariadne: managing fine-grained provenance on data streams. In: Procs. of DEBS. pp. 39–50. ACM (2013)
15. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Procs. of PODS. pp. 31–40. ACM (2007)
16. Green, T.J., Tannen, V.: The semiring framework for database provenance. In: Procs. of PODS. pp. 93–99. ACM (2017)
17. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: Procs. of ISWC. LNCS, vol. 5318, pp. 323–338. Springer (2008)
18. Knorr, M., Alferes, J.J., Hitzler:, P.: Local closed world reasoning with description logics under the well-founded semantics. Artificial Intelligence **175**(9-10), 1528–1554 (2011)
19. Knorr, M., Damásio, C.V., Gonçalves, R., Leite, J.: Towards provenance in heterogeneous knowledge bases. In: Procs. of LPNMR. Springer (2022), to appear
20. Knorr, M., Slota, M., Leite, J., Homola, M.: What if no hybrid reasoner is available? hybrid MKNF in multi-context systems. J. Log. Comput. **24**(6), 1279–1311 (2014)
21. Lukumbuzya, S., Ortiz, M., Simkus, M.: Resilient logic programs: Answer set programs challenged by ontologies. In: Procs. of AAAI. pp. 2917–2924. AAAI Press (2020)

22. Motik, B., Rosati, R.: Reconciling description logics and rules. Journal of the ACM **57**(5), 93–154 (2010)