Limitations and challenges

- Rewritability of queries is not always possible
- Limited query language (e.g., no support for analytical operations like \( \min \), \( \max \), \( \text{count} \), \( \text{sum} \), \( \text{avg} \))
- Semantic mismatch between data sources and ontological level

Data sources

<table>
<thead>
<tr>
<th>Ontological level</th>
<th>Data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed world (CWA)</td>
<td>Sensor( \land )sensor1( \lor )sensor2( \lor )sensor3( \lor )sensor1( \land )sensor2( \land )sensor3( \land )turbine1( \land )turbine2\land \ldots \land )turbine\n\n</td>
</tr>
<tr>
<td>Null values existential quantification</td>
<td></td>
</tr>
</tbody>
</table>

⇒ Problematic with respect to supporting analytical operations

Optimisation of ontological queries is hard

### Approach

**What is the answer to the following questions?**

\( Q_1 \): What is the number of measurements?

\( Q_2 \): What is the maximum measurement value for each sensor?

**Query evaluation**

- ACQ query language: CQ + aggregate functions in head \([1, 2]\)

\( Q_1 \) \( q(\text{count}(y)) \leftarrow \exists x \text{hasMeasmt}(x, y) \)

\( Q_2 \) \( q(x, \text{max}(y)) \leftarrow \text{hasMeasmt}(x, y) \)

**Rewritability:** Are there \( Q_1' \) \( Q_2' \) over the data sources to answer \( Q_1 \) and \( Q_2 \) over \( O \)?

**Semantics**

- Extend DL-Lite\(_4\) with multi-set (bag) semantics
- \( \text{count-ACQs} \): (a) \( \text{minimal} \) \([2]\) and (b) \( \text{skolem-based} \) semantics
- Identify safe queries for which minimality is tractable
- \( \text{non-count-ACQs} \): aggregate over known values \([1]\)

### Expressiveness: Pushing the envelope further

**Can we answer the following questions using ACQs?**

- What is the lowest cost for flying from London to Rome?
- How many bolts does a car manufacturer need to order for a specific model?

### Challenges

- Apart from aggregation (e.g., \( \text{sum} \), \( \text{min} \), \( \text{and} \) \( \text{count} \)), we need also recursion to formulate/answer the above questions
- We need also recursion to answer the above questions
- Assume a rule-based language extended with aggregates
- Interaction between recursion and aggregation is very powerful and non-trivial to manage

### Expressiveness: Pushing the envelope further

**Previous work**

- No semantics for the general case
- Proposals in the literature unsatisfactory:
  - High complexity, no value invention \([3]\)
  - Undecidability of fact entailment \([4, 5, 6]\)
  - Limited expressivity (e.g., functionality) \([3, 6, 7]\)
  - Unnatural syntactic restrictions (hard to write programs) \([4]\)

**Our goal**

- Define intuitive semantics leading to a unique model
- A natural and user-friendly syntax
- Generalise existing approaches
- Low complexity of query evaluation; sufficient expressive power

**Example program for cheapest flights**

\( \text{flight}(X, Z, C) \leftarrow \text{flight}(X, Y, C_1), \text{flight}(Y, Z, C_2), C = C_1 + C_2 \)

\( \text{cheapest_flight}(X, Y, C) \leftarrow C = \min(C), \text{flight}(X, Y, C) \)

### References


