

(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 known) and complex (they have a rich inner structure that must be represented). We present a novel classification of mathematical data and derive an extended set of FAIR requirements, which accommodate the special needs of math datasets. We summarize these as deep FAIR. Finally, we show a prototypical system infrastructure, which can realize deep FAIRness for one category (tabular 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, Findable, Accessible, Interoperable, Reusable

1 Introduction

Modern mathematical research increasingly depends on collaborative tools, computational environments, and online databases. 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.

Mathematical Datasets Traditionally, mathematics has not paid particular attention to the creation and sharing of data — the careful computation of logarithm tables and publication in an 18th century book 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) [Cremona:LMFDB16; lmfdb:on] and the GAP Small Groups Library [BeEiOBSSmallGroups] with ~ 450 million finite groups up to order 2000 (but only ~ 80 MB of data due to the clever use of the computer algebra system). 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 [NAS14], adopted by the International Mathematical Union as the Global Digital Mathematics Library initiative [GDML:on].

Here it is critical to understand that mathematical datasets are incredibly diverse (i.e., using fundamentally different kinds of data), complex (i.e., representing arbitrary mathematical objects), and large (e.g., millions of entries). This results in characteristic difficulties for FAIRness in mathematics. 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 [FAIR; WilDumAal:FAIR16] for making data findable, accessible, interoperable, and reusable. In fact, often the sharing of data is an afterthought —

Dataset	Description
Symbolic Data	
Theorem prover libraries	≈ 5 proof libraries, $\approx 10^5$ theorems each, ≈ 200 GB
Computer algebra systems	e.g., SageMath distribution bundles ≈ 4 GB of various tools and libraries
Modelica libraries	> 10 official, > 100 open-source, ≈ 50 commercial, > 5.000 classes in the Standard Library, industrial models can reach .5M equations
Tabular Data	
Integer Sequences	$\approx 330K$ sequences, ≈ 1 TB
Sequence Identities	$\approx .3M$ sequence identities, ≈ 2.5 TB
Highly symmetric graphs, maps, polytopes	≈ 30 datasets, $\approx 2 \cdot 10^6$ objects, ≈ 1 TB
Finite lattices	7 datasets, $\approx 17 \cdot 10^9$ objects, ≈ 1.5 TB
Combinatorial statistics and maps	≈ 1.500 objects
SageMath databases	12 datasets
L -functions and modular forms	≈ 80 datasets, $\approx 10^9$ objects, ≈ 1 TB
Small Groups Library	$\approx 4.5 \cdot 10^8$ groups, ≈ 80 MB
Linked Data	
zbMATH	$\approx 4M$ publication records with semantic data, $\approx 30M$ reference data, $> 1M$ disambig. authors, $\approx 2,7M$ 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	$\approx 1,1M$ questions/answers, $\geq 11K$ answer authors
Stacks project	≥ 6000 pages, semantically annotated, curated, searchable textbook
nLab	$\geq 13K$ pages on category theory and applications

Figure 1: Summary of mathematical datasets

see [Bercic:cmo:wiki] for an overview of mathematical datasets and their “FAIR-readiness”.

Moreover, the inherent complexity of mathematical data makes them difficult to share in practice: even freely accessible datasets are often 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 their mathematical meaning. Therefore, unfortunately FAIR mathematics essentially does not exist today.

Motivation Our ultimate goal is to standardize a framework for representing mathematical datasets FAIRly. 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. This includes collaborations across disciplines and with industry practitioners, who are currently excluded due to the difficulty of understanding the datasets,
- *rewarding mathematicians for sharing datasets* (which is currently often not the case), e.g., by ma-

king 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, where even papers 50 years old are still routinely cited).

Contribution and Related Work In this article we survey and systematize how mathematical data are represented and shared, and we analyze how these solutions enable or prevent 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 the FAIR principles.

We are not aware of any similar analysis for mathematical data. The FAIR principles are mostly unknown in mathematics and are not systematically considered in every-day mathematical practice, which motivated our analysis in the first place. In fact, we are only aware of one mathematical dataset on the European Open Science Cloud hosting portal EUDAT. By contrast, FAIR sharing of datasets is more common in other disciplines. Independently, many fields have developed standardized, often XML-based formats that enable or simplify FAIR sharing such as SBML [SBML:on] for Systems Biology or Modelica [Modelica:on] for modeling complex systems. Due to its inherent diversity, several different such languages exist for mathematical data but none is close to being a dominant standard, and we discuss these as a part of our systematization.

Moreover, FAIR sharing in many fields has benefited tremendously from the use of linked data and standardized ontologies such as the biological and biomedical ontologies maintained by the OBO Foundry [OBO:on]. Moreover, as a first step towards a universal framework, and as a concrete example of FAIR-enabling mathematics-specific infrastructures, we introduce **MathDataHub**. This is a platform for sharing tabular mathematical datasets in a way that systematically enables FAIRness.

Acknowledgments The authors were supported by DFG grant RA-18723-1 OAF and EU grant Horizon 2020 ERI 676541 OpenDreamKit. Most of the implementation of the current **MathDataHub** prototype is due to Tom Wiesing.

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 characteristic issues, and in Section 4 we present a prototypical system that can help achieve them for the case of tabular data. Section 5 concludes the article.

2 FAIRness in Mathematics

2.1 General Considerations

The FAIR principles as laid out in, e.g., [WilDumAal:FAIR16] 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 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 [WilDumAal:FAIR16].

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.

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 [BilTen:fingerprint13]. For example, while it is good that a catalogue of graphs has a globally unique and persistent identifier, it is much better if in addition to that, every graph in the catalogue also has one. This also extends to other FAIR principles.

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.

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 good for tool-associated data, but may be practically impossible across tools.

Reusable Mathematical datasets are typically not reusable or 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. For instance, if we search for ‘Mathematics’ on *Google Data Search* [GDS:on], you get ca. 230 hits. But instead of data sets of mathematical objects the hits comprise *a)* PDF papers, figures, and slide collections, *b)* fiscal data on math-related institutions, *c)* data (i.e. PDFs of tables) on enrolment in maths curricula, *d)* data (PDFs again) on math achievements of students, *e)* data (real data sets) on psychological

experiments about mathematical skills (these are relatively rare), *f*) math journals, information systems, and publication venues.

Thus, instead of going by indexes or catalogs, 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:on; hog]), 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 difficult.

For example, consider an engineer who wants to prevent an electrical system from overheating and thus needs a tight upper estimate for the energy 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 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

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

To truly make mathematical data (here the statement of Hölder's inequality) findable, it must be in a form where a mathematical search engine like MathWebSearch [MWS-git:on] can find it from a query

$$\int_{\boxed{a}}^{\boxed{b}} |V(t)I(t)|dt \leq \boxed{R}$$

the boxed identifiers are query variables – and can even extend the calculation to

$$\int_a^b |V(t)I(t)|dt \leq \left(\int_a^b |V(x)|^2 dx \right)^{\frac{1}{2}} \left(\int_a^b |I(x)|^2 dx \right)^{\frac{1}{2}}$$

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, $0^\circ C$ in Europe is “called” $271.3^\circ K$ 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

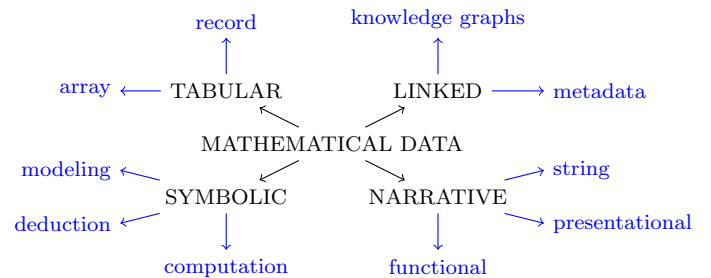


Figure 2: Kinds of mathematical data

To better analyze the current state of the art of FAIRness in mathematical data, we introduce a novel categorization of mathematical data below. An overview is given in Figure 2. As we will see, each kind of data makes different abstractions or focuses on different aspects of mathematical reality, resulting in characteristic strengths and weaknesses. We preview these in Figure 3.

Symbolic data consists of formal expressions such as formulas, formal proofs, programs, etc. These are written in a variety of highly-structured formal languages.

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

These languages are specifically designed for individual domains and allow for the development of associated *tools that understand the semantics* of the expressions. The most important such domains are **modeling**, **deduction**, and **computation** employing modeling languages, logics, and programming languages respectively. The associated tools like simulation tools, proof assistants, and computer algebra systems have access to 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, this comes at the price of being *difficult to produce*: the formalization of a typical narrative theorem as a statement in a proof assistant or a function in a computer algebra system can be prohibitively expensive even for experts. Moreover, the languages are context-sensitive so that expressions cannot be easily moved across environments, which makes *Finding*, *Reusing*, and *Interoperability* difficult.

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 [CarlisleEd:MathML3:on] and OMDoc/MMT [uniformal:on].

Tabular data employ representation theorems that allow encoding mathematical objects as ground data built from numbers, strings, tuples, lists, etc. For instance an elliptic curve can be fully represented by four integers: the coefficients in its minimal Weierstraß equation. Thus, tabular data combine optimized storage and processing with *capturing the complete semantics of the objects in a machine-understandable way*. It is also *easy to produce* and curate as general purpose database technologies and interchange formats such as SQL, CSV or JSON are readily available.

Tabular 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 large, multidimensional arrays stored in optimized array databases.

However, 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 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.

Interoperability is difficult because users need to know the exact representation theorem and the exact way how it is applied to understand the encoding. Even if the representation function is documented, *Finding*, *Reuse*, and *Interoperability* are theoretically difficult, practically expensive, and error-prone. For example, consider the following recent incident from (Jan. 2019): There are two encoding formats for directed graphs, both called **digraph6**: Brendan McKay’s [McKayFormats:on] and the one used by the GAP package Digraphs [GAPDigraphFormat:on], whose authors were unaware of McKay’s format and essentially reinvented a similar one [digraph6issue:on]. The resulting problem has since been resolved but not without causing some misunderstandings first.

Linked data introduce identifiers for objects and then treat objects 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 makes it *easy to produce linked data for any mathematical object at the price of not representing the complete 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, *machine-understandable* linked data formats are well-standardized: datasets 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 [wikidata:on] collects such linked data and uses them to answer queries about the objects.

Thus, contrary to the two categories discussed above, linked data has 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 consist of mathematical documents and text fragments, typically written by and for humans. We speak of **mathematical vernacular** for the mixture of mathematical formula, natural language with special idioms, and diagrams. Because of its free form, this vernacular makes it *easy to produce narrative data that represents the complete semantics of any mathematical object*. But this comes at the price that the semantics of the objects is *not machine-understandable*. Instead, machine-processing is limited to simple manipulations of the syntax of the objects. Based on the kind of limitation, we can distinguish four levels of increasingly machine-understandable narrative data:

0. **image/digitized**: scanned into images from paper-based documents; allows no machine-processing except for communication and archival as a whole,
1. **string**: written in languages such as L^AT_EX or OCR-ed from images; allows accessing fragments, e.g., for text-based search, or compilation into static media like PDF,
2. **presentation**: represented in presentation-oriented markup languages such as HTML and presentation MathML; allows flexible visual or aural rendering on interactive media such as web browsers,
3. **function**: represented in a form that makes explicit the functional structure and the relations between objects, their semantics, and the mathematical context; allows for some semantics-aware knowledge management and reasoning such as search up to equalities.

The last level of this classification transitions into the fully formalized documents of symbolic data where the entire semantics is machine-understandable; but because these abstract from the narrative form, we do not count them as narrative 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 C_k^n for the binomial coefficients or verbalizations in different languages. On the contrary, only limited machine support exists for the converse transformation: each step from a lower to a higher level as well as the transformation from functional markup to fully formalized mathematical knowledge is expensive, non-canonical, and often ambiguous and often requires expert knowledge about the mathematical background.

3 Deep FAIRness

Tabular 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 tabular 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 simple semantics, this can be remedied by attaching informal labels (e.g., column heads for tabular data), metadata, or free-text documentation. But this is not sufficient for datasets in mathematics and related scientific disciplines where the semantics is itself 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.

We speak of **accessible semantics** if data have metadata annotations that allow recovering the exact semantics of the individual entries of a dataset. Notably, in mathematics, semantics metadata are complex, usually symbolic data that cannot be easily annotated ad hoc. Without knowing the semantics, mathematical datasets only allow FAIR services that operate on the dataset as a whole, which we call **shallow** FAIR services. 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. For example, while a shallow accessibility service mediates access to entire datasets, a deep one allows accessing a specific entry or set of entries; naturally that requires deep identifiers, e.g., a DOI for each entry rather than just for the whole dataset. 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 may require it, e.g., deep finding requires checking each entry against the search criteria, which may require evaluating a semantic property. Moreover, services like reuse

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
Tabular	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

and interoperability are impossible without it: neither a human nor a system can reuse a dataset without being able to access and understand the semantic of each entry.

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. In Figure 5, we estimate the difficulty for the different combinations of services and kinds of data. For example, FAIR services for linked data are easy but limited. Services for tabular data are also relative easy *iff* there is accessible semantics, i.e., if the encoding function is known. FAIR services for symbolic and narrative data are much harder, mostly because the data and its semantics are much more complex.

Note that deep FAIR services are particularly desirable in mathematics, their advantages are by no means limited to mathematics. For example, in 2016 [ZieEreELO:GeneErrors16], 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 of Deep FAIR services for mathematics, we extend the original FAIR requirements from [WilDumAal:FAIR16], which focused on shallow FAIR, to deep FAIR, by applying them to dataset fragments (parts or individual entries).

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

DA Each dataset includes a definition of fragments and their function in the dataset. Accessibility protocols make accessible individual fragments, as well as their metadata and semantics.

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

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

4 MathDataHub: Towards Deep FAIR Hosting of Mathematical Data

To show the consistency of the Deep FAIR principles postulated above at least for one math data category, we present a unified infrastructure to support Deep FAIR for tabular 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 services, in particular an interface for Deep FAIR access of data sets (see Figure 6).

To that end, we developed a mathematical data description language MDDL in [BerKohRab:tumdi19] (Math Data Description Language) that uses symbolic data to specify the semantics of tabular data. MDDL schemata combine the low-level schemata of the underlying relational database with high-level descriptions (which critically use symbolic mathematical data) of the mathematical types of the data in the tables.

A census of small connected cubic vertex-transitive graphs

All connected cubic vertex-transitive graphs of order at most **1280**.





This dataset has 111360 objects.

Matches found: 482



[More about this dataset](#)

Display results

Available conditions

+ Order 	Number of vertices in the graph.
CVT Index 	
Graph 	
Name 	

Active conditions

Is Arc-Transitive 
Triangles Count < 50 









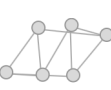
	Order 	Graph 	Name 	Is Arc-Transitive 	Triangles Count 
	4		Tetrahedron	true	4
	6		3-Prism	false	2

Figure 6: Website for the CVT census in MathDataHub

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 fortify our intuition with a concrete example let us consider the census of Cubic vertex-transitive graphs on up to 1280 vertices (CVT) [PotSpiVer:CVT13]. This census is a complete listing of a special class of symmetric graphs with at most 1280 vertices. A richer dataset containing several graph-theoretic invariants for each of the graphs was imported into MathDataHub [DMH:cvt].

Figure 7 shows the initial section of the MDDL theory [DMH:cvt-schema] describing the dataset and connects the dataset structure to the MitM ontology (see below).

For example, the mathematical type of the field `graph` is a finite undirected graph (or `finite_ugraph` in MitM). The specification is imported from `MitM :?Graphs?finite_ugraph` (not shown in Figure 7). The `codec` annotation specifies how this mathematical type is encoded as a low-level database type (in this case as a string format called `sparse6`). 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 `graph`, `order`, and others – as well as a database browser-like website frontend (see Figure 6). Concept definitions in the interface

```
namespace http://data.mathhub.info/schemas |
import MitM http://mathhub.info/MitM/smgom/graphs |

theory CubicVertexTransitiveGraphsS : ?MDDL =
  link /?Metadata?implements MitM:?Graphs?finite_ugraph |
  meta ?MDDL?schemaGroup "CVT" |

  order: int |
    meta ?Codecs?codec StandardInt |
    link /?Metadata?implements MitM:?Graphs?order |
    @_description "Number of vertices in the graph." |
    |

  graph: MitM:?Graphs?finite_ugraph |
    meta ?Codecs?codec GraphAsSparse6 |
    tag ?MDDL?opaque |
    @_description "The graph encoded in the sparse6 format." |
    |
```

Figure 7: CVT Census: Schema Theory Fragment

come from the background theory. In Figure 6, such a definition is shown for the order of the graph. The generation of APIs for computational software such as computer algebra systems is also possible and currently under development.

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 7, the mathematical background knowledge is imported from a theory `Graphs` in the **Math In The Middle ontology** (MitM) [MitM:on], which supplies the full mathematical specification and thus the basis for *Interoperability* and *Reusability*; see [BerKohRab:tumdi19; WieKohRab:vtuimkb17; KohMuePfe:kbimss17]

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 tabular mathematical datasets. These can and should eventually be linked to representation standards in other domains. For mathematical datasets, the math-specific aspects attacked by our work are the dominant factor. We summarize the state of deep FAIRness of MathDataHub in Figure 8.

DF	+	realized by codecs in the architecture and by filters in the generated front-end
	–	dataset search not exposed to general search engines; automatically generated unique identifiers for individual entries are not yet exposed in the interface (imported ones are)
DA	+	MathDataHub makes datasets accessible by offering easy hosting
	–	fragments only at the level of individual entries
DI	+	realized by the infrastructure for formalizing mathematical properties and the practice of using the shared MitM ontology
DR	+	realized by the same infrastructure realized in DI
	–	system support for licensing and provenance is planned, but still rudimentary

Figure 8: State of deep FAIR in MathDataHub

For other datasets in MathDataHub, see [data.mathhub:on]. In a few months since the prototype went online in August 2019, we imported six datasets into MathDataHub, adding up to about 10^9 cells with mathematical content. The initial responses from the mathematical community are positive and it appears that the system is filling a real gap.

In some cases, we only import initial sections of datasets (at least for now). This is backed up by two factors. On one hand, it improves the efficiency of the web interface, resulting in a better user experience. On the other hand, we expect that researchers will switch from a web interface to specialized tools when working with large and complicated datasets. One such example is the Small Groups Library. While it is valuable to have the initial section available for browsing online, the computer algebra system GAP is much better suited for serious exploration and computation.

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 difficult, much more so than for other disciplines. This is because mathematical data are 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 tabular 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. Collecting them sustainably e.g. in a system like MathDataHub is only viable for Deep FAIR datasets.

References



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 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 is a senior researcher at the Universities of Erlangen-Nürnberg. 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