Universität Freiburg
Institut für Informatik
Prof. Dr. G. Lausen
Anas Alzoghbi
Anthony Arrascue

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
lausen@informatik.uni-freiburg.de
alzoghba@informatik.uni-freiburg.de
arrascue@informatik.uni-freiburg.de

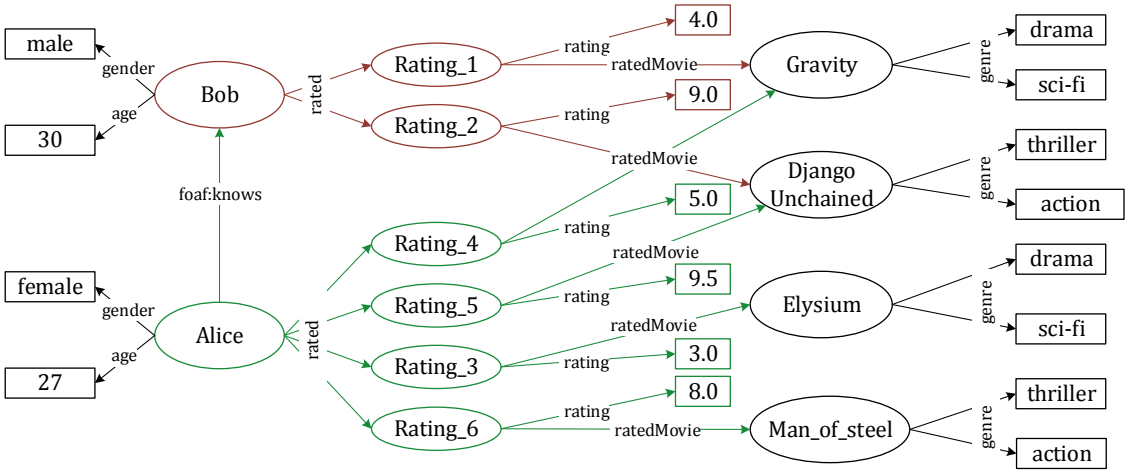
Data Models and Query Languages
Summerterm 2014

6. Exercise Sheet: SPARQL 1.1 & nSPARQL

Discussion: 22.07.2014

Submission Guidlines: Please hand out your written solutions directly to your tutors right before the exercise session. If you want to submit before the deadline, you can leave your solutions in the mail box in building 51-01 (first floor). Hand written solutions are also accepted as long as these are readable.

Exercise 1 (SPARQL 1.1: Aggregations, Subqueries, Explicit Negation, 6 Points)
Consider the following RDF graph in a movie domain, where information about ratings given by users on different movies in addition to movies genres are modelled in terms of RDF triples. The corresponding N3 file can be found in the attached file (movies.n3).



Formulate the following requests as SPARQL 1.1 queries. Evaluate them on the given RDF document and write down the final results.

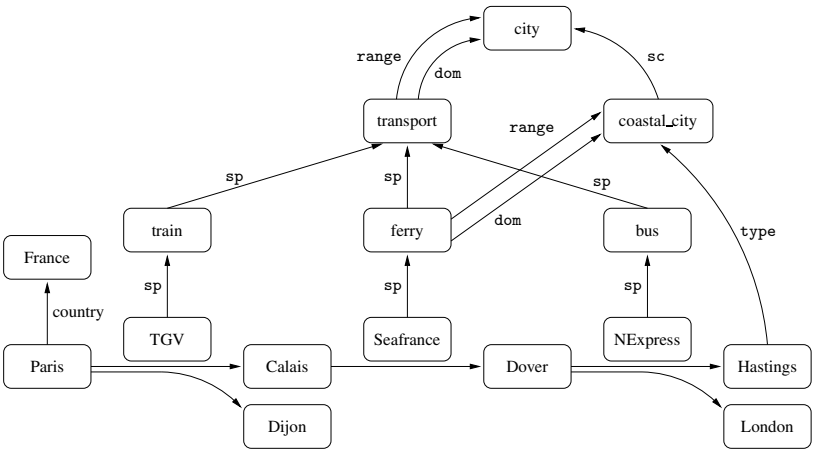
a) Compute the average rating of each user grouped by the genre of the movies he/she has rated. We call this average as the genre rating. (2Pts)

b) Find all users that have similar taste as Bob. Similar taste is defined as:
If user a has a genre rating (defined in a) r_a for genre g and user b has a genre rating r_b for genre g, then a and b have similar taste if for **each genre** the following inequality is true: $|r_a - r_b| < 1$.
This query should build on the previous query. The subquery provided by SPARQL 1.1 can help you in formulating the requested query. (4Pts)

c) Give a movie recommendation to Bob. A recommendation can be derived from the list of movies which where rated highly (>= 8) by other users who have similar taste (defined in b). This type of recommendation is known as Collaborative filtering. Proceed as following using the operations from SPARQL1.1:

- Aggregation to calculate the genre rating as in (a)
- Aggregation + subquery to find the similar users with similar taste as in (b)
- Subquery + negation to find movies rated highly by similar tasted users but have not been watched (rated) by Bob.

Exercise 2 (nSPARQL, 6 Points)
Consider the RDF Graph [1]:



a) Using the semantics of nested regular expressions defined in slides: (106 and 108), show the evaluation steps of the following nSPAQL expression and show the final result:
$P_1 = (?x, (next :: TGV \mid next :: Seafrance)^+, Dover)$

b) The following query finds the pairs of cities such that there is a way of travelling between those cities, and such that every stop in the trip is a coastal city. Change this query[1] to retrieve the pairs of cities which are connected via train or a ferry or any combination of both. And the destination city is a coastal city. (2Pts)
$P_2 = (?x, (trans(transport) \,/\, self :: [trans(type) \,/\, self :: costal\_city])^+, ?y)$

For the following set of nSPARQL expressions, give the final result after the evaluation on the given RDF graph, and formulate the nSPARQL expressions as SPARQL property path queries if possible or explain why they are not expressible.

c) $P_3 = (?x, (next :: Seafrance \mid next :: NExpress)^+ \,/\, self :: [next :: NExpress \,/\, self :: London]$
$\,/\, (next :: Seafrance \mid next :: NExpress)^+, ?y)$. (2Pts)

[1] Keep using trans()

d) $P_4 = (?x, next :: [(next :: sp)^* / self :: transport], ?y)$

e) $P_5 = (?x, (next :: [(next :: sp)^* / self :: transport])^+, ?y)$. (2Pts)

References:

1. Jorge Prez, Marcelo Arenas, Claudio Gutierrez: nSPARQL: A Navigational Language for RDF, 7th International Semantic Web Conference, 2008.