Correct information!

```rdf
@prefix : <http://nunolopes.org/foaf.rdf#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
:me foaf:name "Nuno Lopes" .
```

Incorrect information!

```rdf
:me foaf:workplaceHomepage <http://www.si.uevora.pt/> .
:me foaf:workplaceHomepage <http://www.deri.ie/> .
```
@prefix : <http://nunolopes.org/foaf.rdf#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:me foaf:name "Nuno Lopes" .
@prefix : <http://nunolopes.org/foaf.rdf#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:me foaf:name "Nuno Lopes" .

:me foaf:workplaceHomepage <http://www.si.uevora.pt/> .
:me foaf:workplaceHomepage <http://www.deri.ie/> .
Correct

@prefix : <http://nunolopes.org/foaf.rdf#>  .
@prefix foaf: <http://xmlns.com/foaf/0.1/>  .

:me foaf:name "Nuno Lopes"  .

Incorrect information!

:me foaf:workplaceHomepage <http://www.si.uevora.pt/>  .
:me foaf:workplaceHomepage <http://www.deri.ie/>  .
Annotated RDF

subject  predicate  object  annotation

wpH = workplaceHomepage

Annotations refer to a specific domain temporal trust (fuzzy) provenance...
Annotations refer to a specific domain, temporal trust (fuzzy), provenance, etc. 

subject   predicate   object   annotation


wpH = workplaceHomepage
Annotated RDF

Annotations refer to a specific domain

- temporal
- trust (fuzzy)
- provenance
- ...
**Domain Example - Provenance**

<table>
<thead>
<tr>
<th>Name</th>
<th>Relation</th>
<th>Supervisor</th>
<th>Webpage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>:hasSupervisor</td>
<td>William</td>
<td>Personal Webpage</td>
</tr>
<tr>
<td>Mary</td>
<td>:hasSupervisor</td>
<td>William</td>
<td>Faculty List</td>
</tr>
<tr>
<td>Max</td>
<td>:hasAdvisor</td>
<td>William</td>
<td>Faculty List</td>
</tr>
<tr>
<td>Max</td>
<td>:hasSupervisor</td>
<td>Stephen</td>
<td>Departmental Webpage</td>
</tr>
<tr>
<td>William</td>
<td>:hasSupervisor</td>
<td>Stephen</td>
<td>Graduate School</td>
</tr>
</tbody>
</table>

Diagram:

Partial order $\preceq$:

- "Personal Webpage" $\preceq$ "Departmental Webpage"
- "Faculty List" $\preceq$ "Graduate School"

Annotation domain: partially ordered set $(A, \preceq)$

- $A$ is the set of annotations
- $\preceq$ is the partial order (with a bottom element $\bot$)
Domain Example - Provenance

Annotated RDF

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Mary&quot;</td>
<td>:hasSupervisor</td>
<td>&quot;William&quot;</td>
<td>&quot;Personal Webpage&quot;</td>
</tr>
<tr>
<td>&quot;Mary&quot;</td>
<td>:hasSupervisor</td>
<td>&quot;William&quot;</td>
<td>&quot;Faculty List&quot;</td>
</tr>
<tr>
<td>&quot;Max&quot;</td>
<td>:hasAdvisor</td>
<td>&quot;William&quot;</td>
<td>&quot;Faculty List&quot;</td>
</tr>
<tr>
<td>&quot;Max&quot;</td>
<td>:hasSupervisor</td>
<td>&quot;Stephen&quot;</td>
<td>&quot;Departmental Webpage&quot;</td>
</tr>
<tr>
<td>&quot;William&quot;</td>
<td>:hasSupervisor</td>
<td>&quot;Stephen&quot;</td>
<td>&quot;Graduate School&quot;</td>
</tr>
</tbody>
</table>

Partial order $\preceq$:

- "Personal Webpage" $\preceq$ "Departmental Webpage"
- "Faculty List" $\preceq$ "Graduate School"

Annotation **domain**: partially ordered set $(\mathcal{A}, \preceq)$

- $\mathcal{A}$ is the set of annotations
- $\preceq$ is the partial order (with a bottom element $\bot$)
aRDF can introduce inconsistencies:

e.g. the triple \((Mary, \text{hasSupervisor}, William)\) in the previous example
aRDF can introduce inconsistencies:

e.g. the triple \((\text{Mary}, \text{hasSupervisor}, \text{William})\) in the previous example

"Faculty List" is not comparable to "Personal Webpage"

\[
\begin{array}{c}
\text{"Mary" :hasSupervisor "William". "Personal Webpage"}\\
\text{"Mary" :hasSupervisor "William". "Faculty List"}
\end{array}
\]

"Personal Webpage" \(\preceq\) "Departmental Webpage"
"Faculty List" \(\preceq\) "Graduate School"
aRDF can introduce inconsistencies:

- e.g. the triple \((Mary, \text{hasSupervisor}, William)\) in the previous example

> "Faculty List" is not comparable to "Personal Webpage"

```
"Mary" :hasSupervisor "William". "Personal Webpage"
"Mary" :hasSupervisor "William". "Faculty List"

"Personal Webpage" \(\preceq\) "Departmental Webpage"
"Faculty List" \(\preceq\) "Graduate School"
```

If the partially ordered set \(\mathcal{A}\) contains a top element \(\top\) the aRDF is guaranteed to be consistent.
RDFS schema

Supported vocabulary:

- `rdfs:subClassOf`
- `rdf:type`
- `rdfs:subPropertyOf`
  - *transitive* and *non-transitive* properties
RDFS schema

Supported vocabulary:

- `rdfs:subClassOf`
- `rdf:type`
- `rdfs:subPropertyOf`
  - *transitive* and *non-transitive* properties

Mentions that other RDFS constructs are possible, but consider `rdfs:subPropertyOf` particularly important.
A query is a triple (with possible variables)

\[ q = (\text{Max}, ?p, \text{William}) : (0.8, 2002) \]
A query is a triple (with possible variables)

\[ q = (Max, ?p, William) : (0.8, 2002) \]

Possible annotation answers: all \( a \in A \) where \( (0.8, 2002) \leq a \)

\[
A_\emptyset(q) = \{ \ldots, (Max, hasSupervisor, William) : (0.8, 2002), \ldots, (Max, hasSupervisor, William) : (0.9, 2003) \}
\]
A query is a triple (with possible variables)
\[q = (Max, ?p, William) : (0.8, 2002)\]

Possible annotation answers: all \(a \in A\) where \((0.8, 2002) \preceq a\)
\[A_\varnothing(q) = \{\ldots, (Max, hasSupervisor, William) : (0.8, 2002),\]
\[\ldots, (Max, hasSupervisor, William) : (0.9, 2003)\}\]

\textit{answer} is a set of triples, eliminating redundant annotation
\[Ans_\varnothing(q) = \{(Max, hasSupervisor, William) : (0.9, 2003)\}\]
Query answering algorithms

Algorithms for different types of queries:

- $\textit{atomicAnswerV} - (r, p, ?v) : a$
- $\textit{atomicAnswerP} - (r, ?p, v) : a$
- $\textit{atomicAnswerA} - (r, p, v) : ?a$

Polynomial complexity for these algorithms. Conjunctive query answering yield exponential complexity.
Experimental Results

- Tested using generated aRDF dataset ranging from 10,000 to 100,000 triples

![Graph 1: Time vs. Dataset size](image1)

![Graph 2: Time vs. Dataset size](image2)
Conclusions

- Representation capable of encompassing several annotations
- Consistency results for annotation domains
- no proper support for RDF schema
- no SPARQL
Conclusions

- Representation capable of encompassing several annotations
- Consistency results for annotation domains
- No proper support for RDF schema
- No SPARQL

Friday talk presenting our extensions to this work
Assuming fixed sets:

- $\mathcal{R}$ of resource names
- $\mathcal{P}$ of property names

$\text{dom}(p)$ set of values associated with property $p$

$(r, p, v)$: $a$ is an annotated triple if
- $r$ is a resource name
- $p$ is a property name
- $v$ is a value (may also be a resource)

- An annotated-RDF ontology $\mathcal{O}$ is a set of finite annotated triples
Ontology graphs

$V = \mathcal{R} \cup \bigcup_{p \in \mathcal{P}} \text{dom}(p)$

$E = \{(r, r') \mid (r, p, r') : a \in \mathcal{O}\}$

$\lambda(r, r') = \{p : a \mid (r, p, r') : a \in \mathcal{O}\}$ (edge labelling function)
Ontology graphs

- $V = \mathcal{R} \cup \bigcup_{p \in \mathcal{P}} \text{dom}(p)$
- $E = \{(r, r') \mid (r, p, r') : a \in \mathcal{O}\}$
- $\lambda(r, r') = \{p : a \mid (r, p, r') : a \in \mathcal{O}\}$ (edge labelling function)

Ontology Graph of the example on Slide 3:
Property paths

- for a transitive property, a p-path between nodes $r, r'$ are the triples $\{t_1 = (r, p_1, r_1) : a_1, \ldots, t_i = (r_{i-1}, p_i, r_i) : a_i, \ldots, t_k = (r_{k-1}, p_k, r') : a_k\}, \forall i \in [1, k](p_i, rdfs : subPropertyOf *, p)$
Property paths

- for a *transitive* property, a *p-path* between nodes \( r, r' \) are the triples \( \{ t_1 = (r, p_1, r_1) : a_1, \ldots, t_i = (r_{i-1}, p_i, r_i) : a_i, \ldots, t_k = (r_{k-1}, p_k, r') : a_k \}, \forall i \in [1, k] (p_i, rdfs : subPropertyOf*) , \ p \) 

- \( Univ \) denotes the set of all triples \( (r, p, v) \)

- An aRDF-interpretation \( I \) is a mapping from \( Univ \) to \( A \).
Semantics

Property paths

- for a *transitive* property, a *p-path* between nodes \( r, r' \) are the triples \( \{ t_1 = (r, p_1, r_1) : a_1, \ldots, t_i = (r_{i-1}, p_i, r_i) : a_i, \ldots, t_k = (r_{k-1}, p_k, r') : a_k \}, \forall i \in [1, k] (p_i, rdfs : subPropertyOf *) \)

- \( Univ \) denotes the set of all triples \( (r, p, v) \)

- An aRDF-interpretation \( I \) is a mapping from \( Univ \) to \( A \).

- An aRDF-interpretation \( I \) satisfies \( (r, p, v) : a \) iff \( a \preceq I(r, p, v) \).
Property paths

- For a transitive property, a p-path between nodes \( r, r' \) are the triples \( \{ t_1 = (r, p_1, r_1) : a_1, \ldots, t_i = (r_{i-1}, p_i, r_i) : a_i, \ldots, t_k = (r_{k-1}, p_k, r') : a_k \} \), \( \forall i \in [1, k] \) \((p_i, rdfs : subPropertyOf*, p)\)

- \( Univ \) denotes the set of all triples \((r, p, v)\)

- An aRDF-interpretation \( I \) is a mapping from \( Univ \) to \( A \).

- An aRDF-interpretation \( I \) satisfies \((r, p, v) : a \) iff \( a \preceq I(r, p, v) \).

- \( I \) satisfies \( \mathcal{O} \) iff:
  - \( I \) satisfies every \((r, p, v) : a \in \mathcal{O}\);
  - For all transitive properties \( p \in \mathcal{P} \), for all p-paths \( Q = \{t_1, \ldots, t_k\} \), \( t_i = (r_i, p_i, r_{i+1}) : a_i \), for all \( a \in A \) such that \( a \preceq a_i \), \( 1 \leq i \leq k \), \( a \preceq I(r_1, p, r_{k+1}) \).
aRDF query answering

- Two triples \((r, p, v) : a\) and \((r', p', v') : a'\) are *semi-unifiable* if there exists a substitution \(\theta\) such that \(\theta(r) = \theta(r')\), \(\theta(p) = \theta(p')\) and \(\theta(v) = \theta(v')\).
Two triples \((r, p, v) : a\) and \((r', p', v') : a'\) are semi-unifiable if there exists a substitution \(\theta\) such that \(\theta(r) = \theta(r')\), \(\theta(p) = \theta(p')\) and \(\theta(v) = \theta(v')\).

Given a consistent ontology \(\mathcal{O}\) and a query \(q = (r_q, p_q, v_q) : a_q\), then \(A_{\mathcal{O}}(q) = \{(r, p, v) : a\} \) s.t.

- \((r, p, v) : a\) is semi-unifiable with \(q\)
- \(\mathcal{O} \models (r, p, v) : a\)
- \((a\ is\ a\ variable) \lor \(a_q \preceq a\)\)
aRDF query answering

- Two triples \((r, p, v) : a\) and \((r', p', v') : a'\) are **semi-unifiable** if there exists a substitution \(\theta\) such that \(\theta(r) = \theta(r')\), \(\theta(p) = \theta(p')\) and \(\theta(v) = \theta(v')\).

- Given a consistent ontology \(\mathcal{O}\) and a query 
  \[q = (r_q, p_q, v_q) : a_q,\]
  then \(A_{\mathcal{O}}(q) = \{(r, p, v) : a\}\) s.t.
  - \((r, p, v) : a\) is semi-unifiable with \(q\)
  - \(\mathcal{O} \models (r, p, v) : a\)
  - \((a \text{ is a variable}) \lor (a_q \leq a)\)

- **Eliminate redundant triples:**
  An answer to \(q\) is \(\text{Ans}_{\mathcal{O}}(q) = \{(r, p, v) : a\}\) s.t.:
  - \((r, p, v) : a \in A_{\mathcal{O}}(q)\)
  - \(\forall S \subseteq \text{Ans}_{\mathcal{O}}(q) - \{(r, p, v) : a\} \text{ s.t. } S \models (r, p, v) : a\)