Erratum for 'Query Expressibility and Verification in Ontology-Based Data Access'

Carsten Lutz, Johannes Marti, and Leif Sabellek

November 1, 2019

This erratum discusses a mistake in the paper regarding the relationship between the expressibility problem and the verification problem. In the paragraph before Theorem 6, it is claimed that computing $\mathbf{M}(q_s)$ yields a polynomial time reduction from expressibility to verification. However, this statement is false (unless P = NP) since computing $\mathbf{M}(q_s)$ involves checking the existence of homomorphisms between relational structures, which is NP-complete. We thank Gianluca Cima for pointing out this mistake.

Nevertheless, all theorems and lemmas continue to hold. In the paper, we relied on the incorrect statement to prove upper bounds only for the verification problem and lower bounds only for the expressibility problem. So we need to argue that all claimed upper bounds also hold for the expressibility problem and the claimed lower bounds also hold for the verification problem.

Upper bounds for expressibility. The only place in the paper where we use the false statement to obtain an upper bound for expressibility is in the proof of Theorem 10, the Π_2^p upper bound for UCQ-to-UCQ expressibility in [DL-Lite$^{\mathcal{R}}_{\mathsf{horn}}$, GAV]. The Π_2^p upper bound for expressibility can be achieved by first computing $\mathbf{M}(q_s)$ using a polynomial time Turing machine that has access to an NP-oracle: Recall that all mappings are either unary or binary. For every unary symbol $A \in \mathsf{sch}(\mathbf{M})$ and variable x of q_s, ask the NP-oracle whether $A(x) \in \mathbf{M}(q_s)$. For every binary symbol from $\mathsf{sch}(\mathbf{M})$, consider all pairs of variables instead. After $\mathbf{M}(q_s)$ has been computed, proceed as in the algorithm for verification.

Lower bounds for verification. All hardness proofs for expressibility can be modified to become hardness proofs for verification by constructing not only $\mathbf{M}$ and q_s, but also $\mathbf{M}(q_s)$ in the reduction. In both hardness proofs (Theorem 11 and Theorem 20) the query $\mathbf{M}(q_s)$ has a certain shape that can be computed in polynomial time within the reduction: In Theorem 11, the mappings are defined such that $\mathbf{M}(q_s) = \bigwedge_{i=0}^{n} r_i(y_0, y_1)$. In Theorem 20, $\mathbf{M}$ contains only identity mappings, so we have $\mathbf{M}(q_s) = q_s$.