## Improvement of the query rewrite extension support

### Introduction to query rewrite extension

The query rewrite extension is the most expressive component of the GeoSPARQL standard, it simplifies the query graph pattern and makes it more understandable for users in terms of the specific domain knowledge. The most comprehensible way that people use to compare the relation of two geographical areas is to directly compare them, see Figure 1.

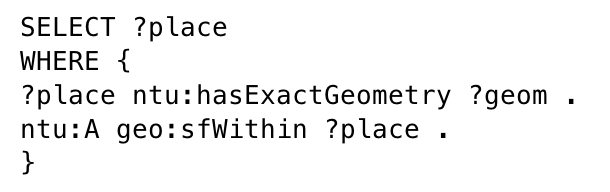


Figure : The direct comparison of two places.

Normally, such query will not yield any result since the query engine has no knowledge about the exact coordinates and spatial reference system of these two places. In fact, in order to compare the relation of two places, the exact geometry serialisations need to be compared rather than the place variables, see Figure 2.

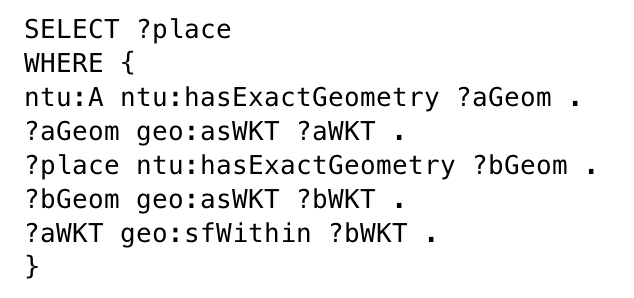


Figure : The most common way to compare two places.

With the implementation of the query rewrite extension, graph patterns like Figure 1 can be directly queried rather than explicitly pass the geometry serialisation to the comparison.

### The initial implementation of query rewrite extension

The initial implementation of the query rewrite extension is achieved by extending the query graph pattern at the SPARQL algebra level. The benefit of this approach is the extra graph patterns can be omitted from the query. Therefore, the topological relation can be compared through not only geometry literals, but also geometries or features. However, the initial query rewrite extension is not efficient because it is hard-coded to match the four comparison situations, such as feature to feature comparison, feature to geometry comparison, geometry to feature comparison, and the geometry to geometry comparison, see Figure 3. Such mechanism will first check the subject and the object from the graph which needs to be rewrote, then match the subject and the object to one of the four situations, finally extend the graph pattern at the algebra level. The query rewrite function will look similar to other topological property functions. Jena automatically match these registered property functions to the correct scenario.

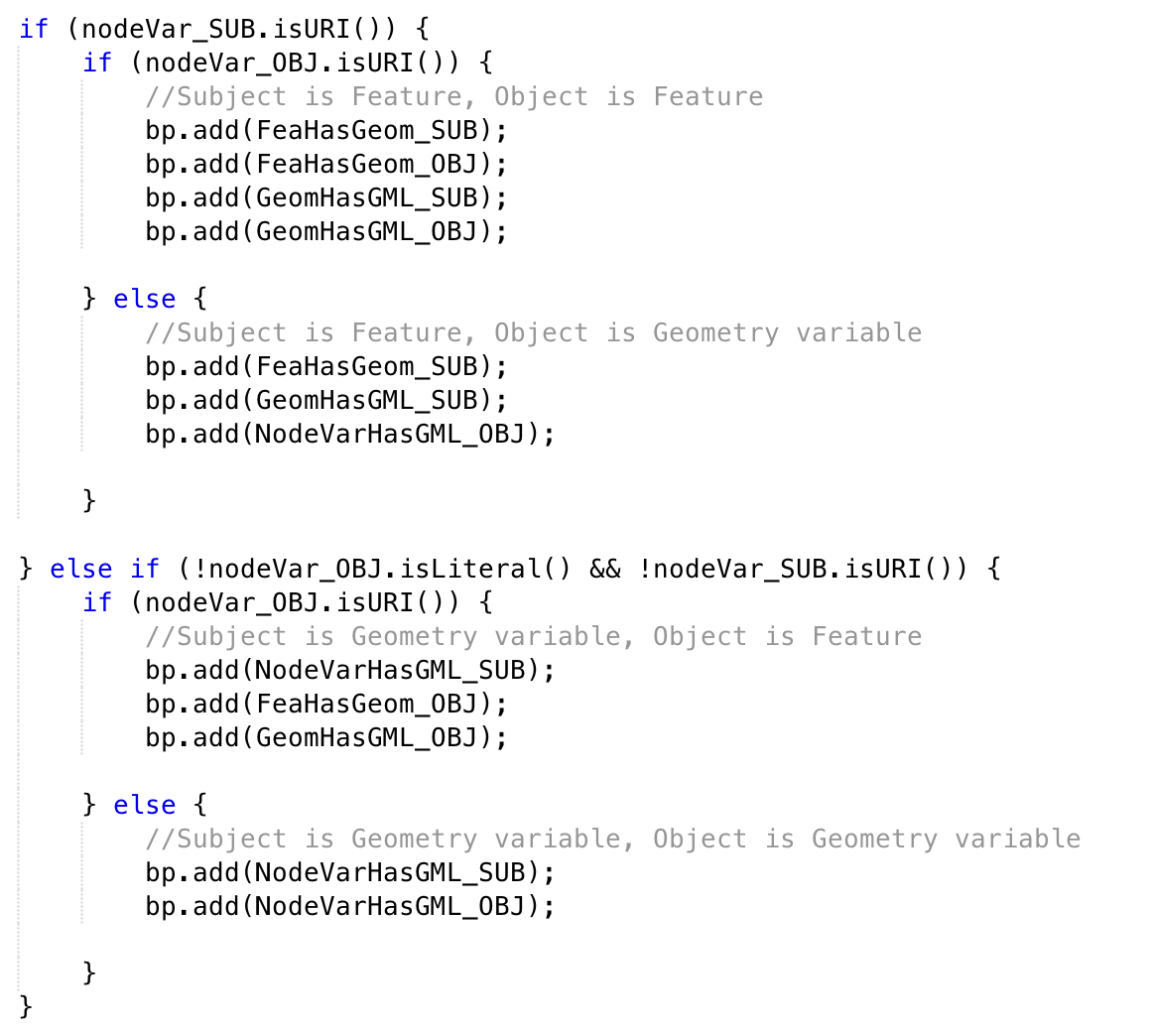


Figure : The initial hard-coded graph pattern matching.

### The improved version of query rewrite extension

Given the fact that the initial implementation of the query rewrite extension is hard-coded and lack of efficiency, an improvement has been made to clean up the query rewrite logic and improve the speed. As mentioned earlier, Jena will automatically match the registered property functions to the correct scenario. Based on this concept, an attempt had been made to separate the initial implementation of query rewrite extension to four sub query rewrite functions, each of them handles for a specific situation of graph pattern. In this case, the decision for using which sub query rewrite function is made by Jena, the only prerequisite is the pre-registration of all the query rewrite functions. For example, Figure 4 shows a feature to feature only query rewrite function.

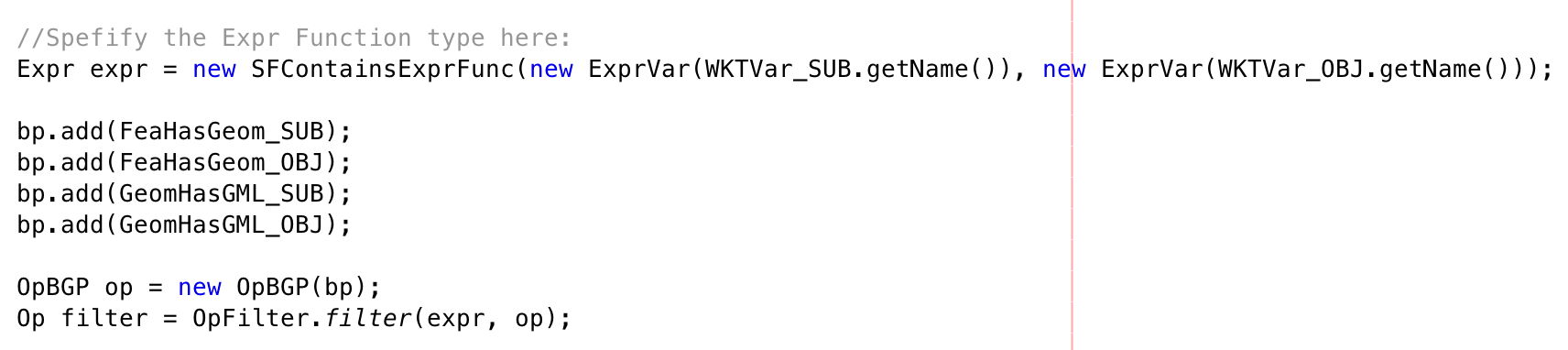


Figure : Code snipe of the feature to feature only query rewrite function.