2. Unit: SPARQL Formal Semantics

Exercise 2.1 (SPARQL Formal Semantics) Consider the SPARQL Formal Semantics.

a) Define a “null-tolerant join” for the relational algebra that acts like the $\bowtie$ of the SPARQL algebra.

b) Which SQL construct is similar to the “\” operator in the SPARQL algebra?

c) In the SPARQL algebra, OPT is expressed via left outer join, which is defined via “\” (while a corresponding MINUS does not exist in the SPARQL syntax).
Such a MINUS (cf. part (b) of this exercise) provides a more intuitive idea of negation than “! bound(x)”. Give a general pattern how to express (P_1 MINUS P_2) in SPARQL 1.0 syntax.

d) Recall the definition of $\sqsupset\!\!\bowtie$ in the relational algebra (DB lecture) and define SPARQL’s $\sqsupset\!\!\bowtie$ in a similar way.

Exercise 2.2 (Outer Join) Recall that SPARQL’s OPTIONAL corresponds to a left outer join.

a) Give a general pattern how to express a *full* outer join (i.e., “outer” to both sides) in the SPARQL algebra (consider as input two mappings R and S and give an expression for $R \sqsupset\!\!\bowtie\!\!\sqsubset S$) and in SPARQL.

b) Give all cities (name as ?XN) that are the capital of a country (:capital) or that are located at a river (:locatedAt) or both (return the names ?CN of the country and/or the river (?RN)).

Exercise 2.3 (SPARQL Formal Semantics: OPTIONAL) Consider the SPARQL Formal Semantics.

Prove or show a counterexample:

The statement (from W3C SPARQL Working Draft 20061004)

> If OPT(A, B) is an optional graph pattern, where A and B are graph patterns, then S is a solution of OPT(A,B) if
>
> - S is a pattern solution of A and of B, or
> - S is a solution to A, but not to A and B.

describes the same semantics as above.