APPENDIX TO “ENABLING PERSONAL CONSENT IN DATABASES”, BY KONSTANTINIDIS ET. AL

Proof. THEOREM 4.4 We need to prove that for all $t \in q^{\lambda}(D)$ and support queries $q_{\pi} = \pi_A(\hat{q}) \in \Pi_q$, there is a constraint q_N and a tuple $t_n \in q_N^{\lambda}(D)$ with $\pi_A(\vec{t}) \sqcap t_n$ iff there is a minimal w.r.t. set inclusion $S \subseteq body(q_{\pi})$ with a query unifier of q_{π} over q_N for S having query unification q_m, and $t_m \in q_m^{\lambda}(D)$ such that $\pi_A(\vec{t}) \sqcap t_m$.

$\Leftarrow$) First note that all query unifications, also the most general ones, of a query over a constraint are queries which are contained both in q_N and q_{π}. Moreover, for their query unifier θ, θ is a containment mapping from the query and the constraint to the unification such that all annotated answer tuples of the unification will “propagate” the same annotations both to the query and the constraint, appropriately.

$\Rightarrow$) Consider the homomorphisms h_{π} from q_{π} to D and h_N from q_N to D that obtain $\pi_A(\vec{t})$ and t_n respectively. Consider their composition and let database instance C (for “canonical”) the set of atoms in their image. Let $\vec{x}, \vec{y}$ the head terms respectively of q_{π}, q_N (with $|\vec{x}|=|\vec{y}|=|\vec{t_n}|=n$).

(Claim I): There is a minimal non-empty set of atoms $S_m \subseteq C$ that contains all terms in $\vec{t_n}$, and a piece unifier θ for a set $S' \subseteq body(q_{\pi})$ over a set $S_N \subseteq body(q_N)$ such that $\theta(S') \cup \theta(S_N) = S_m$ and for all atoms $\alpha \in S_m$ there is a unifiable pair of θ, (β, δ), and in at least one position that α contains term $\vec{t_n}[i]$, β and δ contain head terms $\vec{x}[i]$ and $\vec{y}[i]$ respectively. We prove (Claim I) by contradiction. Suppose that for all $S_m \subseteq C$ there exists an $i \in [1, n]$ for which for some atom $\alpha \in S_m$ with $\vec{t_n}[i] \in terms(\alpha)$ there is no $\beta \in body(q_{\pi})$ with $\vec{x}[i] \in terms(\beta)$, and no $\delta \in body(q_N)$ with $\vec{y}[i] \in terms(\delta)$, such that $\theta(\{\beta, \delta\}) = \{\alpha\}$. This means there are no mappings from q_{π} to S_m and from q_N to S_m that map two atoms, from q_{π} and q_N respectively, containing the i^{th} head term to the same atom α; if there were such mappings we could use their union (with no loss of generality we can assume q_{π},q_N use disjoint variable names) as a unifier θ. Since this is the case, on database C the i^{th} element of at least one tuple will be annotated with an identifier that will not be used for the i^{th} element of any resulting tuple of q_{π} nor the i^{th} element of any resulting tuple of q_N (since no atoms that contain the i^{th} element will map to our canonical database). However, this would prove that q_{π} and q_N do not “overlap” on C which is a contradiction, since, inutitively, C is just the part of D on which $\pi_A(\vec{t})$ intersects with t_N. Having proved (Claim I), note that θ is a most general query unifier of q_{π} over q_N for S'. We can union θ with the restriction of h_{π} and h_N on the variables not in the domain of θ, and obtain θ_m which is still a most general unifier of q_{π} over q_N for S'. Note also that since q_m maps to C by construction it will return an annotated answer tuple t_m on C (and D) that has the same base and the same schema as t_N with at least all annotation labels on each base term that t_N has on the same term. Thus it will be the case that $\pi_A(\vec{t}) \sqcap t_m$. □

Proof Sketch LEMMA 1 Notice that the outer SELECT statement executes the input query and the nested SELECT statement executes q_m without the conditions J_A and J_{θ}. Moreover, notice that $\vec{y}$ are terms from the input query ($\pi_A(\hat{q})$ simply changes the head of the query by projecting terms and including constants), so by using J_A we recognize answer tuples $\vec{t} \in q(D)$ and answer tuples $\vec{t_m} \in q_m(D)$ for which $\pi_A(\vec{t}) = \vec{t_m}$, for any database D. The only additional step that we need is to verify that the tuples that we remove, had we taken annotations into account, would be annotation-intersected with tuples t_m and those in $q_N^{\lambda}(D)$; we do this via J_{θ}, by enforcing that we remove tuples that came from the unified atoms, i.e. where all terms in the appropriate atoms β in the query are equated with terms in atoms α in q_m (α, β as in Thm 4.4). □

Proof. THEOREM 4.5 (The problem is in NP) First notice that there is a polynomial number of most general query unifications for a query and a constraint. These are bounded by the initial pairs of query-constraint atoms that our algorithm tries to unify (the `openECSets`). Further unifications are tried only for combinations of non successfully unified pairs, and successful combinations are exploited further. Moreover, for performing the actual unifications our algorithm goes over polynomially many equivalence classes trying to merge those that have common elements (each class has polynomially many elements). Hence, we need to check that $\vec{t}$ satisfies the queries of lemma 1 and overall we can check this in polynomial time. Thus the problem is in NP. (The problem is NP-hard) Deciding whether $\vec{t}$ is consent-abiding implies deciding whether $\vec{t}$ is an answer of the query to begin with (with no negative constraints at all) which is NP-hard[4]. □

Proof Sketch When can easily prove the contrapositive: if the constraint is not secure with respect to the query, there is a critical tuple shared between the queries, and this means there are homomorphisms from both queries to the same actual tuple. Therefore, there will be an (empty) answer tuple in the answer of both queries annotated with (at least) the same identifier coming from the critical tuple. Thus, the query will violate the constraint. □