

Assignment 4: Query publication data from zbMATH

AI-1 Systems Project (Summer Semester 2026)

Jan Frederik Schaefer

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Informatik

Topic: Semantic web
Due on: August 14, 2026
Version from: June 19, 2026
Author: Jan Frederik Schaefer

Make sure you sign up before working on this assignment.^a

Using someone else's solution code, even as inspiration, is not allowed!
Sharing your solution code with other AISysProj students is not allowed.

^aYou can still decide to postpone the assignment. Signing up includes an eligibility check, which avoids situations where you invest work into an assignment that you are not supposed to take.

1 Task Summary

Translate a large dataset about mathematical publications into RDF triples and load them into a triplestore. Then use SPARQL queries to answer a set of questions. Figure 1 sketches the workflow for this assignment. A link to the dataset is posted in the assignment repository [AR]. For some of the problems, you need additional RDF data that was extracted from Wikidata, which you can also find in the assignment repository. This assignment **does not involve any data scraping**.

Didactic objectives

1. Get hands-on experience with RDF and SPARQL,
2. learn how to find information about standards like RDF and SPARQL,
3. get to know some of the challenges that come from working with large datasets.

Prerequisites and useful methods

1. The XML format and related technologies (in particular, SAX parsing might be useful for the large data set, and DOM parsing with XPath for the problem files),

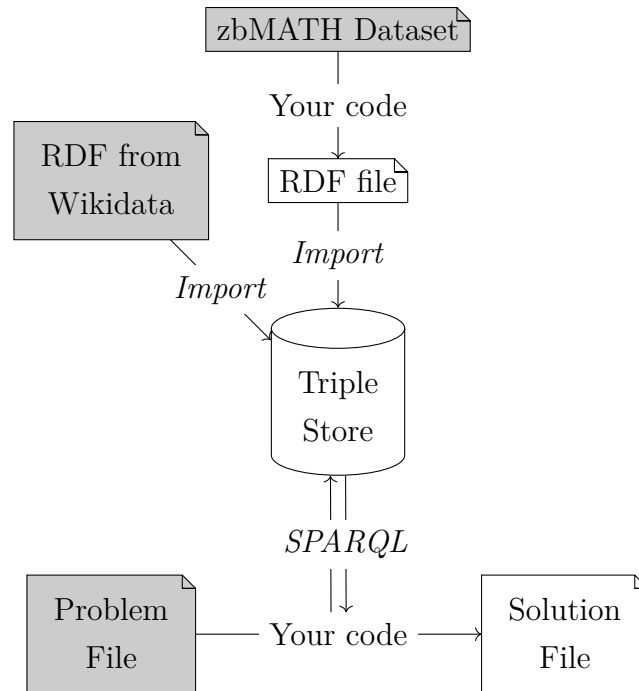


Figure 1: Workflow sketch for the assignment.

2. RDF, triple stores and SPARQL (the AI lecture introduces these concepts briefly, but there are also plenty of online resources),
3. formats for serializing RDF (e.g., Turtle, RDF/XML),
4. the basics of HTTP requests (for communicating with the triple store – ask for help if you have problems with this),
5. URLs and percent-encoding (also known as URL encoding).

2 The Dataset

zbMATH [ZBM] collects abstracts and reviews of papers in the area of mathematics and its applications. Recently, a lot of their data was made publicly accessible via an API as well as a dataset [Pet+]. Furthermore, we have created a mini dataset, which contains only a small subset of the entries and might therefore be easier to work with. A download link is in the assignment repository [AR]. You do not have to do any data scraping.

The datasets consist of records. Each record corresponds to one publication. The following entries in `metadata/oai_zb_preview:zbmath` are of interest to us:

- `zbmath:document_id` is an identifier for the publication.

- `zbmath:classifications/zbmath:classification` lists the classifications of the publication using the Mathematics Subject Classification [MSC].
- `zbmath:author_ids/zbmath:author_id` lists identifiers for the authors.
- `zbmath:keywords/zbmath:keyword` lists keywords of the publication.
- `zbmath:publication_year` is the year of the publication.

RDF requires that we use URIs to identify resources. These could be anything, but we will use the following URLs for this assignment:

Entry type	Example value	Associated URL
Document id	1448.68463	https://zbmath.org/?q=an%3A1448.68463
Classification	03B10	https://zbmath.org/classification/?q=cc%3A03B10
Author	kohlhase.michael	https://zbmath.org/authors/?q=ai%3Akohlhase.michael
Keyword	semantic web	https://zbmath.org/?q=ut%3Asemantic+web

3 Using Blazegraph

You should translate the datasets into RDF triples and load them into a triple store so that you can run SPARQL queries (you have to solve the problems using SPARQL queries, not using the original dataset). There are many different triple stores and you are free to pick one that works for you. In my (limited) experience, the open source triplestore Blazegraph [BG] is relatively easy to set up and use.

You can start Blazegraph with `java -jar blazegraph.jar`. It then runs on port 9999 (<http://localhost:9999/> gives you access to the blazegraph workbench). It also creates a journal file `blazegraph.jnl` to store the data.

You can update the data with a POST request to <http://localhost:9999/blazegraph/namespace/kb/sparql?update=XYZ> where XYZ is the (URL-encoded) update. For example, `DROP ALL` deletes all triples from the database and `LOAD <file:///path/to/triples.rdf>` loads data from a file.

Similarly, you can send a SPARQL query with a GET request to <http://localhost:9999/blazegraph/namespace/kb/sparql?query=XYZ> where XYZ is the (URL-encoded) SPARQL query.

4 Problems and Solution Format

Problems are provided in an XML file and the solutions should also be encoded as an XML file. There are separate problem files for the mini dataset and the full dataset. You can find

```
<Problems>
  <Problem id="0" type="...">
    ...
  </Problem>
  <Problem id="1" type="...">
    ...
  </Problem>
  ...
</Problems>
```

```
<Solutions>
  <Solution id="0">
    <Query>...</Query>
    ...
  </Solution>
  <Solution id="1">
    <Query>...</Query>
    ...
  </Solution>
  ...
</Solutions>
```

Listing 1: Example problems with their solutions. Every problem has an identifier that links it to the solution. The content of the problem and the solution depends on the problem type. Every solution should also include the SPARQL query for solving it. Note: You have to escape a few characters of the SPARQL query to include it in an XML file.

the problem files, along with example problems and solutions at [AR]. Listing 1 illustrates the structure of problems and solutions. Each problem should be answered with a single SPARQL query (and optionally some light post-processing). The following subsections will describe the different problem types.

4.1 Problem type: coauthors

List all co-authors of someone. This should be fairly straight-forward once you have loaded the data into Blazegraph. Listing 2 shows an example.

4.2 Problem type: msc-intersection

List all publications that have a specific set of classifications. Listing 3 shows an example that looks for publications combining group theory (classification 20) and knowledge representation in artificial intelligence (classification 68T30). One of the publications combining those classifications is 5155089, which has the classifications 68T30, 20F10, 68T37 and 68T05. Note that 20F10 is a subclassification of 20. In general, a classification of the form 01C34 is a subclassification of 01C and 01C. Similarly, a classification of the form 01-23 is

```

<Problem id="0" type="coauthors">
  <Author>https://zbmath.org/authors/?q=ai%3Aeinstein.albert</Author>
</Problem>

<Solution id="0">
  <Query>...</Query>
  <Author>https://zbmath.org/authors/?q=ai%3Aschrodinger.erwin</Author>
  <Author>https://zbmath.org/authors/?q=ai%3Apauli.wolfgang</Author>
  ...
</Solution>

```

Listing 2: Example problem and solution for problem type `coauthors`.

a subclassification of 01. It might make sense to add the subclassification relation to your dataset.

4.3 Problem type: ancestor-collab

List papers where the authors include two researchers X and Y where X is an academic ancestor of Y . We say that A is an academic parent of B if A was a PhD advisor of B . An academic ancestor can be an academic parent, grandparent, \dots , i.e. we take the transitive closure of the academic parent relation.

The zbMATH dataset does not contain information about academic ancestors, but Wikidata [WD] (a huge collaborative knowledge base) does. The assignment repository contains an RDF file with the relevant information. Wikidata uses different identifiers for people, but it has a predicate that links Wikidata entries to zbMATH author identifiers: <http://www.wikidata.org/prop/direct/P1556> It also has a predicate that links a person to their PhD students: <http://www.wikidata.org/prop/direct/P185>.

To keep the number of results manageable, you should only consider publications from a specific year. Listing 4 shows an example.

5 What to submit

At the deadline, we will download a snapshot of your repository. It should contain:

1. All your code for solving this assignment.

```

<Problem id="0" type="msc-intersection">
  <Classification>https://zbmath.org/classification/?q=cc%3A20</Classification>
  <Classification>https://zbmath.org/classification/?q=cc%3A68T30</Classification>
</Problem>

<Solution id="0">
  <Query>...</Query>
  <Paper>https://zbmath.org?q=an%3A5155089</Paper>
  <Paper>https://zbmath.org?q=an%3A647709</Paper>
  ...
</Solution>

```

Listing 3: Example problem and solution for problem type msc-intersection.

```

<Problem id="0" type="ancestor-collab">
  <Year>1922</Year>
</Problem>

<Solution id="0">
  <Query>...</Query>
  <Paper>https://zbmath.org/?q=an%3A2601836</Paper>
  <Paper>https://zbmath.org/?q=an%3A2600419</Paper>
  ...
</Solution>

```

Listing 4: Example problem solution for problem type ancestor-collab.

2. Solution files (both for the mini dataset and the full dataset).
3. The RDF generated for the mini dataset (you can compress it to reduce the file size). Please do **not** include the RDF for the full dataset to keep the repository size manageable. Similarly, do not include the Blazegraph files.
4. A README.md file explaining
 - i. dependencies (programming language, version, external libraries and how to get them),
 - ii. how to run your code to solve a problem file,
 - iii. the repository structure,
 - iv. anything else we should know.
5. A solution summary (see [SoS] for more details – it should describe the main ideas, not document the code).

6 Points

This is a relatively small assignment and only worth 80 points. You can get up to 20 points for the quality of the submission (README, evaluation, ...). Furthermore, you get 1 points for every correctly solved problem. There are 30 problems for the mini dataset and 30 problems for the entire dataset, which means that you can get a total of 60 points for the solutions. Alternatively, you can also get 2 points for every correctly solved problem for the entire dataset, and 0 points for the mini dataset so that you do not have to bother with the small dataset if you prefer. We will take the better of the two options (i.e., the one that gives you more points). Note that points will be subtracted if the solution files are not in the right format.

If the grading scheme doesn't seem to work well, we might adjust it later on (likely in your favor).

References

- [AR] *Repository for Assignment 4: Query publication data from zbMATH*. URL: <https://gitlab.rze.fau.de/wrv/AISysProj/ss26/a1.4-query-math-data/assignment>.
- [BG] *Welcome to Blazegraph*. URL: <https://blazegraph.com/> (visited on 02/04/2022).

- [MSC] *Mathematics Subject Classification – MSC2020*. URL: <https://zbmath.org/classification/> (visited on 02/04/2022).
- [Pet+] Matteo Petrera et al. “zbMATH Open: API Solutions and Research Challenges”. In: *Submitted to Joint Conference of Digital Libraries 2021 (JC DL '21), September 27–30, 2021, Online, Global*.
- [SoS] *Solution Summary*. URL: <https://gitlab.rrze.fau.de/wrv/AISysProj/admin/general/-/blob/main/solution-summary.md>.
- [WD] *Wikidata*. URL: <https://www.wikidata.org/> (visited on 11/21/2025).
- [ZBM] *zbMATH Open*. URL: <https://zbmath.org> (visited on 02/04/2022).