Realms: A Structure for Consolidating Knowledge about Mathematical Theories

Jacques Carette[1], William M. Farmer[1], and Michael Kohlhase[2]

[1] Computing and Software
McMaster University
http://www.cas.mcmaster.ca/~carette
http://imps.mcmaster.ca/wmfarmer
[2] Computer Science
Jacobs University Bremen
http://kwarc.info/kohlhase

Abstract. Since there are different ways of axiomatizing and developing a mathematical theory, knowledge about a such a theory may reside in many places and in many forms within a library of formalized mathematics. We introduce the notion of a *realm* as a structure for consolidating knowledge about a mathematical theory. A realm contains several axiomatizations of a theory that are separately developed. Views interconnect these developments and establish that the axiomatizations are equivalent in the sense of being mutually interpretable. A realm also contains an external interface that is convenient for users of the library who want to apply the concepts and facts of the theory without delving into the details of how the concepts and facts were developed. We illustrate the utility of realms through a series of examples. We also give an outline of the mechanisms that are needed to create and maintain realms.

1 Introduction

In [Far11] the second author calls for the establishment of a "universal digital library of mathematics" (UDLM). In our joint work it is understood that the UDLM will be organized as a theory graph, i.e., a set of theories (collections of symbol declarations, definitions, assertions, and their proofs) interconnected by meaning-preserving views (morphisms). The "little/tiny theories approach" approach first put forward in [FGT92] has been very fruitful for formal developments of mathematical knowledge, but it has not found its way into mainstream mathematics. One reason may be that there is an impedance mismatch with the way of mathematicians — the supposed users of the UDLM — think about and work with theories. [CF08] argue for the development of "high-level theories" that better mesh with these expectations. In this note we will re-examine the issues involved and propose a solution.

In the mathematical community the term "*theory*" is used to describe multiple ideas, from the axiomatic theory of the algebraic structure of a group to "Group Theory" as an entire discipline, and various gradations in between.

Looked at more closely, this implies a multi-scale organizational structure to the basic components of mathematics, ranging from individual concepts (e.g., a group) to whole sub-areas of mathematics (e.g., Group Theory). Here our interest in this structure is purely pragmatic: how can it be leveraged to build a better mechanized mathematics systems and, ultimately, a better UDLM.

We will consider this in a bottom-up manner: what is the most natural structure on *theories*[3] that allows us to abstract away from irrelevant details, yet still allow us to get some practical work done? One such structure is that of *mutual interpretability* between theories. Basically this is the case when we have two equivalent theories T_1 and T_2 (in a sense to be made precise later) with presentations that can be markedly different.

But why should *theory presentations* matter at all? Studies of "theories" in mathematics (e.g.„ Lawvere theories [Law04; LR11]) or in logic focus on entities that are *complete* in some sense. But such completeness generally also implies that the object at hand is effectively infinite, and thus cannot be directly represented in software. Hence we are immediately forced to work with *finite representations* of these infinite objects. Furthermore, by Gödel's incompleteness theorem, most of the interesting theories will be fundamentally incomplete, in that no finite representation will be able to adequately represent the complete whole. The relevance here is that we are forced to deal with (syntactic) representations, which will generally be *incomplete*. As this is forced on us, we need to gracefully adapt to dealing with theory presentations in place of the theories they represent.

2 The Setting: Theory Graphs

We will now present an abstract notion of a theory graph that is sufficient to introduce the notion of a realm without committing ourselves to a particular approach such as [CO12] or [RK13].

Let a **theory** be a presentation of an axiomatic theory consisting of a finite sequence of symbol and formula declarations. The symbols denote concepts and the formulas denote facts about these concepts. There are three kinds of formula declarations: **axioms**, **definitions**, and **theorems**.[4] We further assume that for a theory $T = [A_0, A_1, \ldots, A_n]$, for all i with $0 \leq i < n$, A_{i+1} is well formed in the context of $[A_0, A_1, \ldots, A_i]$. A theory thus represents an axiomatization of a mathematical topic. If T is a theory and A is a symbol or formula declaration, $T \vdash A$ **wf**, means that A is well formed in the context of theory T. When $T \vdash A$ **wf**, we define $T \ltimes A$ to mean $[A_0, A_1, \ldots, A_n, A]$; we also extend $\ltimes$ to

[3] Including biform theories [Far07].

[4] If we make use of the Curry-Howard isomorphism — as we do in [RK13] —, then we can get by with typed symbol declarations (with optional definitions) only. In the propositions-as-types paradigm, axioms are typed constant declarations, and theorems are typed definitions — the proofs correspond to the respective definiens of a symbol — which is of the respective type.

note.tex 125 2014-03-16 15:52:19Z kohlhase

apply to sequences of declarations (on the right). If T is a theory and φ is a formula, then $T \models \varphi$ means φ is a logical consequence of T. A theory is **primitive** if it contains only symbol declarations and axioms. A primitive theory represents a set of concepts and facts without a development structure. The **empty theory** is the empty sequence. When $T_1 = T \ltimes A$ and $T_2 = T \ltimes B$, we also define $T_1 \oplus T_2 := (T \ltimes A) \ltimes B$ whenever A and B are disjoint. By also extending $\oplus$ to sequences of declarations, we get a **join** operation on theory presentations.

A **theory graph** is a directed graph in which a node is a theory and we have edges from a theory T to a theory $T \ltimes A$. If T and T' are theories in a theory graph G, an edge from T to T' is designated as $T \xrightarrow{G} T'$. A theory graph is a modular representation of a formalized body of mathematical knowledge.

An **axiomatic development** of a theory T to a theory T' in G is a subgraph G' of G in which T is the only source of G' and T' is the only sink of G'. In this case, T and T' are called the **bottom theory** and the **top theory**, respectively. An axiomatic development is thus a lattice of theories in which the top theory is the join of the members of the lattice.

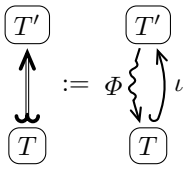
Let T_1 and T_2 be theories in G. A **view of** T_1 **in** T_2 is a homomorphic mapping Φ of the language of T_1 to the language of T_2 such that, for each formula φ of T_1, $T_1 \models \varphi$ implies $T_2 \models \Phi(\varphi)$. We denote a view by $\Phi : T_1 \rightsquigarrow T_2$. A view is thus a meaning preserving mapping that shows how T_1 can be embedded in T_2. It also provides a mapping from the models of T_2 to the models of T_1 (note the reversal of order). T_1 and T_2 are **equivalent** if there is a view in both directions. A view $\Phi : T_1 \rightsquigarrow T_2$ is **faithful** if for each formula φ of T_1, $T_2 \models \Phi(\varphi)$ implies $T_1 \models \varphi$. Views give a second (oriented, multi) graph structure on G, making it into a bigraph. It is important to note that the base theory graph is always acyclic.

Let T and T' be theories. T' is an **extension** of T, and T is a **subtheory** of T', if there exists a sequence S such that $T' = T \ltimes S$. In this case, there is a view $\Phi : T \rightsquigarrow T'$ such that Φ is the identity function. We call Φ the **inclusion** of T into T' and denote it by $T \hookrightarrow T'$. This corresponds to the extensions of [CO12], as well as the display maps of categorical type theory.

An **interface** for T is a view $\Phi : T' \rightsquigarrow T$ such that Φ is injective. T' and T are called, respectively, the **front** and **back** of the interface. Each subtheory of T can be a front of an interface for T. An interface is intended to be a convenient means for accessing (parts of) T. The front of a good interface includes a carefully selected set of symbols and formulas that denote orthogonal concepts and facts that can be easily combined to express the other concepts and facts of T. See section 5 for some concrete examples.

An extension T' of T is **conservative** if there is a view $\Phi : T' \rightsquigarrow T$ such that Φ is the identity function when restricted to T (i.e., $\Phi \circ \iota = \mathrm{Id}_T$ where ι is the inclusion of T in T'). If T' is a conservative extension of T, then for each formula φ of T, $T' \models \varphi$ implies $T \models \varphi$. Common examples of conservative extensions are extensions by symbol declarations, definitions, or theorem (with proofs). Obviously, if T' is a conservative extension of T, then T and T' are equivalent.

We abbreviate the two arrows in a conservative extension with a double inclusion arrow in theory graphs. A subgraph G' of a theory graph G is conservative if T' is a conservative extension of T for each edge $T \xrightarrow{G} T'$ in G'. A **conservative development** is an axiomatic development that is conservative. Note that all the theories in a conservative development are equivalent to each other. We will write a conservative development with bottom theory S and top theory T as $S \subseteq T$.



$\Phi : T_1 \rightsquigarrow T_2$ is **expansive** if there is a $\Psi : T_2 \rightsquigarrow T_2$ such that (1) the range of Ψ is the image of Φ and (2) Ψ is the identity on the image of Φ. That is, a view of T_1 in T_2 is expansive if, roughly speaking, T_2 is a conservative extension of the view's image. A view of T is **conservative** if it is faithful and expansive. A conservative view of T is a generalization of a conservative extension of T. Obviously, if there is a conservative view of T_1 in T_2, then T_1 and T_2 are mutually viewable.

3 Motivation: Developers, Students, and Practitioners

The user of a UDLM can play three different roles. As a *developer*, the user creates new mathematical knowledge or modifies existing mathematical knowledge in the library. As a *student*, the user studies the mathematical knowledge in the library. And, as a *practitioner*, the user applies the mathematical knowledge in the library to problems, both theoretical and practical. A user may play different roles at different times and may even sometimes combine roles.

A theory graph does not fully support all three of the user's roles. In fact, it lacks the structure that is necessary to satisfy the following requirements:

R1 There can be many *equivalent theories* in a theory graph that represent different axiomatizations of the *same mathematical topic*. As a result, concepts and facts about this mathematical topic, possibly expressed in different languages, may be widely distributed across a theory graph. The developer and the student would naturally want to have these *different axiomatic developments* and the *set of concepts and facts* that are produced by them in *one convenient place and in one convenient language*.

R2 Developers prefer developments that start with *minimal bottom theories* and are built as much as possible using *conservative extensions*. This approach minimizes the chance of *introducing inconsistencies* (which would render the developments pointless) and maximizes the *opportunities for reusing the development in other contexts*. While these two benefits may not be of primary concern for the student and the practitioner, such a careful development is usually easier to understand and produces concepts and facts that can be more reliably applied.

R3 The developer would like to *create a view from one theory to another in a convenient manner* by starting with a view of a minimal axiomatization of the theory and then building up the view as needed using conservative

extensions. Also, there is a desire to use *the most convenient axiomatization* amongst equivalent presentations.

R4 The *application of a mathematical fact* usually does not require an understanding of how concepts and facts were derived from first principles. Hence the parts of the theory graph which were needed by the developers may not be useful to the practitioner, and may well get in the way of the practitioner's work. The practitioner would naturally want the concept or fact to be *lifted out of this tangled development bramble.*

R5 Languages are introduced in a theory graph for the purpose of theory development. They may employ vocabulary that is inconvenient for particular applications. The practitioner would naturally like to have *vocabulary chosen for applications instead of development.*

In summary, the developer, the student, and the practitioner have different concerns that are not addressed by the structure of a theory graph. These different concerns lead us to propose putting some additional structure on a theory graph, a notion we call a "realm", to meet these five requirements.

4 Realms

In a nutshell, a realm identifies a subgraph of a development graph, equips it with a carefully chosen interface theory that abstracts from the development, and supplies the practitioner with the symbols and formulas she needs.

Definition 1. A **realm** R is a tuple $(G, F, \mathcal{C}, \mathcal{V}, \mathcal{I})$ where:

1. G is a theory graph.
2. F is a primitive theory in G called the **face** of the realm R.
3. $\mathcal{C}$ is a set $\{C_1, C_2, \ldots, C_n\}$ of conservative developments in G.
4. $\mathcal{V}$ is a set of views that establish that the bottom theories $\bot_1, \bot_2, \ldots, \bot_n$ of $C_1, C_2, \ldots, C_n$, respectively, are pairwise equivalent.
5. $\mathcal{I}$ is a set $\{I_1, I_2, \ldots, I_n\}$ of conservative interfaces such that F is the front of I_i and the top theory $\top_i$ of C_i is the back of I_i for each i with $1 \leq i \leq n$.

For each i we call $(\bot_i, C_i, \top_i)$ the i**-th pillar** of R and I_i its **interface**. Note that every pillar of a realm R forms a realm together with its interface and the face of R. We call realms with just one pillar **simple**, and otherwise **proper**.

Figure 1 shows the general situation. All the theories in the realm R are equivalent to each other since *i*) all the bottom theories are equivalent by the views in $\mathcal{V}$, *ii*) all the members of a conservative development are equivalent, and *iii*) the front and back of a conservative interface are equivalent.

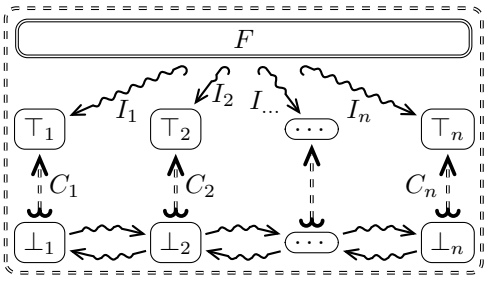


Fig. 1: The Architecture of a Realm

To insure that realms have a pleasant categorical structure (that

note.tex 125 2014-03-16 15:52:19Z kohlhase

of a contractible groupoid), we assume that $\mathcal{V}$ always contains identity views which show self-equivalence.

A realm consolidates a body of formalized mathematics pertaining to one topic. Each bottom theory $\bot_i$ is a different (ideally minimal) axiomatization and each conservative development C_i is a family of extensions of $\bot_i$. F is an (ideally convenient) presentation of the topic without any development structure and without any *scaffolding*, i.e., the concepts and facts that are needed only for development purposes. Finally, each interface I_i establishes that F is indeed a presentation of the topic and how it embeds into each $\top_i$.

The realm $R = (G, F, \mathcal{C}, \mathcal{V}, \mathcal{I})$ minus F and $\mathcal{I}$ records the development structure of the topic; we call $\overline{R} := (G, \mathcal{C}, \mathcal{V})$ the **body** of R. It can be used to study the development structure of the topic or as a basis for extensions. The face F exposes the most important and useful concepts and facts pertaining to that realm. It is also meant to be used as a module for constructing larger bodies of formalized mathematics. In other words, it can be seen as an export facility that only exports carefully selected symbols and formulas from the realm, without duplication or redundancy. Note that, in practice, we will choose for F the "usual symbols" traditionally used for that theory; these will also often correspond to the "original symbols" used in (some of) the bottom theories.

In particular, realms offer the infrastructure to satisfy the five requirements for users of a UDLM described in section 3. **R1** is addressed by the set of theories in the realm R and by the concepts and facts in F. **R2** is addressed by the conservative developments in $\mathcal{C}$. **R3** is addressed by the views in $\mathcal{V}$ and the conservative developments in $\mathcal{C}$. **R4** and **R5** are addressed by F being primitive and the fact that F is the front of an interface to each top theory.

Example 1 (Trivial realm). Any theory S in G induces a single-pillar realm $R = (G, S, \{G_S\}, \{\mathtt{Id}_S\}, \{\mathtt{Id}_S\})$ where G_S is the subgraph G consisting of S alone and $\mathtt{Id}_S$ is the identity view on S. Thus S serves as the top and bottom theories of the trivial (i.e., length 0) conservative development of S, as well as the face of R.

Example 2 (Initial realm). For any theory T in G we can extend G with a copy F_T of T and a conservative interface $F_T \overset{\iota}{\rightsquigarrow} T$, where ι maps any symbol to its copy to obtain a theory graph G'. We call $R_G^T := (G', F_T, \{G_T\}, \{\mathtt{ID}_S\}, \{\iota\})$, where G_T is the subgraph G consisting of T alone, the **initial realm** for T in G.

We can project any realm R to its face F, forgetting all developmental structure. Note that we can lift views between theories to **realm morphisms** (theory morphisms between their faces): given two realms R_1 and R_2 with two interfaces I_i, fronts F_i, and top theories $\top_i$, a view $\top_1 \overset{v}{\rightsquigarrow} \top_1$ induces a partial view $F_1 \overset{\widetilde{v}}{\rightsquigarrow} F_2$ on the faces, where $\widetilde{v} = I_2^{-1} \circ v \circ I_1$ (see Figure 2). In practice, the lifted views will almost always be total, since we prefer to use (in $\top_i$ and F_i) the original symbols from the bottom theories.

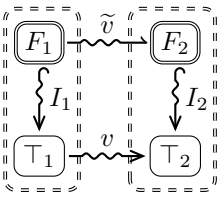


Fig. 2: Lifting

note.tex 125 2014-03-16 15:52:19Z kohlhase

5 Examples

As the development of the last few sections is fairly abstract, we will attempt to give the reader a better feel for realms through a selection of examples. We develop the first one in some detail and then give a more intuitive (and thus shorter) description of the remaining examples.

5.1 Groups

It is well-known that groups can alternatively be described in two ways:

> *Definition 1* [KM79]. A **group**$_1$ is a set G together with an associative binary operation $\circ\colon G \times G \to G$, such that there is a unit element e for $\circ$ in G, and all elements have inverses.
>
> *Definition 2* [Hal59]. A **group**$_2$ is a set G, together with a (not necessarily associative) binary operation $/\colon G \times G \to G$, such that $a/a = b/b$, $a/(b/b) = a$, $(a/a)/(b/c) = c/b$, and $(a/c)/(b/c) = a/b$ For all $a, b, c \in G$.

For any group$_1$ $(G, \circ)$, we can define a binary operation $\diagup_\circ$ by $a \diagup_\circ b := a \circ b^{-1}$ that shows that $(G, \diagup_\circ)$ is a group$_2$, and vice versa — using $a \circ_\diagup b := a/b_\diagup^{-1}$ with $b_\diagup^{-1} := (b/b)/b$. Practitioners want to use both group multiplication and division but are usually indifferent to how and where they are introduced.

In Figure 3, we have assembled this situation into a two-pillar realm (depicted by the double dashed box) with face group. The two bottom theories group_1 and group_2 are equivalent via the views v_1 and v_2 and the back views of c_1 and c_1, respectively.[5] Note that the views $v_1 = \circ \mapsto \circ_\diagup, e \mapsto e_\diagup, i \mapsto i_\diagup$ and $v_2 = / \mapsto \diagup_\circ$ carry proof obligations that show that the newly defined extensions slash_1 and $\mathsf{circ\text{-}i}_2$ behave as expected by the group_{3-i}. The face group contains "new" symbols, for which we use underlined symbols, to distinguish them from the ones in the pillars of the realm. The interface views I_i pick out the respective "original operators" $/$, $\circ$, and i, together with the corresponding axioms (and any theorems that may have been proven along the way). Here we have

$$I_1 = \underline{\circ} \mapsto \circ, \underline{e} \mapsto e, \underline{i} \mapsto i, \underline{/} \mapsto \diagup_\circ \quad \text{and} \quad I_2 = \underline{\circ} \mapsto \circ_\diagup, \underline{e} \mapsto e_\diagup, \underline{i} \mapsto i_\diagup, \underline{/} \mapsto /$$

In particular is is very natural to assume that the interfaces of a realm are conservative since they have access to all symbols in the body of the realm.

5.2 Natural Number Arithmetic

A realm $\mathbb{N}$ of natural number arithmetic would naturally contain conservative developments of several different axiomatizations of the natural numbers with

[5] This is a very common situation; the base theories differ mainly in which symbols are considered primitive, and the conservative developments mainly introduce definitions for the remaining ones.

note.tex 125 2014-03-16 15:52:19Z kohlhase

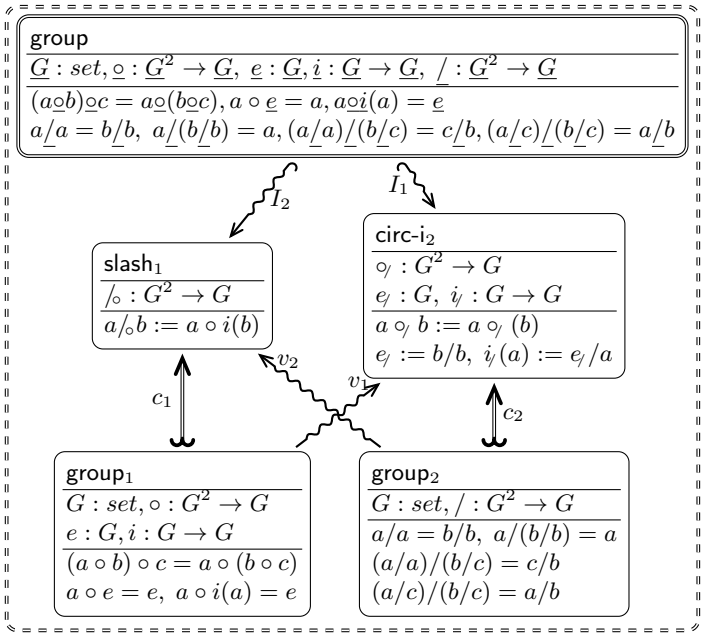


Fig. 3: A Realm of Groups with Face group

the usual arithmetic operations. One conservative development would certainly start with Peano's axiomatization of the natural numbers [Pea89]. The base theory would contain the symbols 0 and S (the successor function) and the (second-order) Peano axioms. The conservative development would include recursive definitions of $+$ (addition) and $*$ (multiplication). This development is particularly useful as it makes the proofs of many properties of the natural numbers simple.

Another kind of conservative development would start with a construction of the natural numbers using machinery available in the underlying logic. There are many such constructions. Some examples are finite von Neumann ordinals constructed from sets, Church numerals constructed from lambda expressions, strings of bits, and various bijective numeration schemes. These constructions define representations of the natural numbers that are semantically equivalent but far from equivalent with respect to computational complexity. It is worth singling out the *sequences of machine-sized words* representation, which tends to be the most efficient.

The face $F_{\mathbb{N}}$ of $\mathbb{N}$ would contain, as expected, the concepts and facts of natural number of arithmetic that are most needed by practitioners. These would naturally include symbols for all the natural numbers (i.e., natural number numerals) and all the true equations of the form $n_1 + n_2 = n_3$ and $n_1 * n_2 = n_3$. Thus $F_{\mathbb{N}}$ would contain an infinite number of symbols and facts. An implemen-

tation of $F_{\mathbb{N}}$ would require an efficient means to represent and compute with natural number numerals.

Since the mathematical theory of natural number arithmetic is exceedingly rich, there are a great many concepts and facts about natural numbers that could be of use to practitioners. For the average practitioner, the usefulness of $F_{\mathbb{N}}$ would be greatly reduced if there was an attempt to include all of these concepts and facts in $F_{\mathbb{N}}$. It would be much better to restrict $F_{\mathbb{N}}$ to the most basic concepts and facts about natural numbers and then derive the remaining concepts and facts from these. A good home for the derived results would be a conservative development that starts with $F_{\mathbb{N}}$ as its bottom theory.

5.3 Real Numbers

The theory of the real numbers covers the algebraic and topological structure of the real numbers. Exceedingly rich in concepts and facts, it is one of most important theories in all of mathematics. It is important to developers since real numbers are needed in most mathematical developments. It is important to students since its includes many of the most important ideas of mathematics. It is important to practitioners since most mathematical problems involve the real numbers in some way.

A realm $\mathbb{R}$ of the real numbers could be used to consolidate and organize all the knowledge about the real numbers that resides in a UDLM. $\mathbb{R}$ would have a structure similar to the realm of natural number arithmetic. It would contain two kinds of conservative developments. The first kind are axiomatizations of a complete ordered field. (All complete ordered fields are isomorphic.) The second kind are constructions of the real numbers, of which there are many. Some examples are Dedekind cuts in the field of rational numbers, Cauchy sequences of rational numbers, infinite decimal expansions, the quotient of the finite hyperrationals by the infinitesimal hyperrationals, and as a substructure of the surreal numbers. It is worth remarking that most of these constructions leverage $\mathbb{N}$, so that constructing realms is also a modular process.

Like the face $F_{\mathbb{N}}$ of $\mathbb{N}$, the face $F_{\mathbb{R}}$ of $\mathbb{R}$ should only contain the basic concepts and facts about the real numbers, with additional concepts and facts residing in a conservative extension $F_{\mathbb{R}}$. The prominent role of $\mathbb{R}$ means that $F_{\mathbb{R}}$, or one of its conservative extensions, would be the basis of many of the more sophisticated theories in a UDLM.

5.4 Monads

Category theorists and (advanced) Haskell programmers are familiar with the expressive power of monads. Most know that there are in fact two equivalent presentations of the theory of monads, one using a multiplication operation μ (called `join` in Haskell) and unit η (`return`), the other using *Kleisli triples* with a lifting operation $-^*$ (called `bind` or `>>=` in Haskell). From there, one can define a large list of generic combinators that work for any monad.

These two presentations are equivalent, and are again similar in flavor to the previous ones: one is more convenient for proofs, the other for computational purposes. Again, these basic theories tend to be followed by a substantial tower of *conservative extensions*. In other words, Haskell's `Control.Monad` should really be seen as the *face* of a theory of monads.

5.5 Modal Logic S4

The modal logic **S4** has a large number of equivalent presentations — John Halleck [Hal] lists 28 of them. This gives developers significant flexibility when using views (aka requirement **R3**) to establish that a structure can interpret **S4**. And, of course, **S4** supports rather significant conservative extensions and applications of it are found in a variety of places.

5.6 Models of Computation

The *Chomsky hierarchy* of regular, context-free, context-sensitive and recursively enumerable languages offer use names for (the face of) four more, nested, realms. As we know, each of the above languages contains many different equivalent formalisms which are nevertheless equivalent.

Inside the recursively enumerable languages (for example), we would have the pillars of Turing machines, Register machines, the Lambda Calculus, certain automata, etc, as alternatives. This particular realm poses extreme difficulty when it comes to the *design* of a suitable face theory.

6 The Realm Idea

The examples of the previous section show the advantages of realms as consolidated structures: a realm hides cumbersome details, while still allowing access to the details for those (such as developers) who must deal with them. Realms thus deal with two structural tensions in the design of theory graphs that formalize a mathematical domain:

Foundational Realms can in many ways be understood as the formalization of the ideas of *information hiding* and *modules* coming from software engineering. The face of a realm corresponds to an interface, its secrets, i.e., what it hides, is the actual conservative development of the theory, and its representation details correspond to an axiomatization. Of course, to get substitutivity, we need to ensure equivalence. In an ad hoc manner, Haskell's type classes, ML's modules and functors, Scala's traits, Isabelle's locales (etc) all capture certain aspects of realms. However, the lack of good support for *views* really hampers the use of these proto-realms as a modular development mechanism. Furthermore, the lack of recognition that "inheritance" (in all its guises) is something best seen on the underlying theory (module, class, trait) graph further complicates matters.

High-Level Realms give practitioners high-level collections of useful symbols and formulae that function like a tool-chest for applications based on the tiny theories developers use as a fine-grained model of dependencies, symbol visibilities, and consistency. For them theories should be static over time, depicting a completed axiomatic development of a mathematical topic. This gives a persistent base (and rigid designators) to develop against. But this means that conservative extensions (like definitions and theorems) need new theories, leading to a severe pollution of the theory namespace. Practitioners, on the other hand, would naturally prefer dynamic theories that continuously grow as new concepts and facts are introduced (another kind of rigid designator).

The contribution of realms is an overlay structure that can implement information hiding and act as fully dynamic high-level theories and an underlying, static theory graph. So users can have their cake and eat it.

7 Representing and Growing Realms in a UDLM

Our work on OMDoc/MMT [Rab; KRZ10] and the MathScheme [CFO11] systems have given us a decent intuition (or so we feel) regarding the services that a theory-graph based system should provide. We now extent this to realms.

Marking Up Realms If we look at the definition of a realm, we see that the body components are already present in the theory graph given by the existing axiomatic developments. Thus, given a theory graph G, we can add a realm R by just tagging a subgraph of G and adding a set of interfaces with their common front (the face of R); all of these are regular components of theory graphs, so we only need to extend the theory graph data structures (and representation languages) by a "realm tagging" functionality. This also shows us that the concept of realms is conservative over theory graphs.

One can easily envision two methods of syntactically identifying realms: globally via a theory-level "realm declaration" which specifies the six components from Definition 1, or locally by extending theory and view declarations with a field that specifies the realm (or realms) it participates in. Given the little theory approach, file-level realm assignments seem a good syntactic compromise.

An implementation will have to check the internal constraints from Definition 1, in particular, that interfaces are total. But the idea of simply "discovering" realms that occur in the wild is a bit optimistic. From our case studies, we expect that realms have to be engineered purposefully: they are grown from a seed, and grow over time by coordinated (and system-supported) additions of theories and views.

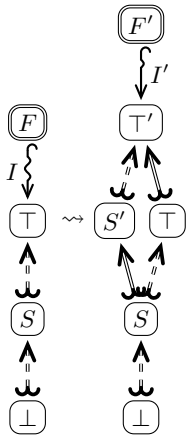
7.1 Supporting the Life Cycle of Realms

We postulate that three realm-level operations will be needed in practice: *i*) realms are initialized by designating chosen theories as initial realms, which *ii*) can be extended by adding conservative extensions, and *iii*) proper realms are created by merging existing realms. These three operations were sufficient to explain the complex realms in our case studies. We will now discuss them in more detail.

Initializing Realms Given a theory graph G, we add any realm (e.g., the initial realm R^T_G for a theory T to G; see Example 1) as a starting point of development.

Extending Realms by (internal) conservative extensions Given a realm $R := (G, F, \mathcal{C}, \mathcal{V}, \mathcal{I})$, a top theory $\top$ of some $C \in \mathcal{C}$, and an interface I for $\top$, then we can extend R by

i) adding a conservative extension $S \hookrightarrow S'$ by declaration c and a commensurate extension $\top \hookrightarrow \top'$ to C and
ii) (optionally) adding a declaration $\underline{c}$ to F, giving a new face F', and extending I so that $I' := I, \underline{c} \mapsto c$. If we do – e.g. for a high-level view – we have to apply *i*) to each of the pillars of R, so that all their interfaces are total; the diagram shows the situation for a simple realm.



realm extension

In particular, an implementation of high-level realms must provide a registration functionality for conservative extensions $\mathcal{C}$ that keeps the interface(s) consistent by ensuring new names appear in the face of the realm. Note that this extension operation does not change the number of pillars of a realm, in particular, if realms are started by initial realms, they will only be extended to simple realms.

The next operation merges two realms if they are mutually interpretable. This operation is mainly used to build up foundational views, the construction makes sure that all symbols in the face are interpreted in all the pillars.

Merging Realms along Views Given two realms $R_1 := (G, F_1, \mathcal{C}_1, \mathcal{V}_1, \mathcal{I}_1)$ and $R_2 := (G, F_2, \mathcal{C}_2, \mathcal{V}_2, \mathcal{I}_2)$, and views $\bot_1 \overset{v}{\rightsquigarrow} \bot_2$ and $\bot'_2 \overset{w}{\rightsquigarrow} \bot'_1$, where $\bot_i$ and $\bot'_i$ are (arbitrary) bottom theories in $\mathcal{C}_i$, then we can define the union realm $R_1 \cup^v_w R_2$ along v and w as $(G, F_1 \cup F_2, \mathcal{C}_1^{+w} \cup \mathcal{C}_2^{+v}, \mathcal{V}_1 \cup \mathcal{V}_2 \cup \{v, w\}, \mathcal{I}_1^{+w} \cup \mathcal{I}_2^{+v})$. Figure 4 shows the situation for two simple realms, generally, we define

i) $\mathcal{C}_1^{+w}$ is the set of conservative developments $\{C^{+w} \mid C \in \mathcal{C}_1\}$, where C^{+w} is C extended by a copy[6] of the development of $\bot'_2$ to $\top'_2$ along w, itself extended to $\top \cup w(\top'_2)$. $\mathcal{C}_2^{+v}$ is defined analogously. In Figure 4, $\mathcal{C}_2^{+v}$ and $\mathcal{C}_1^{+w}$ are the two diamonds on the left and right.

[6] A copy of a development (sub)graph H along a view v is an isomorphic graph H', where for any theory S in H, S' in H' consists of the declarations $c : v(\tau) = v(\delta)$, for all $c : \tau = \delta$ in S. This construction gives us a view $S \overset{v}{\rightsquigarrow} S'$.

note.tex 125 2014-03-16 15:52:19Z kohlhase

ii) $F_1 \cup F_2 \overset{I_1^{+w}}{\rightsquigarrow} \top_1 \cup w(\top_2)$ is $I_1 \cup w \circ I_2$ and $F_1 \cup F_2 \overset{I_2^{+v}}{\rightsquigarrow} \top_2 \cup v(\top_1)$ is $I_2 \cup v \circ I_1$. An implementation of this construction would take great care to merge corresponding symbols in the two faces to minimize the union. Moreover, the copying operation can be optimized to only copy over those conservative extensions that are mentioned in the interface extension.

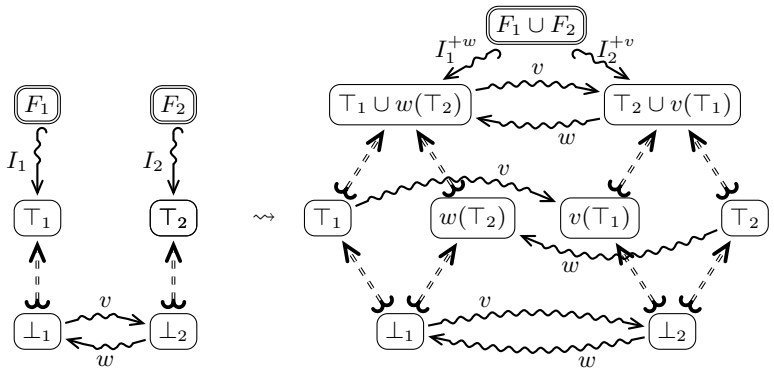


Fig. 4: Union Realm

7.2 Modular Realms

Note that the extension and merging operations highlight an internal invariant of realms that may not have been obvious until now: All pillars of a realm must interpret the full vocabulary of the face to admit total interfaces. This duplication can become quite tedious in practice. Therefore it is good practice to modularize realms in the spirit of a "little realms approach". For instance, the groups realm from Figure 3 could be extended by the usual group theorems via conservative extensions in both pillars. But we can also build a simple realm with base theory group and extend that conservatively (once per theorem). Unless there are proofs that that directly profit from the particulars of the concrete formulations in the pillars below, the modular approach is more efficient representationally and thus more manageable.

7.3 Interface Matters

As the realms are the main interaction points for mathematicians with the UDLM, realms must be discoverable and provide a range of convenient information retrieval methods (after all, realms will get very large in practice). These can range from community tools like peer reviewed periodicals (aka. academic journals) to technical means like intra- and cross-realm search engines (as realms

are built upon theory graphs, specialization of the ♭search engine [KI12] will be a good starting point.)

It will be very important to provide a set of interactions for the interface of a realm that users can understand. It will be important to look up the definienda and proofs of interface items, even though this will usually mean that we need to descend into (conservative extensions of) one of the fronts of the interface, which employ different languages. This needs to be transparent enough to be understandable to users/mathematicians.

Similarly, the equivalence relation of the (tiny) theories that make up the realm should be made transparent and easy to browse to the user.

8 Conclusion

We have presented an extension of the theory graph approach to representing mathematical knowledge. Realms address the impeding mismatch between the successful practice of the little/tiny theory approach natural for developing theory graphs and the high-level theories most useful for practitioners utilizing such mathematical knowledge representations. We have proposed a formal definition for an infrastructure of realms that is conservative over theory graphs and shown its adequacy by applying it to examples from various areas of mathematics and computation.

As a step towards an implementation we have investigated a set of realm-level operations that can serve as a basis for system support of realm management. The next step in our investigation will be realize and test such support in the OMDoc/MMT [Rab; KRZ10] and the MathScheme [CFO11] systems, fully develop the examples sketched in this paper, and test the interactions on developers, students, and practitioners (see section 3).

References

[CF08] Jacques Carette and William Farmer. "High-Level Theories". In: *Intelligent Computer Mathematics*. Ed. by Serge Autexier et al. LNAI 5144. Springer Verlag, 2008, pp. 232–245. DOI: 10.1007/978-3-540-85110-3_19.

[CFO11] Jacques Carette, William M. Farmer, and Russell O'Connor. "MathScheme: Project description". In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 287–288. ISBN: 978-3-642-22672-4. URL: http://imps.mcmaster.ca/doc/cicm-2011-proj-desc.pdf.

[CO12] Jacques Carette and Russell O'Connor. "Theory Presentation Combinators". In: *Intelligent Computer Mathematics*. Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 202–215. ISBN: 978-3-642-31373-8. URL: http://www.cas.mcmaster.ca/~carette/publications/tpc.pdf.

[Far07] William M. Farmer. "Biform Theories in Chiron". In: *MKM/Calculemus*. Ed. by Manuel Kauers et al. LNAI 4573. Springer Verlag, 2007, pp. 66–79. ISBN: 978-3-540-73083-5. DOI: http://dx.doi.org/10.1007/978-3-540-73086-6_6.

[Far11] William M. Farmer. "Mathematical Knowledge Management". In: *Encyclopedia of Knowledge Management*. Ed. by David Schwartz and Dov Te'eni. 2nd ed. Idea Group Reference, 2011, pp. 1082–1089.

[FGT92] William Farmer, Josuah Guttman, and Xavier Thayer. "Little Theories". In: *Proceedings of the 11th Conference on Automated Deduction*. Ed. by D. Kapur. LNCS 607. Saratoga Springs, NY, USA: Springer Verlag, 1992, pp. 467–581.

[Hal] John Halleck. http://home.utah.edu/~nahaj/logic/structures/systems/s4.html. Accessed: 14 March 2014.

[Hal59] Marshal Hall. *The Theory of Groups*. New York: The Macmillan Company, 1959.

[KI12] Michael Kohlhase and Mihnea Iancu. "Searching the Space of Mathematical Knowledge". In: *DML and MIR 2012*. Ed. by Petr Sojka and Michael Kohlhase. in press. Masaryk University, Brno, 2012. ISBN: 978-80-210-5542-1. URL: http://kwarc.info/kohlhase/papers/mir12.pdf.

[KM79] M. I. Kargapolov and J. I. Merzljakov. *Fundamentals of the Theory of Groups*. Graduate Texts in Mathematics. Springer Verlag, 1979.

[KRZ10] Michael Kohlhase, Florian Rabe, and Vyacheslav Zholudev. "Towards MKM in the Large: Modular Representation and Scalable Software Architecture". In: *Intelligent Computer Mathematics*. Ed. by Serge Autexier et al. LNAI 6167. Springer Verlag, 2010, pp. 370–384. ISBN: 3642141277. arXiv: 1005.5232v2 [cs.OH].

[Law04] William F. Lawvere. "Functorial Semantics of Algebraic Theories". In: *Reprints in Theory and Applications of Categories* 4 (2004), pp. 1–121. URL: http://tac.mta.ca/tac/reprints/articles/5/tr5.pdf.

[LR11] Stephen Lack and Jiří Rosický. "Notions of Lawvere Theory". English. In: *Applied Categorical Structures* 19.1 (2011), pp. 363–391. ISSN: 0927-2852. DOI: 10.1007/s10485-009-9215-2.

[Pea89] G. Peano. *Arithmetices principia nova methodo exposita*. Turin, Italy: Bocca, 1889.

[Rab] Florian Rabe. *The MMT Language and System*. URL: https://svn.kwarc.info/repos/MMT (visited on 10/11/2011).

[RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: *Information & Computation* 230 (2013), pp. 1–54. URL: http://kwarc.info/frabe/Research/mmt.pdf.