LACE: A Logical Approach to Collective Entity Resolution (Extended Abstract)

Meghyn Bienvenu[1] , **Gianluca Cima**[1] , **Víctor Gutiérrez-Basulto**[2]
[1]CNRS & University of Bordeaux
[2]Cardiff University
meghyn.bienvenu@cnrs.fr, gianluca.cima@u-bordeaux.fr, gutierrezbasultov@cardiff.ac.uk

The original paper will appear in the proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2022).

The camera-ready paper is publicly accessible at the following URL:
https://hal.archives-ouvertes.fr/hal-03686662/document

Keywords: Collective Entity Resolution, Declarative Framework, Logical Constraints, Complexity Analysis, Answer Set Programming

In this work, we investigate the *entity resolution (ER)* problem (Newcombe et al. 1959), which is one of the most fundamental problems in data quality management. Given a database D, the task is to determine, for each pair (c_1, c_2) of constants (of the same type) occurring in D, whether they represent the same real-world entity, and can thus be merged (Singla and Domingos 2006). In particular, we propose LACE, a logical framework for ER that was designed to satisfy three main desiderata, namely, being *collective*, *declarative*, and *justifiable*. More precisely, our approach (i) supports complex interdependencies between merges of different entities, (ii) adopts a declarative language with logical rules and constraints, and (iii) is able to justify why two constants have been deemed to represent the same entity.

In a nutshell, LACE is a declarative language that, inspired by the Dedupalog approach (Arasu, Ré, and Suciu 2009), employs hard and soft rules to specify conditions under which pairs of entity references must or may be merged. A *hard rule* takes the form $q(x, y) \Rightarrow \mathsf{EQ}(x, y)$, where $q(x, y)$ is a *conjunctive query (CQ)* composed by standard relational atoms and atoms using similarity predicates, and EQ is a special symbol used to store merges. Intuitively, such a rule states that (c_1, c_2) being an answer to q provides sufficient conditions for concluding that c_1 and c_2 refer to the same entity. *Soft rules* have a similar form $q(x, y) \dashrightarrow \mathsf{EQ}(x, y)$, but state instead that (c_1, c_2) being an answer to q provides reasonable evidence for c_1 and c_2 denoting the same entity. In addition to rules, LACE specifications may include *denial constraints* to enforce consistency of the resulting database and constrain the allowed combinations of merges. Formally, we have the following definition.

Definition 1. *An* ER specification *takes the form* $\Sigma = \langle \Gamma, \Delta \rangle$*, where* $\Gamma = \Gamma_h \cup \Gamma_s$ *is a finite set of hard and soft rules, and* Δ *is a finite set of denial constraints.*

We equip LACE with a 'dynamic' and 'global' semantics. In line with works on matching dependencies (MDs) (Bertossi, Kolahi, and Lakshmanan 2013), rule bodies are evaluated on induced databases resulting from applying the already 'derived' merges. It is thanks to the dynamic nature of the semantics that we obtain a collective yet justifiable framework, in which merges can trigger further merges, possibly in a recursive fashion, while still being able to trace back the origins of each merge. In contrast to MDs and in line with Dedupalog and the entity linking (EL) approach (Burdick et al. 2016), our semantics is 'global' since LACE globally merges constants by replacing one constant with the other *everywhere* in the database. Further, as in EL, we consider a space of maximal (w.r.t. set inclusion) solutions, which in our case emerges from adopting denial constraints to enforce consistency of the resulting database and restricting which merges can be performed together, effectively creating choices.

Example 1. *An example of an ER specification in LACE is* $\Sigma = \langle \{\rho, \sigma\}, \{\delta\} \rangle$*, with* ρ*,* σ*, and* δ *defined as follows:*

- ρ *is a hard rule stating that papers with similar titles and presented at the same conference must be the same:*

$$\mathit{Paper}(x, t, c) \wedge \mathit{Paper}(y, t', c) \wedge t \approx t' \Rightarrow \mathsf{EQ}(x, y)$$

- σ *is a soft rule stating that conferences with similar names and held in the same year are likely to be the same:*

$$\mathit{Conference}(x, n, ye) \wedge \mathit{Conference}(y, n', ye) \wedge n \approx n' \dashrightarrow \mathsf{EQ}(x, y)$$

- δ *is a denial constraint that states that there cannot be two distinct chairs for the same conference:* $\forall z, w. \neg(\mathit{Chair}(z, w) \wedge \mathit{Chair}(z, w') \wedge w \neq w')$.

Now consider the database whose facts are:

$$\mathit{Conference}(c_1, \mathit{Conf.\ Data\ Eng.}, 2020), \mathit{Chair}(c_1, a_1)$$
$$\mathit{Conference}(c_2, \mathit{Data\ Eng.\ Conf.}, 2020),$$
$$\mathit{Conference}(c_3, \mathit{Data\ Eng.\ \&\ An.}, 2020), \mathit{Chair}(c_3, a_2)$$
$$\mathit{Paper}(p_1, \mathit{Survey\ on\ ER}, c_1), \mathit{Paper}(p_2, \mathit{ER\ Survey}, c_2)$$

Here, assuming Conf. Data Eng. $\approx$ *Data Eng. Conf. and Data Eng. Conf.* $\approx$ *Data Eng. & An., then we can use soft rule* σ *to merge* (c_1, c_2) *or* (c_2, c_3)*, but we cannot perform both merges, otherwise* c_1 *and* c_3 *would be deemed to be the*

same, in violation of δ. If we decide to include (c_1, c_2), then the hard rule ρ forces the merge (p_1, p_2) due to our dynamic and global semantics.

Contributions. Our first contribution in the paper is the presentation of the syntax and the semantics of LACE. We then provide a comprehensive study of the data complexity of the relevant computational tasks. Our results can be summarized as follows:

1. The problems of existence and recognition of maximal solutions are complete for NP and coNP, respectively, but recognition of arbitrary solutions is P-complete.
2. Recognizing *certain merges* (i.e. merges occurring in every maximal solution) is Π_2^p-complete, while the dual problem of identifying *possible merges* is NP-complete.
3. We define *certain* and *possible* query answers (w.r.t. the set of maximal solutions) and show that the associated decision problems have the same complexity as the problems in Point 2.
4. We investigate the impact of imposing syntactic restrictions. While the hardness results hold if denial constraints consist solely of *functional dependencies*, if one considers denial constraints without inequalities, then several of the problems decrease in complexity. More drastic restrictions ensure tractability of all considered problems.

Towards the development of an ER system based on LACE, our third contribution is an encoding of solutions as stable models of logic programs, which we use to show how the various tasks can be solved using answer set programming (ASP) (Lifschitz 2019). In particular, maximal solutions (w.r.t. set inclusion) can be handled using the meta-programming approach (Gebser, Kaminski, and Schaub 2011) or the Aspirin framework (Brewka et al. 2015).

As a final contribution, we explore the differences in the semantics of EL and LACE and their capability to capture recursive ER scenarios. In particular, we exhibit one such scenario that is easily captured in LACE, but is provably not expressible in EL.

Perspectives. This promising initial investigation opens up many interesting research directions, including:

- Local merges: We believe that the LACE and MD approaches are complementary, and it would be fruitful to combine them to obtain a framework that allows for both global and local merges.
- Quantitative extensions: Using set inclusion to define good solutions might in some cases be too coarse, so it would be interesting to equip rules with quantitative information and use it to assign weights or probabilities to merges and solutions.
- Negative rules: Our hard and soft rules indicate, respectively, mandatory merges and likely merges. It would be interesting to include also 'negative' rules, which can be used to indicate references that must or may be different, and to compare the evidence for and against a merge.
- Repairs and deduplication: While merges can resolve some constraint violations (i.e. those resulting from different representations of the same entity), a holistic framework for data quality will need to combine ER with traditional database repair operations (Bertossi 2011).
- Ontologies: It would also be relevant to enrich LACE with ontological information and to explore ER in the context of ontology-based data integration (Poggi et al. 2008).
- Implementation: We plan to develop an efficient prototype based on the presented ASP encodings and test it on existing ER benchmarks (Köpcke, Thor, and Rahm 2010).

Overall, we believe that the ER problem, which has mostly been studied within the database community (see (Bahmani et al. 2012) for a notable exception), could benefit greatly from KR techniques from various subareas, such as reasoning with inconsistencies, reasoning with uncertainty, ontologies, explanation, and non-monotonic reasoning.

References

Arasu, A.; Ré, C.; and Suciu, D. 2009. Large-scale deduplication with constraints using dedupalog. In *Proc. of ICDE 2009*, 952–963.

Bahmani, Z.; Bertossi, L. E.; Kolahi, S.; and Lakshmanan, L. V. S. 2012. Declarative entity resolution via matching dependencies and answer set programs. In *Proc. of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*.

Bertossi, L. E.; Kolahi, S.; and Lakshmanan, L. V. S. 2013. Data cleaning and query answering with matching dependencies and matching functions. *TOCS* 52(3):441–482.

Bertossi, L. E. 2011. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Brewka, G.; Delgrande, J. P.; Romero, J.; and Schaub, T. 2015. asprin: Customizing answer set preferences without a headache. In *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 1467–1474.

Burdick, D.; Fagin, R.; Kolaitis, P. G.; Popa, L.; and Tan, W. 2016. A declarative framework for linking entities. *TODS* 41(3):17:1–17:38.

Gebser, M.; Kaminski, R.; and Schaub, T. 2011. Complex optimization in answer set programming. *Theory Pract. Log. Program.* 11(4-5):821–839.

Köpcke, H.; Thor, A.; and Rahm, E. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proc. of the VLDB Endowment* 3(1):484–493.

Lifschitz, V. 2019. *Answer Set Programming*. Springer.

Newcombe, H. B.; Kennedy, J. M.; Axford, S. J.; and James, A. P. 1959. Automatic linkage of vital records. *Science* 130(3381):954–959.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *Journal of Data Semantics* 10:133–173.

Singla, P., and Domingos, P. M. 2006. Entity resolution with markov logic. In *Proc. of ICDM 2006*, 572–582.