10/10/2007

Extending SparQL:
This part of my investigation focused on extension mechanisms for SparQL, such as user-defined functions. The SparQL specification allows for extensible value testing. This is to say that custom functions are part of the language spec, but only those that produce results suitable for testing values within FILTER clauses. The following is an example from the SparQL specification:

```
xsd:boolean   func:even (numeric value)

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX func: <http://example.org/functions#>
SELECT ?name ?id
WHERE { ?x foaf:name  ?name ;
          func:empId   ?id .
       FILTER (func:even(?id)) }
```

While this is a useful extension mechanism, it is not useful for my purpose, adding regular path matching to SparQL. ARQ, from Jena, however, adds an additional extension mechanism, *property functions* (aka magic properties). Using this extension mechanism is not without its drawbacks. The most significant drawbacks are:

1. Deviation from the standard – queries with function calls exploiting this extension mechanism will not run across all SparQL processors.
2. Query composition – queries with calls to property function will be more difficult to compose as non-materialized views. The functions, as black boxes, may need to be run or “materialized” when querying such a view.

Given the above limitations, it was still worth investigating whether or not such functions could be used for processing paths in graphs, both transitive closures and more arbitrary regular paths. As a first step, I looked at writing a transitive closure function that could produce the kleene closure over any single relationship from a given source. The signature of this function is as follows:

```
Subject ext:Closure ( Property ?Object )
```

Or, in an actual FMA query it would look like this:

```
PREFIX fma: <http://.../fmaOwlDlComponent_2_0#>
PREFIX ext:<java:edu.washington.sig.arq.>

SELECT ?pharynx_continuity
WHERE
{
 fma:Pharynx  ext:Closure ( fma:continuous_with ?pharynx_continuity )
}
```

The results of the above query are:

```
-----------------------
| pharynx_continuity  |
=======================
| fma:Pharynx         |
| fma:Small_intestine |
| fma:Esophagus       |
| fma:Stomach         |
-----------------------
```

The initial implementation of the above query worked only if both the subject and property were URIs. The implementation is done as a Java class (note prefix in query). The function is recursive and operates on ARQ’s internal data structures.