Large Scale Temporal RDFS Reasoning Using MapReduce

Chang Liu¹, Guilin Qi², Yong Yu³
¹Shanghai Jiao Tong University, China
²School of Computer Science and Engineering, Southeast University, China

Temporal RDF

<table>
<thead>
<tr>
<th>RDF</th>
<th>Temporal</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ChangLiu&gt;</td>
<td>&lt;isUndergradOf&gt;</td>
<td>2005, 2009</td>
</tr>
<tr>
<td>&lt;isUndergradOf&gt;</td>
<td>rdfs:propertyOf</td>
<td>&lt;hasAffiliation&gt;</td>
</tr>
<tr>
<td>&lt;isUndergradOf&gt;</td>
<td>rdfs:domain</td>
<td>&lt;Undergraduate&gt;</td>
</tr>
<tr>
<td>&lt;ChangLiu&gt;</td>
<td>&lt;Graduate&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;isGraduate&gt;</td>
<td>rdfs:propertyOf</td>
<td>&lt;hasAffiliation&gt;</td>
</tr>
<tr>
<td>&lt;Undergraduate&gt;</td>
<td>rdfs:subClassOf</td>
<td>&lt;Student&gt;</td>
</tr>
</tbody>
</table>

Temporal RDFS Entailment Rule

Subproperty 1

\[(Axdfs:subPropertyOfOfOf, B) \rightarrow (X, A, Y): \lambda_1 = (X, B, Y): \lambda_2 \]

Subproperty 2

\[(Axdfs:subPropertyOfOfOf, B) \rightarrow (X, A, Y): \lambda_1 = (X, X, Y): \lambda_2 \]

Subclass 1

\[(Axdfs:subClassOfOfOf, B) \rightarrow (X, A, F): \lambda_1 = (X, A, X): \lambda_2 \]

Subclass 2

\[(Axdfs:subClassOfOfOf, B) \rightarrow (X, A, F): \lambda_1 = (X, X, X): \lambda_2 \]

Typing 1

\[(Axafs:range, B) \rightarrow (X, A, Y): \lambda_1 = (X, A, Y): \lambda_2 \]

Typing 2

\[(Axafs:range, B) \rightarrow (X, A, Y): \lambda_1 = (X, A, Y): \lambda_2 \]

Large Scale Temporal RDFS Reasoning

Naive Implementation

Inefficient Implementation

Compact Representation

\[(\lambda, p, o)[\lambda_1, \lambda_2] \]

Loading Schema Triples Into Memory

Observation: Tbox is small

Fix-point Iteration

Recursive Rules

Subproperty 2

\[(Axdfs:subPropertyOfOfOf, B) \rightarrow (X, A, Y): \lambda_1 = (X, B, Y): \lambda_2 \]

Subclass 2

\[(Axdfs:subClassOfOfOf, B) \rightarrow (X, A, Y): \lambda_1 = (X, A, X): \lambda_2 \]

Shortest Path Calculation

1. Load schema triples (sp&sc) into memory
2. Perform in-memory all-pairs shortest path calculation to get sp&sc closures
3. Use these closures to calculate the results of other rules.