Interactive Mapping Specification with Exemplar Tuples

Angela Bonifati  Ugo Comaignani  Emmanuel Coquy  Romuald Thion
angela.bonifati@univ-lyon1.fr  ugo.comaignani@univ-lyon1.fr  emmanuel.coquy@univ-lyon1.fr  romuald.thion@univ-lyon1.fr
CNRS LIRIS and Université Claude Bernard Lyon 1

1. Motivations and goals
We propose an Interactive Mapping Specification process targeting non-expert users that:

- bootstraps with a set of exemplar tuples \( (E_S, E_T) \), corresponding to a limited number of tuples provided as input;
- challenges the non-expert users with simple boolean questions, which are affordable for such users;
- is guaranteed to always produce a GLAV mapping \( M' \) that generalizes a mapping \( M \) in the user’s mind, which is unknown beforehand.

2. Proposed framework
Prior to the interactive process, a normalization step is undertaken in order to simplify the mappings and make them self-explanatory for non-expert users. Then, an atom refinement step and a join refinement step are respectively devoted to get the normalized mapping as closer as possible to the mapping that the user has in mind. This interactive process leverages simple boolean questions on even smaller refinement-driven tuples, derived from the initial exemplar tuples provided as input. We guarantee that (i) the obtained refined mappings are in normal form and that (ii) they are more general than the canonical mapping.

3. Specifying mappings using user exemplar tuples

<table>
<thead>
<tr>
<th>User exemplar tuples:</th>
<th>Canonical mapping:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_S: )</td>
<td>( E_T: )</td>
</tr>
<tr>
<td>Company</td>
<td>Firm</td>
</tr>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Town</td>
<td>Town</td>
</tr>
<tr>
<td>Flight</td>
<td>Departure</td>
</tr>
<tr>
<td>TC</td>
<td>Lyon</td>
</tr>
</tbody>
</table>

Prior to the interactive process, a normalization step is undertaken in order to simplify the mappings and make them self-explanatory for non-expert users. Then, an atom refinement step and a join refinement step are respectively devoted to get the normalized mapping as closer as possible to the mapping that the user has in mind. This interactive process leverages simple boolean questions on even smaller refinement-driven tuples, derived from the initial exemplar tuples provided as input. We guarantee that (i) the obtained refined mappings are in normal form and that (ii) they are more general than the canonical mapping.

4. Atom refinement
**Principle:**
Remove superfluous atoms in the left-hand sides of normalized tgds.

**Exploration space:**
Given a tgd, the possible sets of left-hand side atoms can be represented by a semi-lattice where the lower levels are the smallest valid sets of left-hand side atoms.

**User feedback:**
During semi-lattice exploration, the user is asked about the target tuples generated using only the subset of source tuples corresponding to the atoms in one node at a time.

**Example with a bottom-up breadth-first approach:**

As the sole set validated is \( C_1 \cdot F_1 \), the following tgd is generated:

\[
\text{Company}(c_1, aa, paris) \land \text{Flight}(lyon, paris, \text{c1})
\rightarrow 3 \text{IdFirm}(\text{id1}, aa, paris) \land \text{Departure}(\text{lyon}, \text{id1}) \land \text{Arrival}(\text{paris}, \text{id1})
\]

5. Join refinement
**Principle:**
For each tgd generated by atom refinement, identify redundant joins entailed by multiple occurrences of a given variable.

**Exploration space:**
As with atom refinement, we can build a semi-lattice of partitions representing possible joins between variable occurrences.

**User feedback:**
Similarly to atom refinement, the user is asked about the validity of small sets of tuples.

6. Experimental study

- We have used seven real data integration scenarios of the iBench benchmark [2].
- We have simulated the user’s ambiguities by adding up to ten superfluous atoms or redundant joins to the initial exemplar tuples, derived from the above scenarios.
- We have tested four exploration strategies: bottom-up and top-down, each one with depth-first and breadth-first variations.

**References**