Math Literate Knowledge Management via Induced Material

Mihnea Iancu, Michael Kohlhase

Computer Science, Jacobs University, Bremen, Germany
`initial.last@jacobs-university.de`

Abstract. Mathematicians integrate acquired knowledge into a mental model. For trained mathematicians, the mental model seems to include not just the bare facts, but various induced forms of knowledge, and the amount of this and the ability to perform all reasoning and knowledge operations can be seen as a measure of mathematical training and literacy. Current MKM systems only act on the bare facts given to them; we contend that they – their users actually – would profit from a good dose of mathematical literacy so that they can better complement the abilities of human mathematicians and thus, enhance their productivity. In this paper we discuss how we can model induced knowledge naturally in highly modular, theory-graph based, mathematical libraries and establish how to access it to make it available for applications, creating a form of mathematical literacy. As an example we present the FLATSEARCH extension of MATHWEBSEARCH system and show this access method in action.

1 Introduction

There is an interesting duality between the forms and extents of mathematical knowledge that is verbally expressed (published in articles, scribbled on blackboards, or presented in talks/discussions) and the forms that are needed to successfully extend mathematical knowledge and/or apply it. To "do mathematics", we need to extract the relevant knowledge structures from documents and reconcile them with the context of our existing knowledge – recognizing parts as already known and identifying those that are new to us. In this process we may abstract from syntactic differences, chain together known and acquired facts, and even employ interpretations via non-trivial mappings as long as they are meaning-preserving. We will call the ability to do all of this relatively effortlessly **mathematical literacy** as it is a prerequisite for doing mathematics effectively. Mathematical literacy is a distinguishing characteristic of a trained mathematician.

Current MKM systems are essentially illiterate mathematically as they only act on the bare facts given to them; this may be one of the reasons why they are not routinely used to support mathematics: mathematicians expect math literacy in their discussion partners. For a query of "binomial coefficient" a math-literate search engine would also find formulae of the form $\binom{3}{5!}$ and $C(n,k)$, instances of the formula $\frac{n!}{(n-k)!}$, and even "choice without repetition" or "Pascal's triangle".

A math-literate proof checker would try to use recognize an idempotent monoid as Abelian and extend its repertoire of applicable theorems accordingly. And finally, a math-literate eLearning system would pose the same exercises, but generate different explanations for students who know groups as axiomatized via an associative composition operation $\circ$ that admits units and inverses and defined a division operation $x/y := x \circ y^{-1}$ and students to whom groups were introduced by axioms for a division operation and composition, unit, and inverses were defined from them. As these examples show, mathematical literacy would make the interaction with MKM systems more natural and effective.

We contend that a large part of mathematical literacy is a function of having at our disposal – or being able to generate on demand a large space of knowledge that is induced in some way by the explicitly represented knowledge we have acquired previously – we call it the **Mathematical Knowledge Space** (MKS).

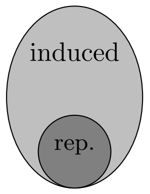


Fig. 1. MKS

In this paper, we will show two ways of how knowledge items can systematically be induced from existing representations to arrive at more mathematically literate services. We regard them as initial case studies that show what math literate MKM could look like; more case studies are certainly needed. The first is based on the mathematical practice of viewing an object of class A as one of class B – which we call **framing**. Following [KK09] we model framing via theory morphisms in modular theory graphs – which act as the MKS – and extend our MathWebSearch engine [KMP12] so that it answer queries “modulo framing”. The second case study takes up the notion of realms that structure theory graphs in a more human-oriented fashion. The various user roles identified in [CFK14] allow us to induce special versions of the underlying theory graph for different roles. Indeed, we show that the realm faces – which were assumed to be hand-curated in [CFK14] – can be induced from the developments in the realms.

We will introduce the two case studies the next two sections: Section 2 interprets the knowledge space as a Mmt theory graph and the induced statements are computed by flattening. The realms case study is presented in Section 4, where we discuss how realm faces (induced theories) can be generated and pillars can be opened for inspection from the faces. Section 5 shows an application of induced material: we can use flattening to make a math search engine literate. Section 6 concludes the paper.

2 Induced Statements in Theory Graphs

In order to build math literate MKM services as defined in section 1 we need to first address the issues of generating (part of) the mathematical knowledge space and then accessing the induced knowledge in order to make it available for MKM applications.

We use the (theory-graph enabled) Mmt language and system as a basis of discussion and we briefly introduce it below.

MMT [RK13] is a generic, formal module system for mathematical knowledge. The MMT language is designed to be applicable to a large collection of declarative formal base languages and all MMT notions are fully abstract in the choice of the base language.

We will only give a brief introduction to MMT here and then discuss the concepts using examples in the following sections. We refer to [RK13] for further details.

The central notion is that of a **theory graph** containing *theories* and *views* (morphisms between theories). Note that MMT libraries actually contain (possibly nested) collections of documents which in turn contain modules but we omit this here for simplicity and focus on theory graphs instead.

Theories S are formed from a set of typed symbols and axioms describing their properties. Views $\mathsf{v} : \mathsf{S} \to \mathsf{T}$ are morphisms between theories which map axioms of the source theory (S) to theorems of the target theory (T). This property ensures that all theorems of the source theory induce theorems of the target theory and induces a homomorphic translation from S-terms to T-terms by replacing every occurrence of an S-axiom with its corresponding T-theorem.

In addition to views, the module level structure in MMT theory graphs is given by theory inheritance. The most general kind of inheritance in MMT is represented by *structures* which are (possibly) partial named imports (and defined using theory morphisms). *Includes* are trivial structures which are unnamed and total while *metas* are distinguished includes representing the meta-theory (each theory can have at most one meta-theory).

Every MMT declaration is identified by a canonical, globally unique URI. Theories and views can be referenced relative to the URI U of the theory graph that contains them by $\mathsf{U}?\langle\langle$theory-name$\rangle\rangle$ and $\mathsf{U}?\langle\langle$view-name$\rangle\rangle$, respectively.

Symbol declarations can be referenced relative to the URI of their containing theory T by $\mathsf{T}?\langle\langle$symbol-name$\rangle\rangle$. Similarly, assignment declarations can be referenced relative to the URI of their containing view v by $\mathsf{v}?\langle\langle$symbol-name$\rangle\rangle$.

Note that the names of symbols, theories and views can have multiple /-separated fragments and are of the general form $\mathsf{f}_1/\ldots/\mathsf{f}_n$. This makes MMT URIs much more expressive and, in particular, allows the following additional access methods:

– if theory T contains structure s_1, s_1 contains structure s_2, ..., and s_n contains symbol const then we can use the symbol name $\mathsf{s}_1/\ldots/\mathsf{s}_n/\mathsf{const}$, to refer to const (as translated over the assignments from the structures) from T.
– if there is a view $\mathsf{v}_1 : \mathsf{T}_1 \to \mathsf{T}$ and a view $\mathsf{v}_2 : \mathsf{T}_2 \to \mathsf{T}_1$, ..., and a view $\mathsf{v}_n : \mathsf{T}_n \to \mathsf{T}_{n-1}$ where T_n contains symbol const then we can use the symbol name $[\mathsf{U}?\mathsf{v}_1]/\ldots/[\mathsf{U}?\mathsf{v}_n]/\mathsf{const}$, to refer to const (as translated over the assignments from the views) from T
– if T_1 is a nested module in T, ..., and T_n is a nested module in T_{n-1} we can use the theory name $\mathsf{T}/\mathsf{T}_1/\ldots/\mathsf{T}_n$ to refer to theory T_n.

The MMT system provides an API to the MMT data structures described above and the MMT implementation [Rab08; RK13] provides a Scala-based [OSV07] open source implementation of the MMT API.

Generating Induced Knowledge In the context of theory graphs we model the process of generating the knowledge space as an operation on theory graphs. Specifically, one that takes a theory graph G and return an enriched graph $\overline{G}$ where a new part of the mathematical knowledge space is explicitly represented. We call $\overline{G}$ the *induced theory graph*.

Accessing Induced Knowledge A key aspect of MMT is that it's URI language is expressive enough such that it allows producing URIs for the induced statements that are not only unique but also informative. Specifically, we can produce URIs that explain existence of each induced statement so that we can relate to the original theory graph. We call this property of URIs *information completeness*.

That allows applications to tie the induced theory graph to the original one and, for instance, produce human-understandable explanations of how each induced statement is produced.

3 Flattening Theory Graphs

To better understand the concept of framing in modular libraries, consider the theory graph in Figure 2. The right side of the graph introduces the elementary algebraic hierarchy building up algebraic structures step by step up to rings; the left side contains a construction of the integers. In this graph, the nodes are theories [1], the solid edges are imports and the dashed edges are views.

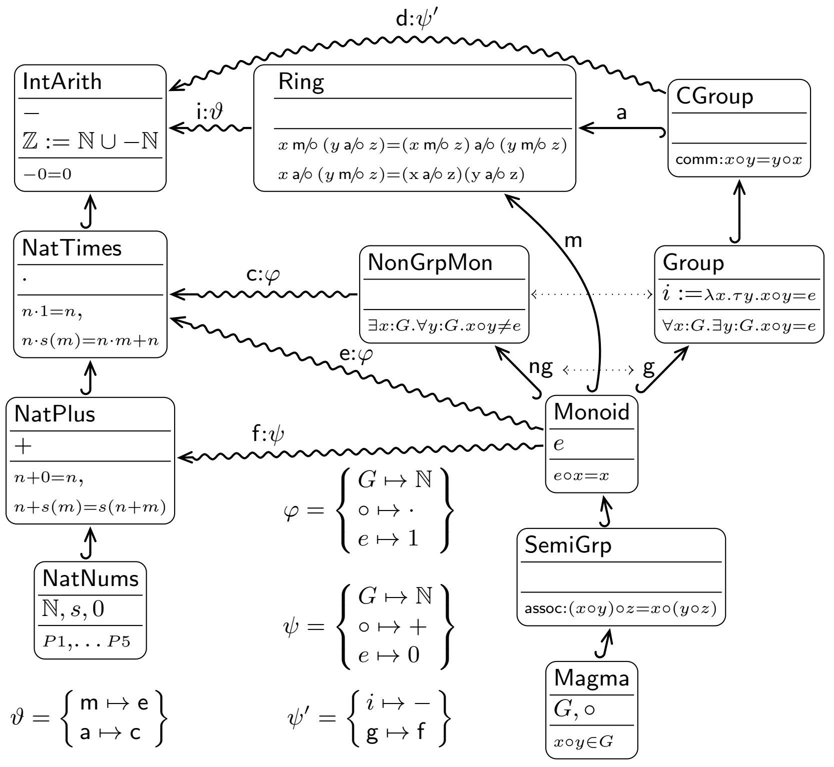
In MMT theory graphs, theory morphisms can carry a name, and inherited constants can be disambiguated via the inclusion that induced them. An application of this is in the definition of the ring theory, which inherits all of its operators (and their axioms) via the two inclusions m (for the multiplicative operations) and a (for the additive operations). To complete the ring we only need to add the two distributivity axioms in the inherited operators $\mathsf{m}/\circ$ and $\mathsf{a}/\circ$.

Furthermore, morphisms can carry an assignment which maps symbols and axioms from the source theory to terms in the target theory. We see this in the view e from Monoid to NatArith, which assigns $\mathbf{N}$ to the base set G, multiplication $(\cdot)$ to $\circ$ and the number 1 to the unit e. The document theory morphism property, e also contains proofs for all Monoid axioms in NatArith.

It is a special feature of MMT that assignments can also map morphisms into the source theory to morphisms into the target theory. We use this to specify the morphism c modularly (in particular, this allows to re-use the proofs from e and c).

Note that already in this small graph, there are a lot of induced statements. For instance, the associativity axiom is inherited in seven times (via inclusions; twice into Ring) and induced four times (via views; twice each into NatArith and IntArith). All in all, we have more than an hundred induced statements from the axioms alone. If we assume just 5 theorems proven per theory (a rather

[1] We have left out the quantifiers for the variables x, y, and z from the axioms to reduce visual complexity. The always range over the respective base set. Furthermore, all axioms are named; but we only state the names we actually use in the examples.



d:ψ'
IntArith
$-$
$\mathbb{Z} := \mathbb{N} \cup -\mathbb{N}$
$-0=0$
i:ϑ
Ring
x m/∘ (y a/∘ z)=(x m/∘ z) a/∘ (y m/∘ z)
x a/∘ (y m/∘ z)=(x a/∘ z)(y a/∘ z)
a
CGroup
comm:$x{\circ}y=y{\circ}x$
NatTimes
$\cdot$
$n{\cdot}1=n$,
$n{\cdot}s(m)=n{\cdot}m+n$
c:φ
NonGrpMon
$\exists x{:}G.\forall y{:}G.x{\circ}y\neq e$
m
Group
$i := \lambda x.\tau y.x{\circ}y=e$
$\forall x{:}G.\exists y{:}G.x{\circ}y=e$
e:φ
ng
g
NatPlus
$+$
$n+0=n$,
$n+s(m)=s(n+m)$
f:ψ
Monoid
e
$e{\circ}x=x$
$\varphi = \{G \mapsto \mathbb{N}, \circ \mapsto \cdot, e \mapsto 1\}$
SemiGrp
assoc:$(x{\circ}y){\circ}z=x{\circ}(y{\circ}z)$
NatNums
$\mathbb{N}, s, 0$
$P1, \dots P5$
$\psi = \{G \mapsto \mathbb{N}, \circ \mapsto +, e \mapsto 0\}$
Magma
$G, \circ$
$x{\circ}y{\in}G$
$\vartheta = \{$m $\mapsto$ e, a $\mapsto$ c$\}$
$\psi' = \{i \mapsto -$, g $\mapsto$ f$\}$

Fig. 2. A Mmt Graph for Elementary Algebra

conservative estimation), then we obtain a number of induced statements that is an order of magnitude higher.

Another crucial ingredient of Mmt for the endeavor of searching for induced statements is the fact that, as discussed in Section 2, Mmt URIs are expressive enough to supply names for all induced statements. In fact, we can already access the induced statements in Figure 2 in Mmt. For example, if we take u to be the URI of the theory graph, the statement $\forall x, y, z : \mathbb{Z}.(x + y) + z = x + (y + z)$ induced by the view c in IntArith has the Mmt URI u?IntArith?c/g/assoc. Still, for external applications, it is essential to have the induced statements explicitly represented.

Generating Induced Statements We already hinted in section 2 that generating the induced statements is a *theory graph operation* i.e. it takes a theory graph as input and returns a different one, specifically the induced theory graph.

Below, we define theory graph flattening as an as an instance of such a generation procedure that produces those statements induced by framing.

Following the algebraic structures example in Figure 2 we can formally define the process of flattening a theory (or a theory graph).

Definition 1. *Given a theory graph* G *the flattening of a theory* T *in* G *is a theory* $\overline{\mathsf{T}}$ *with the same URI as* T *containing:*

- *all symbol declarations that are in* T*.*
- *all symbol declarations that are imported into* T*.*
- *for every view* $\mathsf{v} : \mathsf{S} \to \mathsf{T}$ *the projection of every* S*-based declaration over view* v*. Here, by* S*-based declaration we refer to the declarations in* S *and in theories that import* S*.*

The URIs of the induced declarations are based on the definition of Mmt URIs from section 2 (see also the assoc example above) and permit recovering the origin of the induced declarations (i.e. are *information complete* as defined in 2). Specifically, symbol declarations from T or imported in T by the meta or an include preserve their name while symbol declarations imported in T by a structure or induced by a view are additionally qualified with the name of that structure or view.

Definition 2. *The flattening of theory graph* G *is a theory graph* $\overline{\gamma}$ *with the same URI as* G *containing the following module declarations :*

- *for every theory* T *in* G *the theory* $\overline{\mathsf{T}}$
- *every view* $\mathsf{v} : \mathsf{S} \to \mathsf{T}$ *in* G

Note that, since theory flattening preserves theory URIs and doesn't add new axioms (only new theorems) any view from S to T is also a view from $\overline{\mathsf{S}}$ to $\overline{\mathsf{T}}$ so the views in $\overline{\gamma}$ are still valid.

4 Inducing Realm Faces/Flattening Realms

In [CFK14] we introduce the concept of **realms** to consolidate knowledge about mathematical theories. This is motivated by the intuition that users of a knowledge collection can have different roles and therefore want to see different kinds of materials. In a nutshell, a realm – pictured schematically on the right – is super-structure of a fragment of a theory graph that singles out a set of conservative developments called **pillars** and extends them with a theory called the **face** of the realm that abstracts from all development details in the pillars, which are required to be linked by a chain of views that makes them isomorphic.

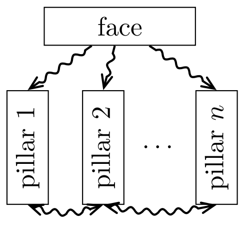


Fig. 3. Realm Schema

The idea of [CFK14] is that practitioners only need access to the face that supplies all the useful facts about a mathematical domain, whereas the student also wants to know how these are established and needs to access the "development history" in one (or more) pillar. The developer finally wants to develop the knowledge about the domain by extending one (or more) pillar, the development of the face is just regarded as a side effect.

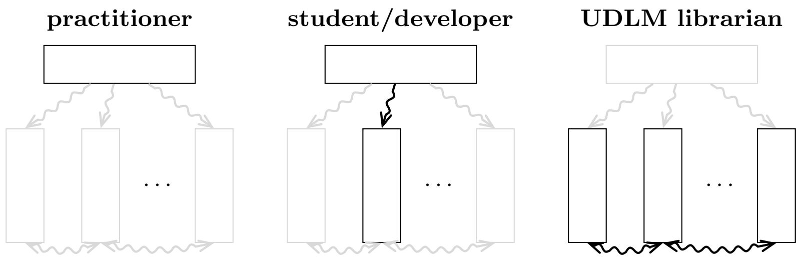


Fig. 4. Realms and User Roles

In Figure 4, we have refined the schematic of Figure 3 by graying out the parts of the realms the users in their respective roles will not be able to see. We can see the restricted theory/realm graphs as *induced material* adapted to particular users: *i*) the face graph – the faces inherit the graph structure of the developments – for the practitioners, *ii*) the graph of faces with selectively opened developments for the student *iii*) the developments with their faces for the developers, and *iv*) the original theory graph for the knowledge librarian – a "user" of the library who maintains the library, e.g. refactoring theories, renaming theorems to avoid name clashes, etc. For the developer and the librarian, the realm faces are secondary objects which have to be maintained without being the primary object of study. For them it would be very convenient to let them be computed from the developments automatically as induced material. Indeed this can be done, as we will show in the rest of this section.

4.1 Generating Realm Faces as Induced Theories

For realms we consider their face as inducible from the pillars and discuss in the following how that can be mechanized.

Firstly, if faces are induced, we can define a realm by just giving it a **name**, specifying the pillars, and listing the cycle of views that shows that the pillars are equivalent. Listing 6 on the right gives the general form of the specification.

```
realm r = {
  pillar $p_1 = \{t_1^{p_1}, t_2^{p_1}, \ldots, t_{k_1}^{p_1}\}$
  pillar $p_2 = \{t_1^{p_2}, t_2^{p_2}, \ldots, t_{k_2}^{p_2}\}$
  ⋮
  pillar $p_n = \{t_1^{p_n}, t_2^{p_n}, \ldots, t_{k_n}^{p_n}\}$
  equivcyc = $\{v_1, \ldots, v_n\}$}
```

Fig. 6. A Realm Specification

For the induced face, the URI g of the file with the realm specification induce the MMT URI of the face, and the MMT URIs of the symbols inside are induced from the pillar names. To induce the face we need to solve three main issues:

1. select which symbols from each pillar should be in the face
2. merge equivalent symbols (such as the e in Figure 5)
3. resolve naming conflicts (equivalent symbols with different names and distinguishable symbols with same name)

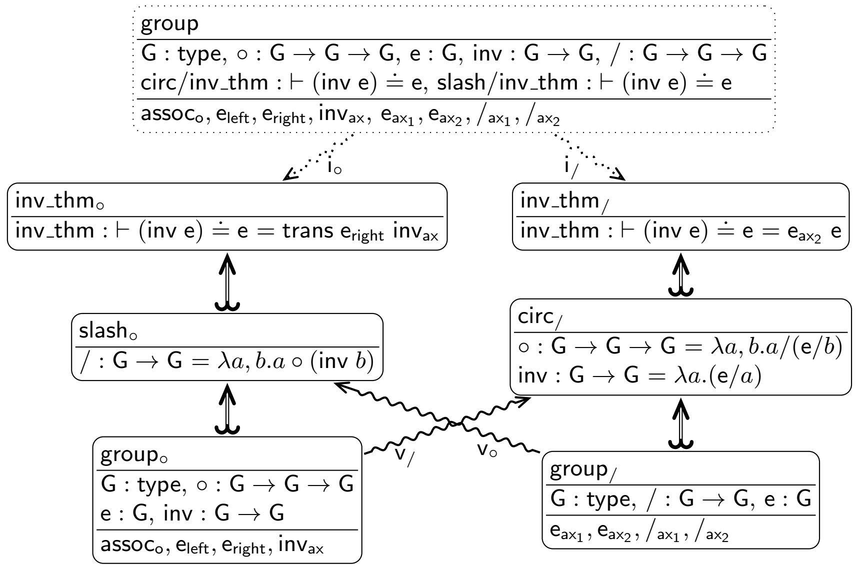

group
G : type, ∘ : G → G → G, e : G, inv : G → G, / : G → G → G
circ/inv_thm : ⊢ (inv e) ≐ e, slash/inv_thm : ⊢ (inv e) ≐ e
assoc∘, eleft, eright, invax, eax1, eax2, /ax1, /ax2
i∘
i/
inv_thm∘
inv_thm : ⊢ (inv e) ≐ e = trans eright invax
inv_thm/
inv_thm : ⊢ (inv e) ≐ e = eax2 e
slash∘
/ : G → G = λa, b.a ∘ (inv b)
circ/
∘ : G → G → G = λa, b.a/(e/b)
inv : G → G = λa.(e/a)
v/
v∘
group∘
G : type, ∘ : G → G → G
e : G, inv : G → G
assoc∘, eleft, eright, invax
group/
G : type, / : G → G, e : G
eax1, eax2, /ax1, /ax2

Fig. 5. A Realm of Groups

Before we formalize this in Definition 4 below, let us adapt the groups realm from [CFK14] to the MMT setting for intuitions.

Example 1. Figure 5 shows a realm with two pillars for the equivalent group definitions based on composition $\circ$ and, respectively, division $/$ in the usual way. The corresponding realm specification is in Figure 5.

Theories $\mathsf{slash}_\circ$ and $\mathsf{inv\_thm}_\circ$ as well as $\mathsf{circ}_/$ and $\mathsf{inv\_thm}_/$ are each conservative developments of $\mathsf{group}_\circ$ and $\mathsf{group}_/$ respectively as they only introduce defined symbols. The views $\mathsf{v}_\circ$ and $\mathsf{v}_/$ ensure the equivalence of the two pillars with the obvious assignments. The proofs that the axioms hold (i.e. the assignments for them in the views) are all straightforward and omitted for simplicity. The **face** for the realm is shown in theory group and has the intuitive shape, containing all the important concepts as primitive symbols and all their properties as axioms. We explain how the face was produced below.

```
realm group = {
  pillar circ = {group∘, slash∘}
  pillar slash = {group/, circ/}
  equivcyc = {v∘,v/}}
```

Fig. 7. A Group Realm Spec.

Definition 3. We define the following partial ordering on symbols in the realm. Let t and s be symbols in theories T and S respectively such T and S are in different pillars. If there is an assignment $\mathsf{s} := \mathsf{t}$ in one of the pillar equivalence views then we write $\mathsf{s} \geq_\mathsf{r} \mathsf{t}$. If there is also an assignment $\mathsf{t} := \mathsf{s}$ in such a view then we write. $\mathsf{s} =_\mathsf{r} \mathsf{t}$, otherwise $\mathsf{s} >_\mathsf{r} \mathsf{t}$. We call a symbol that is $>_\mathsf{r}$-maximal **essential** in r.

The intuition behind Definition 3 is that if there is an assignment $\mathsf{s} := \mathsf{t}$ then there is an equivalence between s and t at the realm level and we need to decide which should appear in the face. Then, the ordering captures the fact that s is a primitive concept in its realm while t is derived, possibly only to give the equivalence view. This is, for instance, the case of the symbols inv and $\circ$ from theory $\mathsf{circ}_/$ in Figure 5.

Definition 4. Let r be a realm as specified in Figure 6. Then, we generate a face for r by adding copies of all essential symbols (with type but no definition – following the definition of a realm face from [CFK14]) with the following provisions:
1. If two essential symbols in different pillars have the same name we prefix them with the pillar name (to ensure unique URIs)
2. If n essential symbols $\mathsf{s}_1, \ldots, \mathsf{s}_\mathsf{n}$ are equal (with respect to $=_\mathsf{r}$) we add the later ones as aliases of the first (order is irrelevant but must be consistent). An exception occurs when (some) of the equal symbols have the same name in which case we only add them once and effectively merge them (instead of prefixing the names with the pillar name as usual).

Example 2. The face group from Figure 5 is generated following Definition 4. The essential symbols (omitting axioms for simplicity) are $\mathsf{G}, \circ, \mathsf{e}, \mathsf{inv}, \mathsf{inv\_thm}$ for the first pillar and $\mathsf{G}, /, \mathsf{e}, \mathsf{inv\_thm}$ for the second. We have two name clashing pairs: $\mathsf{group}_\circ\mathsf{?e}$ and $\mathsf{group}_/\mathsf{?e}$ as well as $\mathsf{inv\_thm}_\circ\mathsf{?inv\_thm}$ and $\mathsf{inv\_thm}_\circ\mathsf{?inv\_thm}$. For the first pair ($\mathsf{e}$) we have an equality since $\mathsf{v}_\circ$ and $\mathsf{v}_/$ assign them to each other so we merge. Then, for the second pair ($\mathsf{inv\_thm}$) we prefix with the realm name producing $\mathsf{circ/inv\_thm}$ and $\mathsf{slash/inv\_thm}$ to obtain the face shown in Figure 5.

4.2 Curating Realms through Alignments

The problem with Definition 4 is that we can have duplicate symbols that are actually equivalent but appear as different because of slightly different formalizations. A common example is theorems with different proofs as is the case of $\mathsf{inv\_thm}$ in Example 2.

In [CFK14] realm faces are meant to be manually generated and curated to avoid such issues. However, we propose an alternative method of curating faces by giving *alignments* between pillar theories to establish symbols as being equivalent. This idea is inspired from [KRSC11], but has not been made formal before.

Definition 5 (Alignment). An **alignment** is a view pair $v_1 : \widehat{S} \to T$ and $v_2 : \widehat{T} \to S$, where $\widehat{S}$ is the **abstraction** of S: $\widehat{S}$ omits all definitions in S.

The abstraction operation $\widehat{\cdot}$ is needed to allow us to assign a new interpretation for defined symbols which would otherwise be translated via definition expansion. We will concentrate on the case where S and T are in different pillars here.

For instance take the situation in Figure 5 where the (trivial) theorem $\mathsf{inv\_thm}$ proving that e is its own inverse appears in each pillar. Still the proofs are different over the translation so that both symbols appear different at the MMT level. However, we can fix the problem by giving an alignment between the two theories containing the theorem. Listing 1.1 below shows the alignment and the resulting, curated face.

Listing 1.1. Alignment Example

$\mathsf{view}\ \mathsf{a}_/ : \widehat{\mathsf{inv\_thm}_\circ} \to \mathsf{inv\_thm}_/ = \{\mathsf{inv\_thm} := \mathsf{inv\_thm}\}$

$\mathsf{view}\ \mathsf{a}_\circ : \widehat{\mathsf{inv\_thm}_/} \to \mathsf{inv\_thm}_\circ =\{\mathsf{inv\_thm} := \mathsf{inv\_thm}\}$

$\mathsf{theory\ group} = \{\mathsf{G} : \mathsf{type},\ \mathsf{e} : \mathsf{G},\ \ldots,\ / : \mathsf{G}{\to}\mathsf{G}{\to}\mathsf{G}\ \ldots,\ \mathsf{inv\_thm} : \vdash (\mathsf{inv\ e}) \doteq \mathsf{e}\}$

4.3 Opening a Pillar

For the student/developer view described above we need the operation of *opening a pillar* that allows the developer to access the internals of the symbols and axioms in the face as formalized in one of the pillars. We model this by creating a new theory for each pillar that combines the symbol aggregation and name abstraction of the face theory with the implementation details of that pillar.

Concretely, given a realm r and a pillar p we induce a theory r/p that is generated following the same procedure as the face (i.e. from Definition 4) but without omitting the definitions. For symbols in a different pillar we generate the definition by translating it over the view from that pillar into p. Effectively, we obtain the symbol definitions *as seen from* p which corresponds to the intuition of opening a pillar.

In the theory graph, we represent this as 1. a structure s from the pillar p that adds its symbols to r/p but with the renamings used in the face generation and, 2. a view $\mathsf{v} : \mathsf{r} \to \mathsf{r/p}$ that formalizes the relation that r/p is an implementation for the face.

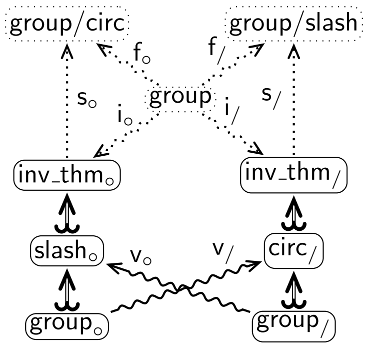


Fig. 8. Opening a Pillar

Listing 1.2 shows the theory group/circ representing the opening of pillar circ in the realm of groups above. It has the same symbols as the face but with all symbols that are non-primitive in circ having a definition. For the two symbols originating in the second pillar (/ and slash/inv_thm) their definition is obtained by translating over $\mathsf{v}_\circ$.

Listing 1.2. Developer View Example

$\mathsf{theory\ group/circ} = \{$

$\quad \mathsf{G} : \mathsf{type},\ \circ : \mathsf{G} \to \mathsf{G} \to \mathsf{G},\ \mathsf{e} : \mathsf{G},\ \mathsf{inv} : \mathsf{G} \to \mathsf{G},$

$\quad / : \mathsf{G} \to \mathsf{G} \to \mathsf{G} = \lambda \mathsf{a,b.a} \circ (\mathsf{inv\ b}),$

$\quad \mathsf{circ/inv\_thm} : \vdash (\mathsf{inv\ e}) \doteq \mathsf{e} = \mathsf{trans}\ \mathsf{e}_{\mathsf{right}}\ \mathsf{inv}_{\mathsf{ax}},$

$\quad \mathsf{slash/inv\_thm} : \vdash (\mathsf{inv\ e}) \doteq \mathsf{e} = \mathsf{e}_{\mathsf{ax}_2}\ \mathsf{e},$

$\quad \vdots$

}

The resulting theory graph is shown in 8 where the content of each theory is omitted. Note that the theories group, group/circ, group/slash as well as the structures $\mathsf{s}_\circ$ and $\mathsf{s}_/$ and the views $\mathsf{i}_\circ, \mathsf{i}_/, \mathsf{f}_\circ$ and $\mathsf{f}_/$ are all induced.

5 Searching the Knowledge Space of the LATIN library

As a first application of the concepts described in this paper we build a system for searching the knowledge space (i.e. flattened theory graph) of the highly modular LATIN library.

MathWebSearch For searching, we use our MathWebSearch system [KMP12], which is a content-oriented search engine for mathematical expressions. that indexes formula-URL pairs and provides a web interface querying the formula index via unification.

This can be used for:

Instance Search e.g. to find all instance of associativity we can issue the query $\forall x, y, z : \boxed{\mathrm{S}}.(x\boxed{\mathrm{op}}y)\boxed{\mathrm{op}}z = x\boxed{\mathrm{op}}(y\boxed{\mathrm{op}}z)$, where the $\boxed{\text{-}}$ are query variables that can be instantiated in the query. In the library from Figure 2 we would find the commutativity axiom SemiGrp/assoc, its directly inherited versions in Monoid, to Ring and in particular the version u?IntArith?c/g/assoc.

Applicable Theorem Search where universal variables in the index can be instantiated as well; this was introduced for a non-modular formal libraries in [Ian+13]. Here we could search for $3 + 4 = \boxed{\mathrm{R}}$ and find the induced statement u?IntArith?c/comm with the substitution $R \mapsto 4+3$, which allows the user to instantiate the query and obtain the equation $3 + 4 = 4 + 3$ together with the justification u?IntArith?c/comm that can directly be used in a proof.

The implementation of a web service that conducts such searches is very simple: instead of harvesting formulae from a formal digital library directly as in [Ian+13], we flatten the library first, and then harvest formulae. Conveniently, Mmt flattening gives the included constants and axioms local names that are syntactically identical to the respective symbol paths in Mmt URIs, so that the generation of Mmt URIs for the formula harvests is trivial. Moreover, we can leverage the Mmt URIs to determine the origin the induced formulae.

But first we need to replace the human-oriented search front-end of MWS, i.e. the input of search queries and the presentation of search results. The LATIN atlas is written in an extension of the TWELF encoding [RS09] of LF [HHP93], so it is natural to use an extension of LF notation with query variables for input. Therefore, we use the Mmt notation language and interpretation service described in [IR12] to transform LF-style input into Mmt objects and subsequently to MWS queries.

Induced statements in the LATIN *library* We implemented library flattening as described in section 3 in Mmt and applied it to the LATIN library. The flattening (once) of the LATIN library increases the number of declarations from 2310 to 58847 (a factor of 25.4) and the total size of the library from 123.9 MB to 1.8 GB (a factor of 14.8). As expected, the multiplication factor depends on the level of modularity of the library. For instance, the highly modular math sub-library containing mainly algebraic structures increases from 2.3 MB to 79 MB thus having a multiplication factor of 34.3, more than double the library average. The size of the MWS harvests also increases considerably, from 25.2 MB to 539.0 MB.

Explaining URIs of induced statements The presentation of the MMT URIs requires some work as well: while the MMT system can directly dereference the MMT URI and thus be used to present the induced statement, humans want a justification that is more understandable than a MMT URI. Fortunately, this can be generated from the MMT URI by a simple template-based algorithm. Let us consider the search result u?IntArith?c/g/assoc from the instantiation search above, where we take u to be `http://cds.omdoc.org/cds/elal`. The first step is to localize the result in the theory u?IntArith with the sentence

Induced statement $\forall x, y, z : \mathbb{Z}.(x + y) + z = x + (y + z)$ found in `http://cds.omdoc.org/cds/elal?IntArith` (subst, justification). (1)

Here the underlined fragments carries hyperlinks, the second pointing to the justification:

IntArith is a CGroup if we interpret $\circ$ as $+$ and G as $\mathbb{Z}$. (2)

which can be directly from the information associated to the morphism c in the MMT URI. Then we skip over g, since its assignment is trivial and generate the sentence.

CGroups are SemiGrps by construction (3)

and finally we ground the explanation by the sentence

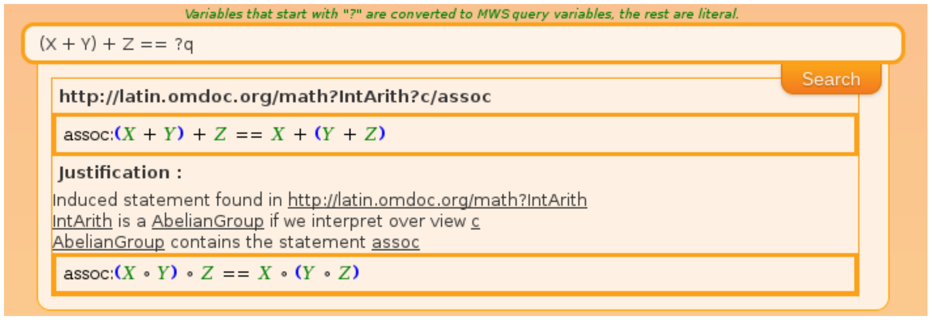
In SemiGrps we have the axiom assoc : $\forall x, y, z : G.(x \circ y) \circ z = x \circ (y \circ z)$ (4)

The sentences (1) to (4) can be generated from templates, since the MMT system gives access to the necessary information: source and target theory as well as the assignment ψ' for (2), the fact that the path from SemiGrp to CGroup only consists of inclusion that triggers the template for (3) and the original formulation of the axiom assoc.

The resulting search interface is shown in Figure 9. Note that we make use of another peculiarity of the MMT system in this explanation: all constants in the theory graph carry notation declarations [KMR08], which can be used to generate human-readable presentations of arbitrary formal objects in the graph.

6 Conclusion and Future Work

One of the characteristic abilities and practices of trained mathematicians is the ability to integrate new mathematical knowledge into their mental model, inter-


Variables that start with "?" are converted to MWS query variables, the rest are literal.

(X + Y) + Z == ?q

http://latin.omdoc.org/math?IntArith?c/assoc

assoc:$(X + Y) + Z == X + (Y + Z)$

Justification :

Induced statement found in http://latin.omdoc.org/math?IntArith
IntArith is a AbelianGroup if we interpret over view c
AbelianGroup contains the statement assoc

assoc:$(X \circ Y) \circ Z == X \circ (Y \circ Z)$

Fig. 9. The FlatSearch Web Interface for LATIN

pret it via non-trivial semantic mappings, and take a conceptual and deductive closure of the acquired knowledge in all the processes of "doing mathematics". Current MKM systems that want to support "doing mathematics" directly act on the represented mathematical knowledge they are fed with and therefore fall short of humans which make them less useful as tools and interaction counterpart.

The main hypothesis of the work presented here is the idea that running classical MKM algorithms on a suitably structured "mathematical knowledge space" (MKS) which extends the represented knowledge by a class of "induced knowledge items" will let them approximate mathematical literacy. We test this hypothesis on two classes of knowledge items in the context of theory graphs

In a first case study we extend a theory graph with statements induced by views in the theory graph of the formal LATIN library. Indexing this in the math-specific, but otherwise illiterate MathWebSearch engine turns it into the FlatSearch engine that gives us results that approximate mathematical literacy. In the second case study, we build on a realm-structured knowledge collections and turn it into a MKS by inducing realm faces and into a personal MKS by opening pillars as needed. Here the induced knowledge items are theories, structures, and views in the theory graph.

In both cases much of the heavy lifting has been done by special URIs, that serve as systematic identifiers of induced elements. In the case of FlatSearch, these URIs are the Mmt URIs already introduced in [RK13]. They are all we need to explain the results in terms of the original LATIN graph. In the realms case study we took great care to introduce new URIs for all induced knowledge items. It speaks for the strength and versatility of the Mmt design that the realm-based URIs can be interpreted and justified in the Mmt framework.

In the future, we plan to extend the "math literacy via induced knowledge structures" approach proposed in this paper with more facets and applications. We conjecture the availability of some form of systematic naming scheme that uses the structural parts of the original knowledge to name induced knowledge items and thus represents the induction path will be the crucial step in such extensions.

One extension that seems immediately profitable is to extend flattening and realms to flexiformal representations (representations of mathematical knowledge at flexible levels of formality; see [Koh13]) and apply it to traditional mathematical documents. [Lau07] revealed a theory graph of 51 theory nodes and 107 theory morphisms of which 12 were views, but 63 had non-trivial assignments in the first 35 pages of Bourbaki's Algebra. Applying FlatSearch to this graph would solve of the problems readers face with the Bourbaki books – which are otherwise well-liked for their structured approach: particular mathematical structures and objects can only be understood, if one already knows all the material they depend on. One author even said that

> *Bourbaki was a dinosaur, the head too far away from the tail. Explaining: [...] You could say "Dieudonné what is the result about so and so?" and he would go to the shelf and take down the book and open it to the right page. After Dieudonné retired no one was able to do this. So Bourbaki lost awareness of his own body* [Ric]

A flexiformalization of the Bourbaki books together with an extension of MMT that can deal with flattening of informal texts would go a long way to alleviate these problems.

Acknowledgments This work has been supported by the German Research Council (DFG) under grant KO 2428/13-1.

References

[CFK14] Jacques Carette, William Farmer, and Michael Kohlhase. "Realms: A Structure for Consolidating Knowledge about Mathematical Theories". In: *Intelligent Computer Mathematics 2014*. Ed. by Stephan Watt et al. Lecture Notes in Computer Science. MKM Best-Paper-Award. Springer, 2014, pp. 252–266. URL: `http://kwarc.info/kohlhase/submit/cicm14-realms.pdf`.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. "A framework for defining logics". In: *Journal of the Association for Computing Machinery* 40.1 (1993), pp. 143–184.

[Ian+13] M. Iancu et al. "The Mizar Mathematical Library in OMDoc: Translation and Applications". In: *Journal of Automated Reasoning* 50.2 (2013), pp. 191–202.

[IR12] M. Iancu and F. Rabe. "(Work-in-Progress) An MMT-Based User-Interface". In: *Workshop on User Interfaces for Theorem Provers*. Ed. by C. Kaliszyk and C. Lüth. 2012.

[KK09] Andrea Kohlhase and Michael Kohlhase. “Spreadsheet Interaction with Frames: Exploring a Mathematical Practice”. In: *MKM/Calculemus Proceedings*. Ed. by Jacques Carette et al. LNAI 5625. Springer Verlag, July 2009, pp. 341–356. URL: `http://kwarc.info/kohlhase/papers/mkm09-framing.pdf`.

[KMP12] Michael Kohlhase, Bogdan A. Matican, and Corneliu C. Prodescu. “MathWebSearch 0.5 – Scaling an Open Formula Search Engine”. In: *Intelligent Computer Mathematics*. Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 342–357. URL: `http://kwarc.info/kohlhase/papers/aisc12-mws.pdf`.

[KMR08] Michael Kohlhase, Christine Müller, and Florian Rabe. “Notations for Living Mathematical Documents”. In: *Intelligent Computer Mathematics*. Ed. by Serge Autexier et al. LNAI 5144. Springer Verlag, 2008, pp. 504–519. URL: `http://omdoc.org/pubs/mkm08-notations.pdf`.

[Koh13] Michael Kohlhase. “The Flexiformalist Manifesto”. In: *14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*. Ed. by Andrei Voronkov et al. Timisoara, Romania: IEEE Press, 2013, pp. 30–36. URL: `http://kwarc.info/kohlhase/papers/synasc13.pdf`.

[KRSC11] Michael Kohlhase, Florian Rabe, and Claudio Sacerdoti Coen. “A Foundational View on Integration Problems”. In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 107–122. URL: `http://kwarc.info/kohlhase/papers/cicm11-integration.pdf`.

[Lau07] Bastian Laubner. “Using Theory Graphs to Map Mathematics: A Case Study and a Prototype.” MA thesis. Bremen: Jacobs University, Aug. 2007. URL: `https://svn.eecs.jacobs-university.de/svn/eecs/archive/msc-2007/blaubner.pdf`.

[OSV07] M. Odersky, L. Spoon, and B. Venners. *Programming in Scala*. artima, 2007.

[Rab08] F. Rabe. *The MMT System*. see `https://svn.kwarc.info/repos/MMT/doc/html/index.html`. 2008.

[Ric] Émilie Richter. *Nicolas Bourbaki*. URL: `http://planetmath.org/NicolasBourbaki.html`.

[RK13] Florian Rabe and Michael Kohlhase. “A Scalable Module System”. In: *Information & Computation* 0.230 (2013), pp. 1–54. URL: `http://kwarc.info/frabe/Research/mmt.pdf`.

[RS09] F. Rabe and C. Schürmann. “A Practical Module System for LF”. In: *Proceedings of the Workshop on Logical Frameworks: Meta-Theory and Practice (LFMTP)*. Ed. by J. Cheney and A. Felty. ACM Press, 2009, pp. 40–48.