OWL2 based Data Cleansing Using Conditional Exclusion Dependencies

Olivier CURE, Chan LE DUC and Myriam LAMOLLE

Université Marne La Valée, Université Paris8 - IUT de Montreuil

Motivation

- Our goal is to use the Ontology Based Data Access (OBDA) approach to improve data data quality of underlying databases
- This is performed with dependencies that capture real world inconsistencies : conditional dependencies
- Two forms have been studied : Conditional Functional Dependencies and Conditional Inclusion Dependencies (CINDs)
- We propose a novel one : Conditional Exclusion Dependencies (CEDs)

Definition

- A CED is an extension of Exclusion Dependencies that forbids the appearance of tuples in S when tuples satisfying some patterns appear in R.

Definition

- A CED is an extension of Exclusion Dependencies that forbids the appearance of tuples in S when tuples satisfying some patterns appear in R.

Syntax :

- a CED ϕ, defined over a pair of relations R and S, is a pair $(R(X; X_p) \subseteq \neg S(Y; Y_p), T_p)$ where X, X_p and Y, Y_p are attribute sets of respectively R and S.
- $R(X) \subseteq \neg S(Y)$ is a standard exclusion dependency and T_p is a tableau pattern of ϕ with attribute sets X_p and Y_p such that for each pattern t_p and each attribute B in X_p and Y_p, $t_p[B]$ is either a constant in the domain of B or a wild card, denoted $'\_'$.

Definition (2)

Semantics :

- An instance (I_1, I_2) of (R, S) satisfies a CED ϕ, denoted $(I_1, I_2) \models \phi$, iff for each tuple t_p in T_p and for each t_1 in I_1, if $t_1[X_p] = t_p[X_p]$ then there does not exist a tuple t_2 in S such that $t_1[X] = t_2[Y]$ and $t_2[Y_p] = t_p[Y_p]$.

OWL2 constructors

- Discovery of (C)ED is a hard problem since RDBMS do not store negative facts.
- OWL2 corresponds to the $\mathcal{SROIQ}$ description logic and allows for new role constructors.
- Some of them can be used to discover CEDs :
  - RBox axioms of the form $R \sqsubseteq \neg S$ with R and S DL roles.
  - General Concept Inclusion (GCI) of the form $\exists R.C \sqsubseteq \neg S.D$ with C a nominal and D a nominal or $\top$.
  - Negative property assertions.

SPARQL queries

- CEDs are represented using the formalism of SPARQL
- They aim to detect violations of CEDs and are generated by considering a CED has a graph over elements of the domain ontology.
- In this graph, the negated property is asserted to be true. Thus a translation of this graph into an SPARQL query enables to detect objects being violated.
- We compact these queries whenever some tableaux are compatible and/or using elements of concept hierarchies

Detecting violation

- Violation detections of a CED violation are activated whenever a tuple of the data sources is updated, via an SQL query
- This is handled by the definition of SQL triggers.
- We automatically generate an BEFORE/ROW LEVEL SQL trigger for each relation mapped to a property involved in a CED.

Future Work

- Propose algorithms to generate and tune CED using instances of the database
- Study interactions between conditional functional, inclusion and exclusion dependencies
- Test efficiency of the approach in larger application domains

Questions ?

Thank you

ocure@univ-paris-est.fr