Reasoning for knowledge enrichment

Part 1 : Medical ontologies exploration

My Corpus Fabrica (MyCF : https://mybody.inrialpes.fr/mycf/downloads.html) and Gene Ontology (GO : http://geneontology.org) are two ontologies that describe the anatomy and the genetic of human being.

MyCF allows answering questions like : *On which anatomical entities does knee stability carry?*

```
PREFIX mcf:<http://www.mycorporisfabrica.org/ontology/mcf.owl#>
SELECT distinct ?s
WHERE { ?s mcf:ContributesTo mcf:Body_stability }
```

However, executing this query over the ontology using Jena returns an empty result ! This is due to one simple fact, the knowledge that permits the answer of this query is implicit and not explicit, it also needs the integration of extra knowledge.

In other words, we have to add a new rule that enriches the property mcf:ContributesTo :

(?a, mcf:ContributesTo, ?c), (?c, mcf:IsInvolvedIn, ?b) -> (?a, mcf:ContributesTo, ?b)

This rule defines a relation between the properties *contributesTo* and *IsInvolvedIn* in a way that allows the rule engine interpreter to infer new relations. If we had to *materialize* this rule using SPARQL, we would have written the following request :

```
PREFIX mcf:<http://www.mycorporisfabrica.org/ontology/mcf.owl#>
SELECT distinct ?s
WHERE { ?s mcf:ContributesTo ?o .
        ?o mcf:IsInvolvedIn mcf:Body_stability }
```

Using the SPARQL construct pattern, write a query that will enrich the RDF model with new statements based on the described rule.
After the execution of your query, count the number of statements present in your model and execute the first query. You should obtain some interesting results.

Note : In practice, entering general relationships using rules allows medical experts to avoid entering all the relationships induced by hand, as well as to query all combinations of possible rule usage. Many knowledge bases in the field of life sciences (which constitute about 8% of Linked Open Data) use rules to describe all relationships between data.
The following example illustrates the value of enriching RDF knowledge using reasoning :

Part 2 : RDF Triple Store Saturation

Download the MyCF and GO Knowledge bases in your machine, as well as the QueryAndReasonOnLocalRDF.java file available on Moodle.

Using Jena and SPARQL, answer these questions :

1. What is the size (number of triplets) of each knowledge base before and after saturation?
2. Consider rdfs: subClassOf, part of and 3 other properties specific to each ontology of your choice. For each property, how many triplets are there before and after saturation?
3. Choose three rules for each ontology. How many triplets are inferred for each rule?
4. What is the inference time of the set of rules for each knowledge base?
   P.S. You should measure the execution time for the same machine.
5. Choose two classes of MyCF, one of which is subclass of mcf: Anatomical_entity and the other is subclass of mcf: Functional_entity, and calculate the number of respective subclasses. Measure time response to the query, using the three different reasoning engines (Hybrid, Forward, and Backward) implemented by Jena.
6. Choose two GO classes, and for each class, calculate the number of its instances. What's the time of response to the query, using the Hybrid, Forward, and Backward engines?