(Deep) FAIR Mathematics

Katja Berčič, Michael Kohlhase, Florian Rabe

Abstract: In this article, we analyze the state of research data in mathematics. We find that while the mathematical community embraces the notion of open data, the FAIR principles are not yet sufficiently realized. Indeed, we claim that the case of mathematical data is special, since the objects of interest are abstract (all properties can be be known) and complex (they have a rich inner structure that must be represented). We present a novel classification of mathematical data and derive a set of requirements, which we summarize as deep FAIR, which accomodate the special needs of math datasets. Finally, we show a prototypical system infrastructure, which can realize deep FAIRness for one category (relational data) of mathematical datasets.

ACM CCS: CCS → Information systems → Data management systems → Database design and models → Data model extensions → Semi-structured data

Keywords: Mathematical Data, deep FAIR

1 Introduction

Mathematical Datasets Modern mathematical research increasingly depends on collaborative tools, computational environments, and online databases, and these are changing the way mathematical research is conducted and how it is turned into applications. For example, engineers now use mathematical tools to build and simulate physical models based on systems of differential equations with millions of variables, combining building blocks and algorithms taken from libraries shared all over the internet.

Traditionally, mathematics has not paid particular attention to the creation and sharing of data — the careful computation and publication of logarithm tables is a typical example of the extent and method. This has changed with the advent of computer-supported mathematics, and the practice of modern mathematics is increasingly data-driven. Today it is routine to use mathematical datasets in the Gigabyte range, including both human-curated and machine-produced data. Examples include the L-Functions and Modular Forms Database (LMFDB; ~ 1 TB data in number theory) [Cre16; LM] and the GAP Small Groups Library [EBO] with ~ 450 million finite groups. In a few, but increasingly many areas, mathematics has even acquired traits of experimental sciences in that mathematical reality is "measured" at large scale by running computations.

There is wide agreement in mathematics that these datasets should be a common resource and be open and freely available. Moreover, the software used to produce them is usually open source and free as well. Such an ecosystem is embraced by the mathematics community as a general vision for their future research infrastructure [Cou14], adopted by the International Mathematical Union as the Global Digital Mathematics Library initiative [GDML].

To better understand the scale of the problem, Figure 1 gives an overview over some state-of-the-art datasets. Here we already use the division into four kinds of mathematical data that we will develop in Section 2.

State of FAIRness for Mathematical Datasets Mathematical datasets are generally produced, published, and maintained with virtually no systematic attention to the FAIR principles [FAIR18; Wil+16] for making data findable, accessible, interoperable, and reusable. In fact, often the sharing of data is an afterthought — see [Ber] for an overview of mathematical datasets and their "FAIR-readiness".

Moreover, the inherent complexity of mathematical data makes it very difficult to share in practice: even freely accessible datasets are often very hard or impossible to reuse, let alone make machine-interoperable because there is no systematic way of specifying the relation between the raw data and its mathematical meaning. Therefore, unfortunately FAIR mathematics essentially does not exist today.

Motivation Our ultimate goal is to standardize a framework for representing mathematical datasets. As a

Dataset	Description
Symbolic Data	
Theorem prover libraries	$\approx$ 5 proof libraries, $\approx 10^5$ theorems each, $\approx$ 200 GB
Computer algebra systems	e.g., SageMath distribution bundles $\approx$ 4 GB of various tools and libraries
Modelica libraries	$>$ 10 official, $>$ 100 open-source, $\approx$ 50 commercial, $>$ 5.000 classes in the Standard Library, industrial models can reach .5M equations
Relational Data	
Integer Sequences	$\approx$ 330K sequences, $\approx$ 1 TB
Sequence Identities	$\approx$ .3M sequence identities, $\approx$ 2.5 TB
Highly symmetric graphs, maps, polytopes	$\approx$ 30 datasets, $\approx 2 \cdot 10^6$ objects, $\approx$ 1 TB
Finite lattices	7 datasets, $\approx 17 \cdot 10^9$ objects, $\approx$ 1.5 TB
Combinatorial statistics and maps	$\approx$ 1.500 objects
SageMath databases	12 datasets
L-functions and modular forms	$\approx$ 80 datasets, $\approx 10^9$ objects, $\approx$ 1 TB
Linked Data	
zbMATH	$\approx$ 4M publication records with semantic data, $\approx$ 30M reference data, $>$ 1M disambig. authors, $\approx 2,7$M full text links: $\approx$ 1M OA
swMATH	$\approx$ 25K software records with $>$ 300K links to $>$ 180K publications
EuDML	$\approx$ 260K open full-text publications
Wikidata	34 GB linked data, thereof about 4K formula entities, interlinked, e.g., with named theorems, persons, and/or publications
Narrative Data	
arXiv.org	$\approx$ 300K math preprints (of $\approx$ 1.6M) most with LaTeX sources
EuDML	$\approx$ 260K open full-text publications, digitized journal back issues
MathOverFlow	$\approx 1,1$M questions/answers, $\geq$ 11K answer authors
Stacks project	$\geq$ 6000 pages, semantically annotated, curated, searchable textbook
nLab	$\geq$ 13K pages on category theory and applications

Figure 1: Summary of mathematical datasets

first step, we present MathDataHub, an infrastructure for systematically sharing relational datasets.

Such a standard for FAIR data representations in mathematics would lead to several incidental benefits:

- increased productivity for mathematicians by allowing them to focus on the mathematical datasets themselves while leaving issues of encoding, management, and search to dedicated systems,
- improved reliability of published results as the research community can more easily scrutinize the underlying data,
- collaborations via shared datasets that are currently prohibitively expensive due to the difficulty of understanding other researchers' data, including collaborations across disciplines and with industry practitioners, who are currently excluded due to the difficulty of understanding the datasets,
- reward mathematicians for sharing datasets (which is currently often not the case), e.g., by making datasets citable and their reuse known,
- more sustainable research by guaranteeing that datasets can be archived and their meaning understood in perpetuity (which is essential especially in mathematics).

Contribution In this article we survey and systematize how mathematical data is represented and shared and analyze how it enables or prevents FAIR mathematics. We pay particular attention to the mathematics-specific aspects of FAIR sharing, which, as we will observe, go significantly beyond the original formulation of FAIR.

As a first step towards a universal framework, and as a concrete example of FAIR-enabling mathematics-specific infrastructure, we introduce MathDataHub. This is a platform for sharing relational mathematical datasets in a way that systematically enables FAIRness.

Overview In the next section, we survey the particular challenges to FAIR sharing in mathematics. In Section 3, we develop the concept of "deep FAIR" to accommodate for the semantics issues, and in Section 4 we present a prototypical system that can help achieve them for the case of relational data. Section 5 concludes the article

2 FAIRness in Mathematics

2.1 General Considerations

The FAIR principles as laid out in, e.g., [Wil+16] are strongly inspired by scientific datasets that contain arrays or tables of simple values like numbers. In these cases, it is comparatively easy to achieve FAIRness. But in mathematics and related sciences, the objects of interest are often highly structured entities which are much less uniform. Moreover, the meaning and provenance of the data must usually be given in the form of complex mathematical data themselves — not just as simple metadata that can be easily annotated. Even more critically, while datasets in other disciplines are typically meant to be shared as a whole, it is very important for mathematical datasets to find, access, operate on, and

reuse individual entries or sets of entries of a dataset. As a consequence, the representation and modeling of mathematical data is much more difficult than anticipated in [Wil+16].

There are (at least) two aspects of FAIRness that are particularly important for mathematical data and are not strongly stressed in the original principles. The first one of these is that the data need to be semantics aware. Computer applications and mathematically sound, interoperable services can only work if the mathematical meaning of the data is FAIR in all its depth. We call this "deep FAIR" in this article.

Due to the mathematical standard of rigor and the inherent complexity of mathematical data, deep FAIRness is both more difficult and more important for mathematics than for other scientific disciplines. That also means that mathematics is an ideal test case for developing the semantic aspects of the FAIR principles in general.

The second one is that the relevant principles need to apply to every datum. The importance of this requirement, particularly for identifiers (Findable), has already been pointed out in [BT13]. For example, while it is good that a catalogue of graphs has a globally unique and persistent identifier, but it is much better if in addition to that, every graph in the catalogue also has one. This also extends to other FAIR principles.

In the sequel, we discuss the four FAIR principles and the challenges they pose for mathematical data in increasing order of difficulty.

Accessible While they often lack unique identifiers, most mathematical datasets are available online on researchers' websites or via repository managers like GitHub. Barriers typical for sensitive data are rare, and open sharing is common. However, the level of accessibility desirable in practice is much higher due to the wide variety of internal structure in mathematical datasets. Access to individual entries or the rich internal structure of these entries is less common.

Because each specialized tool is typically released with its own library, often written in tool-specific language, accessibility is very good for tool-associated data, but may be practically impossible across tools.

Reusable Mathematical datasets are typically not reusable or very hard to reuse in the sense of FAIR. First of all, they are often shared without licenses with the implicit, but legally false assumption that putting them online makes them public domain. In practice, this is often unproblematic because this false assumption of the publisher may be canceled out by the same false assumption by the reuser.

More critically, the associated documentation often does not cover how precisely the data was created or how the data is to be interpreted. This documentation is usually provided in ad hoc text files or implicitly in journal papers or software source code that potential users may not be aware of and whose detailed connection to the dataset may be elusive. And the lack of a standard for associating complex semantics and provenance data effectively precludes or impedes most reuse in practice.

Findable It is not common for datasets in mathematics to be indexed in registries. One often has to first find a paper describing the dataset, and then follow a link from there. The datasets themselves are sometimes searchable (such as [OEIS; Bri+13]), and the objects inside them often get a dataset-level unique identifier. This is particularly successful for bibliographic metadata (e.g. in Math Reviews, zbMATH or swMATH). However, for individual datasets, identifiers are often non-persistent, e.g., when shared on researchers' homepages.

Finding a mathematical object by its identifier or metadata is theoretically easy. But being findable in the sense of FAIR does not always imply being findable in practice: especially in mathematics, it is much more important to find objects by their semantic properties rather than by their identifier. The indexing necessary for this is very difficult.

For example, consider an engineer who wants to prevent an electrical system from overheating and thus needs a tight estimate for the term $\int_a^b |V(t)I(t)|dt$ for all a, b, where V is the voltage and I the current. Search engines like Google are restricted to word-based searches of mathematical articles, which barely helps with finding mathematical objects because there are no keywords to search for. Computer algebra systems cannot help either since they to do not incorporate the necessary special knowledge. But the needed information is out there, e.g., in the form of

Theorem 17. (Hölder's Inequality)
If f and g are measurable real functions, $l, h \in \mathbb{R}$, and $p, q \in [0, \infty)$, such that $1/p + 1/q = 1$, then

$$\int_l^h |f(x)g(x)|\,dx \leq \left(\int_l^h |f(x)|^p\,dx\right)^{\frac{1}{p}} \left(\int_l^h |g(x)|^q\,dx\right)^{\frac{1}{q}} \quad (1)$$

and will even extend the calculation $\int_a^b |V(t)I(t)|dt \leq \left(\int_a^b |V(x)|^2\,dx\right)^{\frac{1}{2}} \left(\int_a^b |I(x)|^2\,dx\right)^{\frac{1}{2}}$ after the engineer chooses $p = q = 2$ (Cauchy-Schwarz inequality). Estimating the individual values of V and I is now a much simpler problem.

Admittedly, Google would have found the information by querying for "Cauchy-Schwarz Hölder", but that keyword itself was the crucial information the engineer was missing in the first place. In fact, it is not unusual for mathematical datasets to be so large that determining the identifier of the sought-after object is harder than recreating the object itself.

Interoperable The FAIR principle base interoperability on describing data in a "formal, accessible, shared, and broadly applicable language for knowledge representation". But due to the semantic richness of mathematical data, defining an appropriate language to allow for interoperability is a hard problem itself. Therefore, existing interoperability solutions tend to be domain-specific, limited, and brittle.

For trivial examples, consider the dihedral group of order 8, which is called D_4 in SageMath but D_8 in GAP due to differing conventions in different mathematical communities (geometry vs. abstract algebra). Similarly, $0°C$ in Europe is "called" $271.3°K$ in physics. In principle, this problem can be tackled by standardizing mathematical vocabularies, but in the face of millions of defined concepts in mathematics, this has so far proved elusive. Moreover, large mathematical datasets are usually shared in highly optimized encodings (or even a hierarchy of consecutive encodings), which knowledge representation languages must capture as well to allow for data interoperability.

2.2 FAIRness for Different Kinds of Mathematical Data

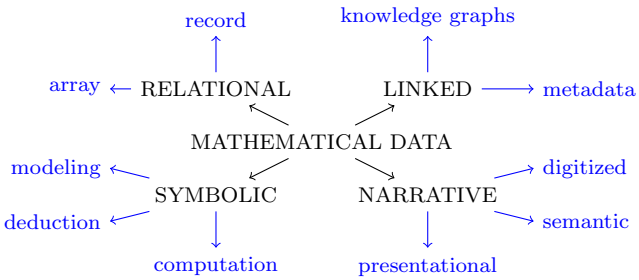

Figure 2: Kinds of mathematical data

In order to better analyze the current state of the art of FAIRness in mathematical data, we introduce a novel categorization of mathematical data. An overview is given in Figure 2. Each kind of data makes different abstractions or focuses on different aspects of mathematical reality, resulting in characteristic strengths and weaknesses. We summarize these in Figure 3.

Kind of data	Sym.	Rel.	Lin.	Nar.
Machine-understandable	+	+	+	−
Complete description	+	+	−	+
Applicable to all objects	+	−	+	+
Easy to produce	−	+	+	+

Figure 3: Advantages of different kinds of data

Symbolic data consists of formal expressions such as formulas, formal proofs, programs, etc. These are written in a variety of highly-structured formal languages specifically designed for individual domains and with associated tools. The most important such domains are **modeling**, **deduction**, and **computation** employing modeling languages, logics, resp. programming languages. The associated tools like simulation tools, proof assistants, resp. computer algebra systems can understand the entire semantics of the data.

Because symbolic data allows for abstraction principles such as underspecification, quantification, and variable binding, it can capture the complete semantics of any mathematical object. However, the formalization of a typical narrative theorem as a statement in a proof assistant or a function in a computer algebra system can be very expensive. This comes at the price of being context-sensitive: expressions cannot be easily moved across environments, which makes *Finding*, *Reusing*, and *Interoperability* difficult.

Moreover, because each tool usually defines its own formal language and because these are usually mutually incompatible, interoperability and reuse across these individual tools are practically non-existent. To overcome this problem, multiple representation formats have been developed for symbolic data, usually growing out of small research projects and reaching different degrees of standardization, tool support, and user following. These are usually optimized for specific applications, and little cross-format sharing is possible. In response to this problematic situation, standard formats have been designed such as MathML [MML310] and OMDoc/MMT [MMT].

Relational data employs representation theorems that allow encoding mathematical objects as ground data built from numbers, strings, tuples, lists, etc. Thus, relational data combines optimized storage and processing with capturing the whole semantics of the objects. It is also easy to produce and curate as general purpose database technologies and interchange formats such as CSV or JSON are readily available.

Relational datasets can be subdivided based on the structure of the entries, which often enable different optimized database solutions. The most important ones are record data, where datasets are sets of records conforming to the same schema and which are stored in relational databases, and **array** data, which consists of very large, multidimensional arrays stored in optimized array databases.

However, these representation theorems do not always exist because sets and functions, which are the foundation of most mathematics, are inherently hard to represent concretely. Moreover, the representation theorems may be very difficult to establish and understand, and there may be multiple different representations for the same object. Therefore, applicability is limited and must

be established on a case by case basis.

Therefore, *Interoperability* is difficult because users need to know the exact representation theorem and the exact way how it is applied to understand the encoding. Therefore, even if the representation function is documented, *Finding*, *Reuse*, and *Interoperability* are theoretically difficult, practically expensive, and error-prone. For example, consider the following very recent incident from (Jan. 2019): There are two encoding formats for directed graphs, both called `digraph6`: Brendan McKay's [McK] and the one used by the GAP package Digraphs [Beu+], whose authors were unaware of McKay's format and essentially reinvented a similar one [DG]. The resulting problem has since been resolved but not without causing some misunderstandings first.

Linked data introduces identifiers for objects and then treats them as blackboxes, only representing the identifier and not the original object. The internal structure and the semantics of the object remain unspecified except for maintaining a set of named relations and attributions for these identifiers. This abstraction allows for universal applicability at the price of not representing the complete mathematical object.

The named relations allow forming large networks of objects, and the attributions of concrete values provide limited information about each one. Linked data can be subdivided into **knowledge graphs** based on mathematical ontologies and **metadata**, e.g., as used in publication indexing services.

As linked data forms the backbone of the Semantic Web, linked data formats are very well-standardized: data formats come as RDF, the relations and attributes are expressed as ontologies in OWL2, and RDF-based databases (also called triplestores) can be queried via SPARQL. For example, services like DBPedia and Yago crawl various aspects of Wikipedia to extract linked data collections and provide SPARQL endpoints. The WikiData database [WD] collects such linked data and uses them to answer queries about the objects.

Thus, contrary to, linked data has very good FAIR-readiness, in particular allowing for URI-based *Access*, efficient *Finding* via query languages, and URI-mediated *Reuse* and *Interoperability*. However, this FAIR-readiness comes at the price of not capturing the complete semantics of the objects so that *Access* and *Finding* are limited and *Interoperability* and *Reuse* are subject to misinterpretation.

Narrative data consists of mathematical documents and text fragments. We speak of **mathematical vernacular** for the peculiar mixture of mathematical formulae, natural language with special idioms, and diagrams. There are four levels of formality of narrative data:

1. **digitized:** scanned into images from documents,
2. **presentational:** represented in a form that allows flexible presentation on electronic media, such as web browsers; born digital or OCRed from digitized ones,
3. **semantic**: in a form that makes explicit the functional structure and the relations between formulae, the objects they denote and the mathematical context.

All levels of formality are relevant for mathematical communication, but machine support for reasoning and knowledge management can only be given at the semantic level. We could extend this classification by a fourth level of narrative data for formalized documents; but these abstract from the narrative form and are therefore counted as symbolic data.

Note that we can always go from higher levels to lower ones, by styling: presenting semantic features by narrative patterns. Therefore we also count such patterns as narrative data – e.g. **notation definitions** such as $\binom{n}{k}$ or $\mathcal{C}_k^n$ for the binomial coefficients or verbalizations in different languages.

3 Deep FAIRness

Relational and linked data can be easily processed and shared using standardized formats such as CSV or RDF. But in doing so, the semantics of the original mathematical objects is not part of the shared resource: in relational data, understanding the semantics requires knowing the details of the representation theorem and the encoding; in linked data, almost the entire semantics is abstracted away anyway, which also makes it hard to precisely document the semantics of the links. For datasets with very simple semantics, this can be remedied by attaching informal labels (e.g., column heads for relational data), metadata, or free-text documentation. But this is not sufficient for datasets in mathematics and related scientific disciplines where the semantics is itself very complex.

For example, an object's semantic type (e.g., "polynomial with integer coefficients") is typically very different from the type as which it is encoded and shared (e.g., "list of integers"). The latter allows reconstructing the original, but only if its type and encoding function (e.g., "the entries in the list are the coefficients in order of decreasing degree") are known. Already for polynomials, the subtleties make this a problem in practice, e.g., consider different coefficient orders, sparse vs. dense encodings, or multivariate polynomials. Even worse, it is already a problem for seemingly trivial cases like integers: for example, the various datasets in the LMFDB use at least 3 different encodings for integers (because the trivial encoding of using the CPU's built-in integers does not work because the involved numbers are too big). But mathematicians routinely use much more complex objects like graphs, surfaces, or algebraic structures.

Service	Shallow	Deep
Identification	DOI for a dataset	DOIs for each entry
Provenance	who created the dataset?	how was each entry computed?
Validation	is this valid XML?	does this XML represent a set of polynomials?
Access	download a dataset	download a specific fragment
Finding	find a dataset	find entries with certain properties
Reuse	impractical without accessible semantics	
Interoperability	impossible without accessible semantics	

Figure 4: Examples of shallow and deep FAIR services

Data	Findable	Accessible	Interoperable	Reusable
Symbolic	Hard	Easy	Hard	Hard
Relational	Impossible without access to the encoding function			
Linked	Easy but only applicable to the small fragment of the semantics that is exposed			
Narrative	Hard	License-encumbered	Human-only	

Figure 5: Deep FAIR readiness of mathematical data

We speak of **accessible semantics** if data has metadata annotations that allow recovering the exact semantics of the individual entries of a data set. Notably, in mathematics, this semantics metadata is very complex, usually symbolic data itself that cannot be easily annotated ad hoc. But without knowing the semantics, mathematical datasets only allow FAIR services that operate on the dataset as a whole, which we call **shallow** FAIR services. But it is much more important to users to have **deep** services, i.e., services that process individual entries of the dataset.

Figure 4 gives some examples of the contrast between shallow and deep services. Note that deep services do not always require accessible semantics for every entry, e.g., deep accessibility can be realized without. But many deep services are only possible if the service can access and understand the semantics of each entry of the dataset, e.g., deep search requires checking for each entry whether it matches the search criteria.

In mathematics, shallow FAIR services are relatively easy to build but have significantly smaller practical relevance than deep FAIR services. Deep services, on the other hand, are so difficult to build that they are essentially non-existent except when built ad hoc for individual datasets. Figure 5 gives an overview of the difficulty for the different kinds of data.

Note that deep FAIR services are particularly desirable in mathematics, their advantages are by no means limited to mathematics. For example, in 2016 [ZEEO16], researchers found widespread errors in papers in genomics journals with supplementary Microsoft Excel gene lists. About 20% of them contain erroneous gene name because the software misinterpreted string-encoded genes as months. In engineering, encoding mistakes can quickly become safety-critical, i.e., if a dataset of numbers is shared without their physical units, precision, and measurement type. With accessible semantics, datasets can be validated automatically against their semantic type to avoid errors such as falsely interpreting a measurement in inch as a measurement in meters, a gene name as a month, or a column-vector matrix as a row-vector matrix.

In order to support the development Deep FAIR services for mathematics, we extend the original FAIR requirements from [Wil+16], which focused on shallow FAIR, to deep FAIR:

DF The internal structure of each object is represented and indexed in a way that allows searching for individual entries.

DA Each dataset includes a representation of the semantics of the represented objects.

DI The representation of each object uses a formal, accessible, shared, and broadly applicable language for knowledge representation, uses FAIRly shared vocabularies, and where applicable includes qualified references to other representations.

DR The representation of each object is richly described with a plurality of accurate and relevant attributes, is released with a clear and accessible data usage license, is associated with detailed provenance information, and meets domain-relevant community standards.

4 MathDataHub

We present a unified infrastructure to support Deep FAIR for relational mathematical data. It builds on our MathHub system, a portal for narrative and symbolic mathematical data. MathDataHub is a part of the MathHub portal and provides storage and hosting with integrated support for Deep FAIR. In the future, this will also allow for the development of mathematical query languages (i.e., queries that abstract from the encoding) and mathematical validation (e.g., type-checking relative to the mathematical types, not the database types).

To that end, we developed a mathematical data description language MDDL in [BKR19] (Math Data Descrip-

tion Language) that uses symbolic data to specify the semantics of relational data. MDDL schemas combine the low-level schemas of relational database with high-level descriptions (which critically use symbolic mathematical data) of the mathematical types of the data in the tables.

```
theory MatrixS : ?MDDL =
    include MitM:?IntegerMatrix
    mat: matrix ℤ 2 2 |
        meta ?Codecs?codec MatrixAsArray IntIdent |
    trace: ℤ |
        meta ?Codecs?codec IntIdent |
    orthogonal: bool |
        meta ?Codecs?codec BoolIdent | |
```

Figure 6: Schema theory for Joe's dataset

To fortify our intuition let us assume that Joe has collected a set of integer matrices together with their trace and the Boolean property whether they are orthogonal. Figure 6 shows a MDDL theory that describes his database schema. For example, the mathematical type of the field mat is integer 2×2 matrices; the codec annotation specifies how this mathematical type is be encoded as a low-level database type (in this case: arrays of integers). Concretely, the codec is MatrixAsArray codec operator applied to the identity codec for integers. These codec annotations capture the representation theorem that allows representing the mathematical objects as ground data that can be stored in databases.

The information is sufficient to generate a database schema – here one table with columns mat, trace, and orthogonal – as well as a database browser-like website frontend (see Figure 7). The generation of APIs for computational software such as computer algebra systems is also possible and currently under development.

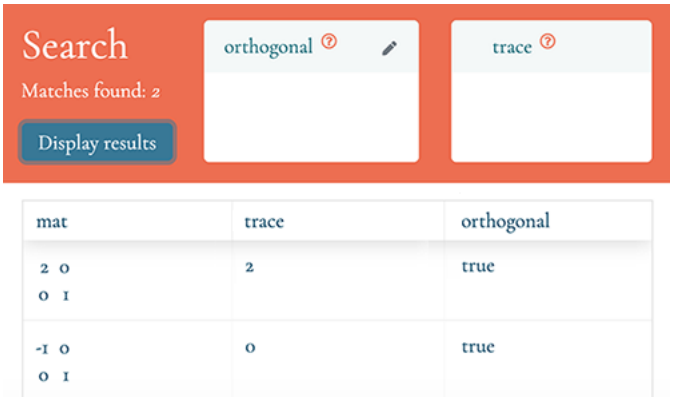

Figure 7: Website for Joe's dataset

Crucially, the codec-based setup transparently connects the mathematical level of specification with the database level – a critical prerequisite for the deep FAIR properties postulated above. Moreover, in Figure 6, the mathematical background knowledge is imported from a theory IntegerMatrix in the Math In The Middle ontology (MitM) [Mit], which supplies the full mathematical specification and thus the basis for *Interoperability* and *Reusability*; see [BKR19; WKR17; Koh+17] for details. The overhead of having to specify the semantics of the mathematical data is offset by the fact that we can reuse central resources like the MitM ontology and codec collection. Thus, MitM and MDDL form the nucleus of a common vocabulary for typical mathematical relational datasets.

5 Conclusion

In this paper we have analyzed the state of research data in mathematics with a focus on the instantiation of the general FAIR principles to mathematical data. We surveyed mathematical datasets, classified current practices of publishing and sharing them, and discussed the specific difficulties for FAIR practices.

In summary we found that realizing FAIR mathematical data is very difficult, much more so than for other disciplines. This is because mathematical data is inherently complex, so much so that datasets can only be understood (both by humans or machines) if their semantics is not only evident but itself suitable for automated processing. Thus, the accessibility of the mathematical meaning of the data in all its depth becomes a prerequisite to any strong infrastructure for FAIR mathematical data.

Based on these observations, we developed the concept of Deep FAIR research data in mathematics. As a first step towards developing a Deep FAIR–enabling standard for mathematical datasets, we focused on relational datasets We presented the prototypical MathDataHub system that lets mathematicians integrate a dataset by specifying its semantics using a central knowledge and codec collection. We hope that MathDataHub also helps alleviate the problem of *disappearing datasets*: Many datasets are created in the scope of small, underfunded or unfunded research projects, often by junior researchers or PhD students, and are often abandoned when developer change research areas or pursue a non-academic career.

References

[Ber] Katja Berčič. *Math Databases wiki*. URL: https://github.com/MathHubInfo/Documentation/wiki/Math-Databases (visited on 01/15/2019).

[Beu+] Jan De Beule et al. *GAP package Digraphs*. URL: https://www.gap-system.org/Packages/digraphs.html (visited on 01/25/2019).

[BKR19] Katja Berčič, Michael Kohlhase, and Florian Rabe. "Towards a Unified Mathematical Data Infrastructure: Database and Interface Generation". In: *Intelligent Computer Mathematics (CICM) 2019*. Ed. by Cezary Kaliszyck et al. LNAI 11617. Springer, 2019, pp. 28–43. DOI: 10.1007/978-3-030-23250-4.

[BKS17] Johannes Blömer, Temur Kutsia, and Dimitris Simos, eds. *MACIS 2017*. LNCS 10693. Springer Verlag, 2017.

[Bri+13] Gunnar Brinkmann et al. "House of Graphs: a database of interesting graphs". In: *Discrete Appl. Math.* 161.1-2 (2013), pp. 311–314. ISSN: 0166-218X. DOI: 10.1016/j.dam.2012.07.018.

[BT13] Sara C. Billey and Bridget E. Tenner. "Fingerprint databases for theorems". In: *Notices Amer. Math. Soc.* 60.8 (2013), pp. 1034–1039. ISSN: 0002-9920. DOI: 10.1090/noti1029.

[Cou14] National Research Council. *Developing a 21st Century Global Library for Mathematics Research*. Washington, DC: The National Academies Press, 2014. DOI: 10.17226/18619.

[Cre16] John Cremona. "The L-Functions and Modular Forms Database Project". In: *Foundations of Computational Mathematics* 16.6 (2016), pp. 1541–1553. ISSN: 1615-3383. DOI: 10.1007/s10208-016-9306-z.

[DG] *Digraph6 file format incompatibility*. URL: https://github.com/gap-packages/Digraphs/issues/158 (visited on 01/25/2019).

[EBO] Bettina Eick, Hans Ulrich Besche, and Eamonn O'Brien. *SmallGrp – The GAP Small Groups Library*. URL: https://www.gap-system.org/Manuals/pkg/SmallGrp-1.3/doc/chap1.html (visited on 10/13/2018).

[FAIR18] European Commission Expert Group on FAIR Data. *Turning FAIR into reality*. 2018.

[GDML] *GDML*. URL: https://en.wikipedia.org/wiki/Global_Digital_Mathematics_Library (visited on 01/28/2019).

[Koh+17] Michael Kohlhase et al. "Knowledge-Based Interoperability for Mathematical Software Systems". In: *MACIS 2017: Seventh International Conference on Mathematical Aspects of Computer and Information Sciences*. Ed. by Johannes Blömer, Temur Kutsia, and Dimitris Simos. LNCS 10693. Springer Verlag, 2017, pp. 195–210. URL: https://github.com/OpenDreamKit/OpenDreamKit/blob/master/WP6/MACIS17-interop/crc.pdf.

[LM] *The L-functions and Modular Forms Database*. URL: http://www.lmfdb.org (visited on 02/01/2016).

[McK] Brendan McKay. *Graph formats*. URL: http://users.cecs.anu.edu.au/~bdm/data/formats.html (visited on 01/25/2019).

[Mit] *MitM: The Math-in-the-Middle Ontology*. URL: https://mathhub.info/library/group?id=MitM (visited on 02/05/2017).

[MML310] Ron Ausbrooks et al. *Mathematical Markup Language (MathML) Version 3.0*. Ed. by David Carlisle, Patrick Ion, and Robert Miner. 2010. URL: http://www.w3.org/TR/MathML3.

[MMT] *MMT – Language and System for the Uniform Representation of Knowledge*. project web site. URL: https://uniformal.github.io/ (visited on 01/15/2019).

[OEIS] *The On-Line Encyclopedia of Integer Sequences*. URL: http://oeis.org (visited on 05/28/2017).

[WD] *Wikidata:Introduction*. URL: https://wikidata.org/wiki/Wikidata:Introduction (visited on 01/25/2015).

[Wil+16] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3 (2016). URL: https://doi.org/10.1038/sdata.2016.18.

[WKR17] Tom Wiesing, Michael Kohlhase, and Florian Rabe. "Virtual Theories – A Uniform Interface to Mathematical Knowledge Bases". In: *MACIS 2017: Seventh International Conference on Mathematical Aspects of Computer and Information Sciences*. Ed. by Johannes Blömer, Temur Kutsia, and Dimitris Simos. LNCS 10693. Springer Verlag, 2017, pp. 243–257. URL: https://github.com/OpenDreamKit/OpenDreamKit/blob/master/WP6/MACIS17-vt/crc.pdf.

[ZEEO16] Mark Ziemann, Yotam Eren, and Assam El-Osta. "Gene name errors are widespread in the scientific literature". In: *Genome Biology* 17.177 (2016). DOI: 10.1186/s13059-016-1044-7.

Dr. Katja Berčič Dr. Katja Berčič is a postdoc in the research group for Knowledge Representation/Processing (Computer Science) at FAU Erlangen-Nürnberg. While working on her PhD in mathematics (combinatorics) she became interested in how to improve the way mathematicians work with data. She followed this interest during a postdoc at the National Autonomous University of Mexico. Her research interests lie in the areas of knowledge representation and management, particularly for mathematics.

Address: Computer Science, FAU Erlangen Nürnberg, D-91059 Erlangen, E-Mail: Katja.Bercic@fau.de

Prof. Dr. Michael Kohlhase Dr. Michael Kohlhase is professor for Knowledge Representation/Processing (Computer Science) at FAU Erlangen-Nürnberg and adjunct associate professor for Computer Science at Carnegie Mellon University. His research interests include knowledge representation for STEM (science, technology, engineering, mathematics), inference-based techniques for natural language processing, computer-supported education, and user assistance. He pursues these (interrelated) topics focusing on the aspects of modular foundations (usually logical methods) and large-scale structures in document corpora. He has pursued these interests during extended visits to Carnegie Mellon University, SRI International, and the Universities of Amsterdam, Edinburgh, and Auckland.

Address: Computer Science, FAU Erlangen Nürnberg, D-91059 Erlangen, E-Mail: Michael.Kohlhase@fau.de

PD Dr. Florian Rabe Florian Rabe is a senior researcher at the Universities of Erlangen-Nuremberg and Paris-Sud. His research interests include logics, type systems, and programming languages for computer science and mathematics as well as knowledge representation, scalable implementation, and system interoperability for them. He is the creator and main author of the MMT language and system.

Address: Computer Science, FAU Erlangen Nürnberg, D-91059 Erlangen, E-Mail: florian.rabe@fau.de