Composing Mathematical Software Systems via the Math-in-the-Middle Paradigm

Michael Kohlhase

Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
http://kwarc.info

Computer Algebra in the Age of Types, Hagenberg, August 17. 2018

Conclusion

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.

Conclusion and Future Work

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.
- Implementation: Docker with ODK systems and Jupyter front-end at `https://github.com/vv20/mitm_proof_of_concept` (deploy publically soon)

Conclusion and Future Work

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.
- Implementation: Docker with ODK systems and Jupyter front-end at `https://github.com/vv20/mitm_proof_of_concept` (deploy publically soon)
- MitM Economics: these will decide on the utility!
  - MitM network costs = $\mathcal{O}(3k(n+1))$, where $k \widehat{=} \#$ (constr. + API ops.) instead of $\mathcal{O}(nk^2)$ (6 vs. 9 for three systems)
  - MitM joining costs linear in API size. (interoperability workflows star-shaped)
- What can you do?: Connect your system to MitM $\rightsquigarrow$ API theories/Phrasebook
- What will we do?: OpenDreamKit still runs 13 months
  - compiling MitM pivoting translations into P2P translations (eliminate SCSCP too)
  - provide MitM-based documentation for all systems (translate docs not terms)
  - math service discovery (via alignment paths $\leftrightsquigarrow$ priority?)

Running Example/Use Case: Jane's Invariant Experiments

- Jane wants to experiment with invariant theory of finite groups.
  - She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$,
  - Goal: construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I.
  - Idea: pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = Orbit(G, R, p) \subseteq R$.
  - For effective further computation with I, she needs a Gröbner base of I.

Running Example/Use Case: Jane's Invariant Experiments

- Jane wants to experiment with invariant theory of finite groups.
  - She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$,
  - Goal: construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I.
  - Idea: pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = Orbit(G, R, p) \subseteq R$.
  - For effective further computation with I, she needs a Gröbner base of I.
- Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP's orbit algorithm and Singular's Gröbner base algorithm, which she knows to be very efficient.

Running Example/Use Case: Jane's Invariant Experiments

- Jane wants to experiment with invariant theory of finite groups.
  - She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$,
  - Goal: construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I.
  - Idea: pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = Orbit(G, R, p) \subseteq R$.
  - For effective further computation with I, she needs a Gröbner base of I.
- Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP's orbit algorithm and Singular's Gröbner base algorithm, which she knows to be very efficient.
- Problem: Jane has to learn the GAP and Singular languages and retype the results in them. (error-prone)

Running Example/Use Case: Jane's Invariant Experiments

- Jane wants to experiment with invariant theory of finite groups.
  - She works in the polynomial ring $R = \mathbb{Z}[X_1, \ldots, X_n]$,
  - Goal: construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I.
  - Idea: pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = Orbit(G, R, p) \subseteq R$.
  - For effective further computation with I, she needs a Gröbner base of I.
- Jane is a SageMath user and wants to receive the result in SageMath, but she wants to use GAP's orbit algorithm and Singular's Gröbner base algorithm, which she knows to be very efficient.
- Problem: Jane has to learn the GAP and Singular languages and retype the results in them. (error-prone)
- For the sake of example, we will work with $n = 4$, $G = D_4$ (the dihedral group), and $p = 3 \cdot X_1 + 2 \cdot X_2$, but our results apply to arbitrary values.
- Caveat: G is called "D_4" in SageMath but "D_8" in GAP due to differing conventions in different mathematical communities

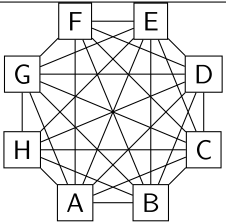
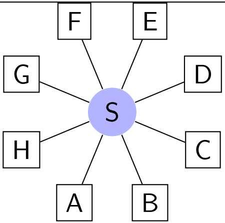
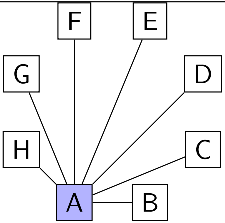
1 Towards a Math VRE
— Interoperability via a Joint Meaning Space —

Interoperability in OpenDreamKit

- OpenDreamKit (ODK): EU Project 2015-19, 16 Partners
  $\leadsto$ build a "mathematical VRE (Virtual Research Environment) toolkit"
- ODK Approach: VRE by connecting existing OSS systems. (and improve them)
- Advantages: well-known Open Source Software
  1. Let the specialists do what they do best and like (and avoid what they don't)
  2. collaboration exponentiates results
  3. competition fosters innovation (+ no vendor lock-in)
- Problem: does an elliptic curve mean the same in GAP, SageMath, LMFDB?
  - otherwise delegating computation becomes unsound
  - storing data in a central KB becomes unsafe
  - the user cannot interpret the results in an UI
- Idea: Need a common meaning space for safe distributed computation in a VRE!

Obtaining a Common Meaning Space for our VRE

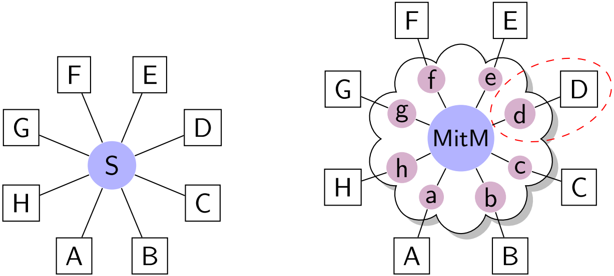
- Three approaches for safe distributed computation/storage/UIs

peer to peer	open standard	industry standard
		
$n^2/2$ translations symmetric	$2n$ translations symmetric	$2n-2$ translations asymmetric

- Observation: We already have a "standard" for expressing the meaning of concepts/objects/models: mathematical vernacular! (e.g. in math. documents)
- Problem: mathematical vernacular is too
  - ambiguous: need a human to understand structure, words, and symbols
  - redundant: every paper introduces slightly different notions.
- Math-in-the-Middle Paradigm: encode math knowledge in modular flexiformal format as a frame of reference for joint meaning (OMDoc/MMT)

Standardization with Interfaces

- Problem: We are talking about knowledge-based systems (large investment)
- Problem: Knowledge is part of both the
  - System ⇝ system-specific representation requirements and release cycle
  - Interoperability Standard ⇝ stability and generality requirements.
- Idea: Open standard knowledge base with API theories



- **Definition 1.1.** API theories are
  - system-near (import/export facilities maintained with system)
  - declarative, in standard format (refine general theories, relation documented)

OpenMath System Dialects

- Observation: Every system has its own input language (optimized to domain)
- Idea: Abstract away from system surface languages (use internal syntax trees)

OpenMath System Dialects

- Observation: Every system has its own input language (optimized to domain)
- Idea: Abstract away from system surface languages (use internal syntax trees)
- Observation: There are two kinds of symbols in syntax trees of a system S
  - constructors build primitive objects without involving computation, and
  - operations compute objects from other objects.
- **Definition 1.2.** The API theories $A(S)$ of S document them $\rightsquigarrow$ we can represent the API of S as *OpenMath* objects with constants from $A(S)$ (the $A(S)$-objects). We call the set of $A(S)$-objects the system dialect of S.

OpenMath System Dialects

- Observation: Every system has its own input language (optimized to domain)
- Idea: Abstract away from system surface languages (use internal syntax trees)
- Observation: There are two kinds of symbols in syntax trees of a system S
  - constructors build primitive objects without involving computation, and
  - operations compute objects from other objects.
- **Definition 1.2.** The API theories $A(S)$ of S document them $\leadsto$ we can represent the API of S as *OpenMath* objects with constants from $A(S)$ (the $A(S)$-objects). We call the set of $A(S)$-objects the system dialect of S.
- Idea: For each system S generate the API theories $A(S)$ and a serializer/deserializer into the system dialect: an OpenMath phrasebook.
- Progress: For system interoperability we only need to relate system dialects meaningfully.

Meaning-Preserving Relations between System Dialects

- **Definition 1.3.** We call a pair of identifiers (a_1, a_2) that describe the same mathematical concept an alignment.
  We call an alignment perfect, if it induces a total, truth-preserving translation. (e.g. alignment up to argument order)
- Intuition: Alignments don't need to be perfect to be useful!
  - Alignment up to Totality of Functions (e.g. division undefined on 0 and with $\frac{x}{0} = 0$)
  - Alignment for Certain Arguments (e.g. Addition on natural numbers and addition on real numbers)
  - Alignment up to Associativity (e.g. binary addition and "sequential" addition)

  They still allow for translating expressions between libraries. (under certain conditions)

MitM-Based Distributed Computation

- Observation: For interoperability between systems A and B with OpenMath phrasebooks and API theories, we only need
  1. a way of transporting *OpenMath* objects between systems A and B
  2. a system dialect mediator that translates A-objects into B-objects based on alignments.
- Idea: Mediator-based architecture

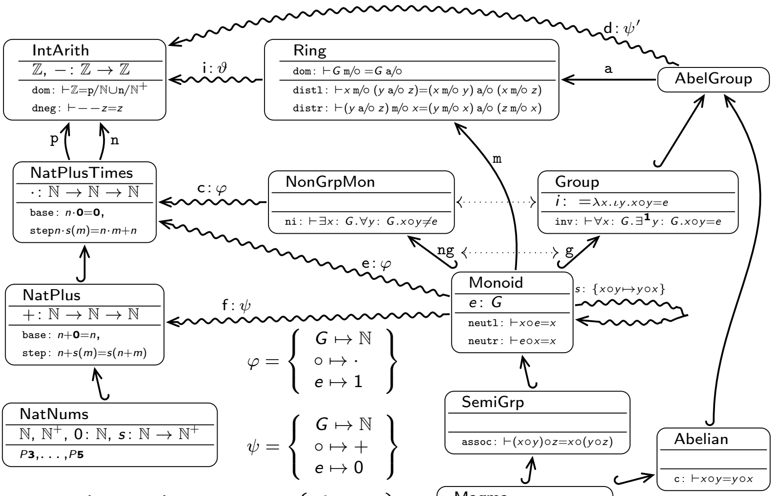
- Idea for 1.: translate A-objects to B-objects in two steps: A to ontology and ontology to B.
  Implemented in [Mül+17] based on the MMT system [Rab13; MMT], which implements the OMDoc/MMT format.
- Idea for 2.: Use the OpenMath SCSCP (Symbolic Computation Software Composability) protocol [Fre+] for that.
  Implemented SCSCP clients/server by for various OpenDreamKit systems.

2 Realizing MitM Interoperability – The Computational Group Theory Case Study –

2.1 Modular Knowledge Representation

Modular Representation of Math (MMT Example)

- **Example 2.1 (Elementary Algebra and Arithmetics).**

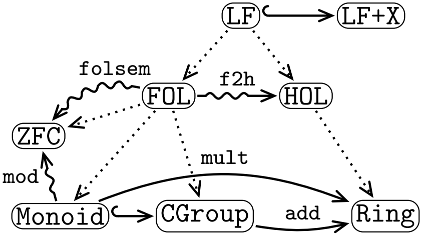


$$\varphi = \left\{ \begin{array}{l} G \mapsto \mathbb{N} \\ \circ \mapsto \cdot \\ e \mapsto 1 \end{array} \right\} \qquad \psi = \left\{ \begin{array}{l} G \mapsto \mathbb{N} \\ \circ \mapsto + \\ e \mapsto 0 \end{array} \right\}$$

$$\vartheta = \left\{ \begin{array}{l} \mathtt{m} \mapsto \mathtt{e} \\ \mathtt{a} \mapsto \mathtt{c} \end{array} \right\} \qquad \psi' = \left\{ \begin{array}{l} i \mapsto - \\ \mathtt{g} \mapsto \mathtt{f} \end{array} \right\}$$

Representing Logics and Foundations as Theories

- **Example 2.2.** Logics and foundations represented as MMT theories



- **Definition 2.3.** Meta-relation between theories – special case of inclusion
- Uniform Meaning Space: morphisms between formalizations in different logics become possible via meta-morphisms.
- **Remark 2.4.** Semantics of logics as views into foundations, e.g., `folsem`.
- **Remark 2.5.** Models represented as views into foundations (e.g. ZFC)
- **Example 2.6.** `mod` $:= \{G \mapsto \mathbb{Z}, \circ \mapsto +, e \mapsto 0\}$ interprets Monoid in `ZFC`.

A MitM Theory in MMT Surface Language

- **Example 2.7.** A theory of Groups

Declaration $\widehat{=}$
name : type [= Def] [# notation]

Axioms $\widehat{=}$ Declaration with type $\vdash F$

ModelsOf makes a record type from a theory.

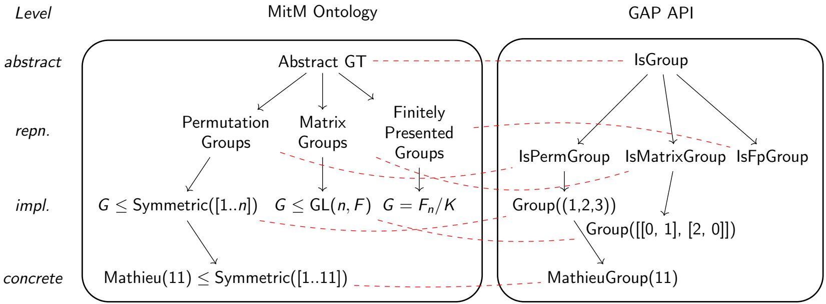
```
theory group : base:?Logic =
  theory group_theory : base:?Logic =
    include ?monoid/monoid_theory ❙

    inverse : U → U ❘ # 1 ⁻¹  prec 24 ❙
    inverseproperty : ⊦ ∀ [x] x ∘ x ⁻¹ ≐ e ❙
  ❚
  group = ModelsOf group_theory ❙
```

- MitM Foundation: optimized for natural math formulation
  - higher-order logic based on polymorphic λ-calculus
  - judgements-as-types paradigm: $\vdash F \mathrel{\widehat{=}}$ type of proofs of F
  - dependent types with predicate subtyping, e.g. $\{n\}\{'a \in \mathrm{mat}(n,n)|\mathrm{symm}(a)'\}$
  - (dependent) record types for reflecting theories

MitM Computational Group Theory

- Four levels of modeling (Following the GAP template)
  - Abstract Level: the group axioms, generating sets, homomorphisms, group actions, stabilisers, orbits, centralizers, normalizers.
  - Representation Level: axiomatizations concrete objects suitable for computation – permutation groups, matrix groups, . . . , also group actions, group homomorphism
  - Implementation Level: permutation groups as subgroups of $S_{\mathbb{N}+}$, concretely $S_{[1,\ldots,n]}$.
  - Concrete Level: where actual computations happen.
- Alignments between the MitM Ontology and the GAP API



2.2 API Theories for Computer Algebra Systems

SageMath, GAP, and Singular API Theories/Phrasebooks

- Observation: Most of the information is already present in mature systems
  - name/type information of constructors and operations.
  - existing API documentation for flexiformal specification.

SageMath, GAP, and Singular API Theories/Phrasebooks

- Observation: Most of the information is already present in mature systems
  - name/type information of constructors and operations.
  - existing API documentation for flexiformal specification.
- SageMath: extracting 500+ API theories
  - type information by "categories" where possible, else introspection of Python classes/method call patterns
  - Phrasebook via Python's pickling infrastructure (full structure sharing)

SageMath, GAP, and Singular API Theories/Phrasebooks

- Observation: Most of the information is already present in mature systems
  - name/type information of constructors and operations.
  - existing API documentation for flexiformal specification.
- SageMath: extracting 500+ API theories
  - type information by "categories" where possible, else introspection of Python classes/method call patterns
  - Phrasebook via Python's pickling infrastructure (full structure sharing)
- GAP: existing phrasebook adapted to 350+ API theories
  - very good, structured API documentation (found 3000 structure errors)
  - regularized constructor calls in ca. 2400 places (performance gain by typed method dispatch?)

SageMath, GAP, and Singular API Theories/Phrasebooks

- Observation: Most of the information is already present in mature systems
  - name/type information of constructors and operations.
  - existing API documentation for flexiformal specification.
- SageMath: extracting 500+ API theories
  - type information by "categories" where possible, else introspection of Python classes/method call patterns
  - Phrasebook via Python's pickling infrastructure (full structure sharing)
- GAP: existing phrasebook adapted to 350+ API theories
  - very good, structured API documentation (found 3000 structure errors)
  - regularized constructor calls in ca. 2400 places (performance gain by typed method dispatch?)
- Singular: thanks to Sebastian Gutsche
  - return types are missing $\leadsto$ infer from call patterns $\leftsquigarrow$ C++ parsing
  - documentation strings are often semi-structured $\leadsto$ string-scraping

The SageMath API Theories

- API theories can be automatically exported from SageMath categories (in-memory structures)
- Problem: SageMath relies on the Python object system where categories are missing
- Solution?: Introspection of method calls for "typical SageMath objects.(what are the mathematically meaningful methods?)
- Future: The (ongoing) port of SageMath to Python 3, will enable gradual typing (MitM type inference?)

The SageMath Phrasebook

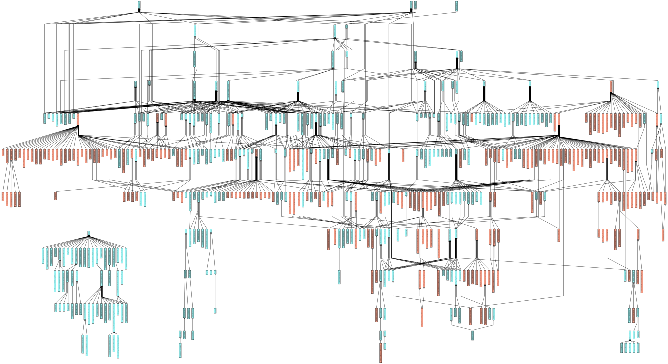
- For the SageMath phrasebook we use Python serialization/deserialization
- **Example 2.8.** The dihedral group D_4 is serialized to

```
pg_unreduce = unpickle_global('sage.structure.unique_representation', 'unreduce')
pg_DihedralGroup =
    unpickle_global('sage.groups.perm_gps.permgroup_named', 'DihedralGroup')
pg_make_integer = unpickle_global('sage.rings.integer', 'make_integer')
pg_unreduce(pg_DihedralGroup, (pg_make_integer('4'),), {})
```

- This is already very close to the SageMath system dialect.
- Extend the Python deserializer to generate OpenMath objects from the constructors.
- We profit from the optimizations (structure sharing) in Python.

The GAP API Theories and Phrasebook

- GAP exports types, constructors, functions, data, and their documentation from type system and documentation.



  This exercise revealed ca. 2000 documentation inconsistencies
- GAP phrasebook serializes/de-serializes OpenMath in JSON and XML
- GAP source code was refactored with ca. 700+1700 constructor macros

  (independently useful for static typing⇝ Markus Pfeiffer)

2.3 Alignments: Glueing System APIs and MitM together

Meaning-Preserving Relations between System Dialects

- **Definition 2.9.** We call a pair of identifiers (a_1, a_2) that describe the same mathematical concept an alignment.
  We call an alignment perfect, if it induces a total, truth-preserving translation. (e.g. alignment up to argument order)
- Intuition: Alignments don't need to be perfect to be useful!
  - Alignment up to Totality of Functions (e.g. division undefined on 0 and with $\frac{x}{0} = 0$)
  - Alignment for Certain Arguments (e.g. Addition on natural numbers and addition on real numbers)
  - Alignment up to Associativity (e.g. binary addition and "sequential" addition)

  They still allow for translating expressions between libraries. (under certain conditions)

Addition on Natural Numbers

- *Constructive Type Theory:* defined as fixed point of some equation (e.g. Coq, Matita)
  In our syntax:

  coq:?Init/Nat?add matita:?nat/plus?plus direction="both"

- Addition defined more generically (restricted to natural numbers via subtyping) (e.g. HOL Light, HOL4, PVS)

  coq:?Init/Nat?add pvs:/Prelude?number_fields?+ direction="forward"

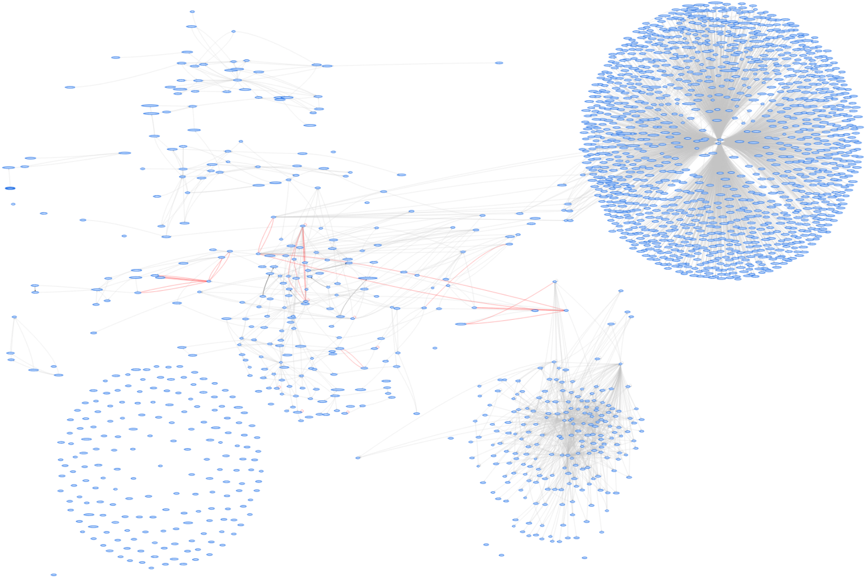
- *Set theories:* least straight-forward; often primitive recursion (e.g. Isabelle/ZFm Mizar)
  Mizar: Ordinal addition $\rightarrow$ rational addition $\rightarrow \mathbb{R}^+ \rightarrow \mathbb{R} \rightarrow \mathbb{C}$ and finally restricted to $\mathbb{N}$.

Collecting Alignments

- Git repository at `https://gl.mathhub.info/alignments/Public`
  - text files with one alignment per line
  - hundreds of manual alignments (students at Jacobs University)
  - thousands of alignments by AI techniques (Cezary Kaliszyk's group)
  - anyone can add new alignments (using pull requests).
- The more alignments we have, the more useful they are

Submit your alignments!

The Knowledge Graph for MitM, SageMath, GAP, Singular



2.4 MitM-based Distributed Computation

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

- The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

- The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.
- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

- The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.
- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- The MitM server translates MitM.Singular(O).Ideal().Groebner() to the Singular system dialect and sends it to Singular..

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

- The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.
- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- The MitM server translates MitM.Singular(O).Ideal().Groebner() to the Singular system dialect and sends it to Singular..
- Singular returns the Gröbner base B.

Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- She calls

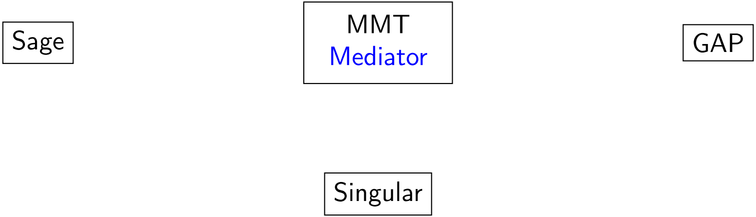
```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```

- The MitM server translates MitM.Gap.orbit(G,A,p) to the GAP system dialect and sends it to GAP.
- GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- The MitM server translates MitM.Singular(O).Ideal().Groebner() to the Singular system dialect and sends it to Singular..
- Singular returns the Gröbner base B.
- The MitM server translates B to the SageMath system dialect and sends it to SageMath, where the result is shown to Jane.

$$B = [X_1 - X_4, X_2 - X_4, X_3 - X_4, 5 * X_4] \, .$$

Distributed Computational Group Theory

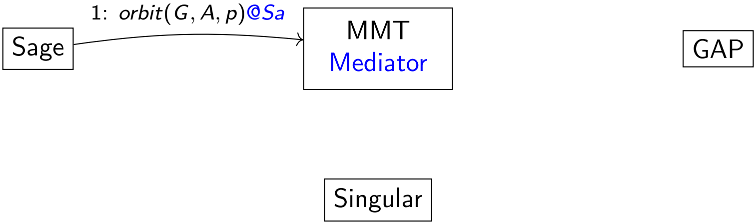
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

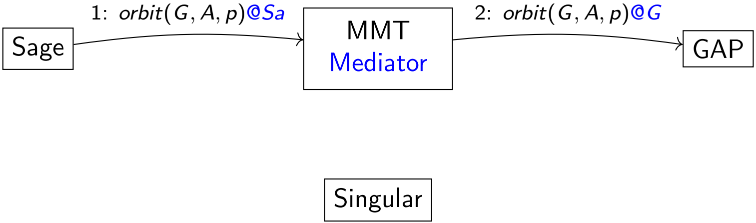
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

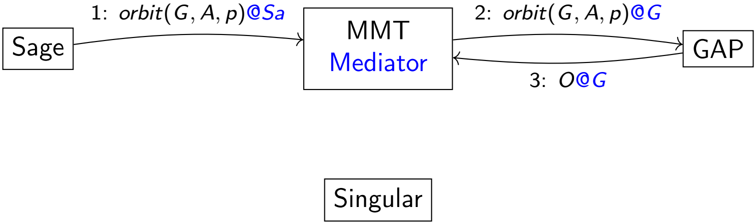
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

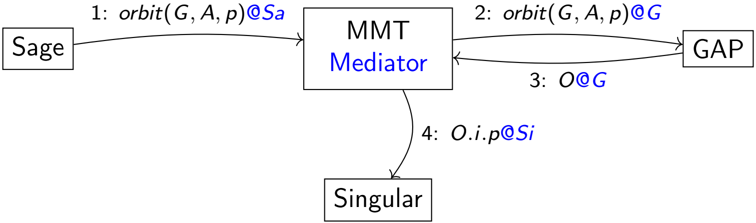
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

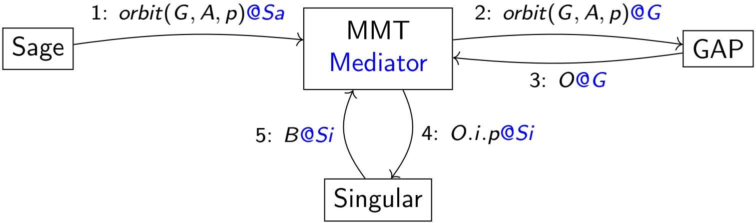
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

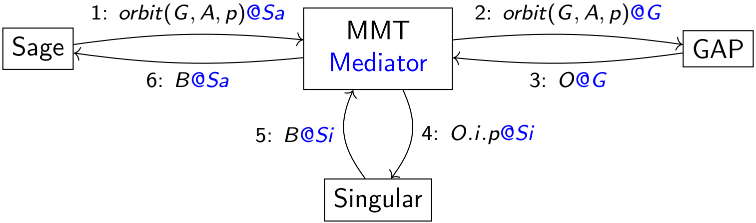
- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

- Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- Nucleus of the OpenDreamKit interoperability layer.
  Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Future Use Case (Steve is Jane's Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- He discovers the GAP package `radiroot` (does not work for p)

Future Use Case (Steve is Jane's Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- He discovers the GAP package `radiroot` (does not work for p)
- Jane suggests PARI/GP: he calls (once that is MitM-connected) `G := MitM("PARIGP","GaloisGroup",p)` from PARI/GP which gives him the desired Galois group as a GAP permutation group.

Future Use Case (Steve is Jane's Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- He discovers the GAP package `radiroot` (does not work for p)
- Jane suggests PARI/GP: he calls (once that is MitM-connected) `G := MitM("PARIGP","GaloisGroup",p)` from PARI/GP which gives him the desired Galois group as a GAP permutation group.
- Steve repeats Jane's experiments on `G`, without leaving GAP.

Future Use Case (Steve is Jane's Colleague)

- Steve prefers working in GAP, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- He discovers the GAP package `radiroot` (does not work for p)
- Jane suggests PARI/GP: he calls (once that is MitM-connected) `G := MitM("PARIGP","GaloisGroup",p)` from PARI/GP which gives him the desired Galois group as a GAP permutation group.
- Steve repeats Jane's experiments on `G`, without leaving GAP.
- Finally, Steve installs a GAP method by calling

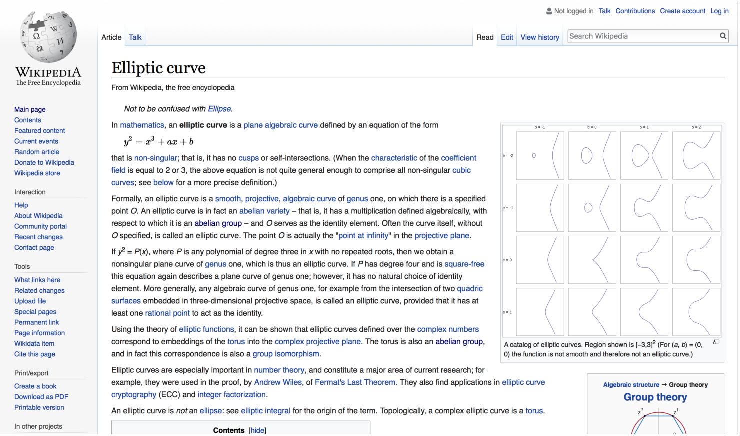
```
InstallMethod(GaloisGroup, "for a polynomial", [IsUnivariatePolynomial],
              p -> MitM("PARIGP", "GaloisGroup", p))
```

  $\leadsto$ extends GaloisGroup to rational polynomials in GAP.
- This replaces a significant part of the 1800-LoC radiroot package (by PARI/GP delegation)

3 Integrating Mathematical Knowledge/Object Bases

Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)



Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects

Elliptic curve

From Wikipedia, the free encyclopedia

Not to be confused with Ellipse.

In mathematics, an **elliptic curve** is a plane algebraic curve defined by an equation of the form

$$y^2 = x^3 + ax + b$$

that is non-singular; that is, it has no cusps or self-intersections. (When the characteristic of the coefficient field is equal to 2 or 3, the above equation is not quite general enough to comprise all non-singular cubic curves; see below for a more precise definition.)

Formally, an elliptic curve is a smooth, projective, algebraic curve of genus one, on which there is a specified point O. An elliptic curve is in fact an abelian variety – that is, it has a multiplication defined algebraically, with respect to which it is an abelian group – and O serves as the identity element. Often the curve itself, without O specified, is called an elliptic curve. The point O is actually the "point at infinity" in the projective plane.

If $y^2 = P(x)$, where P is any polynomial of degree three in x with no repeated roots, then we obtain a nonsingular plane curve of genus one, which is thus an elliptic curve. If P has degree four and is square-free this equation again describes a plane curve of genus one; however, it has no natural choice of identity element. More generally, any algebraic curve of genus one, for example from the intersection of two quadric surfaces embedded in three-dimensional projective space, is called an elliptic curve, provided that it has at least one rational point to act as the identity.

Using the theory of elliptic functions, it can be shown that elliptic curves defined over the complex numbers correspond to embeddings of the torus into the complex projective plane. The torus is also an abelian group, and in fact this correspondence is also a group isomorphism.

Elliptic curves are especially important in number theory, and constitute a major area of current research; for example, they were used in the proof, by Andrew Wiles, of Fermat's Last Theorem. They also find applications in elliptic curve cryptography (ECC) and integer factorization.

An elliptic curve is *not* an ellipse: see elliptic integral for the origin of the term. Topologically, a complex elliptic curve is a torus.

A catalog of elliptic curves. Region shown is $[-3,3]^2$ (For $(a, b) = (0, 0)$ the function is not smooth and therefore not an elliptic curve.)

Algebraic structure → Group theory
Group theory

Contents [hide]

Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)
- Informal mathematical document collections (Cornell preprint arXiv)

arXiv.org > math > arXiv:1711.02170

Search or Article ID | All papers

(Help | Advanced search)

Mathematics > Number Theory

On Elliptic Curves of prime power conductor over imaginary quadratic fields with class number one

John Cremona, Ariel Pacetti

(Submitted on 6 Nov 2017)

The main result of this paper is to generalize from $\Q$ to each of the nine imaginary quadratic fields of class number one a result of Serre and Mestre-Oesterl\'e of 1989, namely that if E is an elliptic curve of prime conductor then either E or a 2-isogenous curve or a 3-isogenous curve has prime discriminant. The proof is conditional in two ways: first that the curves are modular, so are associated to suitable Bianchi newforms; and secondly that a certain level-lowering conjecture holds for Bianchi newforms. We also classify all elliptic curves of prime power conductor and non-trivial torsion over each of the nine fields: in the case of 2-torsion we find that such curves either have CM or with a small (finite) number of exceptions arise from a family analogous to the Setzer-Neumann family of elliptic curves over $\Q$.

Comments:	27 pages
Subjects:	**Number Theory (math.NT)**
MSC classes:	11G05 (Primary), 14H52 (Secondary)
Cite as:	arXiv:1711.02170 **[math.NT]**
	(or arXiv:1711.02170v1 **[math.NT]** for this version)

Download:

- PDF
- PostScript
- Other formats

(license)

Current browse context:

math.NT

< prev | next >

new | recent | 1711

Change to browse by:

math

References & Citations

- NASA ADS

Bookmark (what is this?)

Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)
- Informal mathematical document collections (Cornell preprint arXiv)
- Literature information systems (zbMATH, MathSciNet)

zbMATH

Documents | Authors | Journals | Classification | Software | Formulæ

Structured Search

an:06802543

Fields ▾ Operators ▾

Help ▾

Kriz, Igor

On the arithmetic of elliptic curves and a homotopy limit problem. (English) Zbl 06802543

J. Number Theory 183, 466-484 (2018).

Summary: In this note, I study a comparison map between a motivic and étale cohomology group of an elliptic curve over $\mathbb{Q}$ just outside the range of Voevodsky's isomorphism theorem. I show that the property of an appropriate version of the map being an isomorphism is equivalent to certain arithmetical properties of the elliptic curve.

MSC:

11 Number theory

Keywords:

elliptic curves; Tate-Shafarevich group; homotopy limit problem; motivic cohomology; etale cohomology

BibTeX Cite Full Text: DOI WorldCat

References:

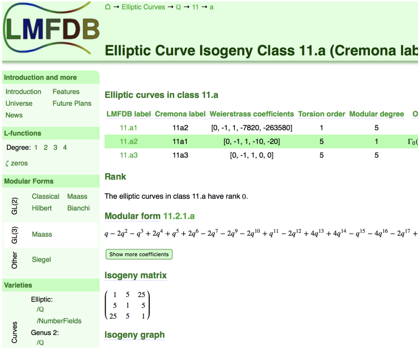
[1] Breuil, C.; Conrad, B.; Diamond, F.; Taylor, R., On the modularity of elliptic curves over $\mathbb{Q}$, or 3-adic exercises, J. amer. math. soc., 14, 849-939, (2001)

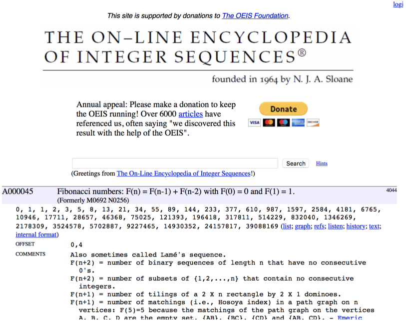
[2] Deligne, P.; Deligne, P., La conjecture de Weil II, Publ. math. IHES, Publ. math. IHES, 52, 137-252, (1980)

[3] Jannsen, U., Continuous étale cohomology, Math. ann., 280, 2, 207-245, (1988)

Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)
- Informal mathematical document collections (Cornell preprint arXiv)
- Literature information systems (zbMATH, MathSciNet)
- Mathematical object databases (GAP libraries, OEIS, LMFDB)





Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)
- Informal mathematical document collections (Cornell preprint arXiv)
- Literature information systems (zbMATH, MathSciNet)
- Mathematical object databases (GAP libraries, OEIS, LMFDB)
- Formal theorem prover libraries (Mizar, Coq, PVS, HOL)

```
|- the_kepler_conjecture <=>
     (!V. packing V
          ==> (?c. !r. &1 <= r
                   ==> &(CARD(V INTER ball(vec 0,r))) <=
                       pi * r pow 3 / sqrt(&18) + c * r pow 2))
```

Mathematical Knowledge Sources (MKS)

- Generic information systems (Wikipedia)
- Informal mathematical document collections (Cornell preprint arXiv)
- Literature information systems (zbMATH, MathSciNet)
- Mathematical object databases (GAP libraries, OEIS, LMFDB)
- Formal theorem prover libraries (Mizar, Coq, PVS, HOL)
- We will concentrate on mathematical object databases here.

Searching in OEIS

- **Question:** Find all sequences starting with $0, 1, 1, 2, 3, 5, 8$

Search: **seq:0,1,1,2,3,5,8**

Displaying 1-10 of 124 results found. page 1 2 3 4 5 6 7 8 9 10 ... 13

Sort: relevance | references | number | modified | created Format: long | short | data

A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1. +20 4044

(Formerly M0692 N0256)

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169 (list; graph; refs; listen; history; text; internal format)

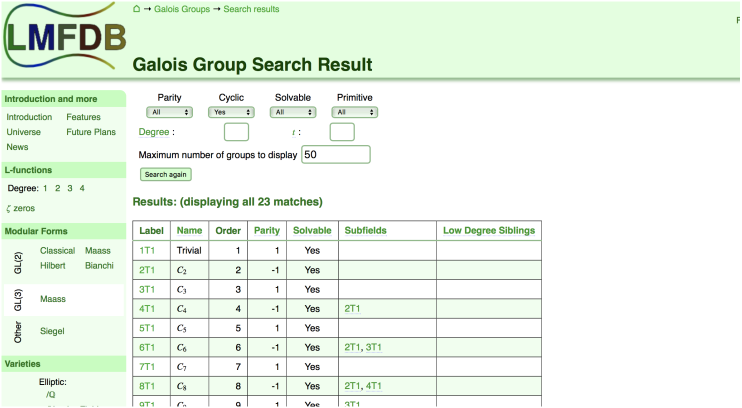
OFFSET 0,4

COMMENTS

```
Also sometimes called Lamé's sequence.
F(n+2) = number of binary sequences of length n that have no consecutive
  0's.
F(n+2) = number of subsets of {1,2,...,n} that contain no consecutive
  integers.
F(n+1) = number of tilings of a 2 X n rectangle by 2 X 1 dominoes.
F(n+1) = number of matchings (i.e., Hosoya index) in a path graph on n
  vertices: F(5)=5 because the matchings of the path graph on the vertices
  A, B, C, D are the empty set, {AB}, {BC}, {CD} and {AB, CD}. - Emeric
  Deutsch, Jun 18 2001
F(n) = number of compositions of n+1 with no part equal to 1. [Cayley,
  Grimaldi]
Positive terms are the solutions to z = 2*x*y^4 + (x^2)*y^3 - 2*(x^3)*y^2 -
  y^5 - (x^4)*y + 2*y for x,y >= 0 (Ribenboim, page 193). When x=F(n), y=F(n
  + 1) and z > 0 then z=F(n + 1).
For Fibonacci search see Knuth, Vol. 3; Horowitz and Sahni; etc.
```

Searching in in the LMFDB

- **Question**: Find all cyclic transitive groups



⌂ → Galois Groups → Search results

LMFDB

Galois Group Search Result

Introduction and more

Introduction Features
Universe Future Plans
News

L-functions

Degree: 1 2 3 4

ζ zeros

Modular Forms

GL(2) Classical Maass
Hilbert Bianchi

GL(3) Maass

Other Siegel

Varieties

Elliptic:
/Q

Parity: All | Cyclic: Yes | Solvable: All | Primitive: All

Degree : t :

Maximum number of groups to display 50

Search again

Results: (displaying all 23 matches)

Label	Name	Order	Parity	Solvable	Subfields	Low Degree Siblings
1T1	Trivial	1	1	Yes		
2T1	C_2	2	-1	Yes		
3T1	C_3	3	1	Yes		
4T1	C_4	4	-1	Yes	2T1	
5T1	C_5	5	1	Yes		
6T1	C_6	6	-1	Yes	2T1, 3T1	
7T1	C_7	7	1	Yes		
8T1	C_8	8	-1	Yes	2T1, 4T1	
9T1	C_9	9	1	Yes	3T1	

- **Problem**: But what if I want to compute with them?

MitM-based Integration of Math Knowledge Bases

- Requirements:
  - a uniformal programatic API to multiple MKB
  - interacting with MKB at the "mathematics Level".
- Idea: use the Math-in-the-Middle Paradigm
  - OMDoc/MMT-based API theories for the mathematical interface ($\leadsto$ MKB records as OM objects)
  - alignments into MitM Ontology (for OM-dialect mediation)
  - extend MMT's built-in query language QMT to general Math query language
- Problems:
  - MKB tables become OMDoc/MMT theories (size problems)
  - how to reconcile MKB records with OMDoc/MMT terms. (encoding/decoding)
  - tow to translate math-level queries to physical database queries

4 Virtual Theories

LMFDB Data (Database Level)

- **Example 4.1 (A transitive group represented in in LMFDB).**

```
{
  "ab": 1,
  "arith_equiv": 0,
  "auts": 1,
  "cyc": 1,
  "label": "1T1",
  "n": 1,
  ...
}
```

Legend: for understanding them (LMFDB improved documentation)

- the cyc field represents being cyclic (0 is **false**, 1 is **true**)
- the n field represents degree (IEEE Float 1 corresponds to $1 \in \mathbb{N}$)
- ...

Two Problems: that have to be solved for MitM integration

- data base schema is not at the mathematical level (let alone interoperable)
- values are encoded for MongoDB convenience (what do they mean?)

Codecs: Encoding and Decoding Database Values

- **Definition 4.2 (Codec).** A codec consists of two functions that translate between semantic types and realized types.

Codecs

codec : type → type	
StandardPos : codec $\mathbb{Z}^+$	JSON number if small enough, else JSON string of decimal expansion
StandardNat :codec $\mathbb{N}$	
▶ StandardInt :codec $\mathbb{Z}$	
IntAsArray :codec $\mathbb{Z}$	JSON List of Numbers
IntAsString :codec $\mathbb{Z}$	JSON String of decimal expansion
StandardBool :codec $\mathbb{B}$	JSON Booleans
BoolAsInt :codec $\mathbb{B}$	JSON Numbers 0 or 1
StandardString :codec $\mathbb{S}$	JSON Strings

- StandardInt decodes 1 into the float 1, but 2^{54} into the string "18014398509481984"

Elliptic Curve Code Operators


```
{
   "degree": 1,
   "x-coordinates_of_integral_points": "[5,16]",
   "isogeny_matrix": [[1,5,25],[5,1,5],[25,5,1]],
   "label": "11a1",
   "_id": "ObjectId('4f71d4304d47869291435e6e')",
   ...
}
```


- Matrix in the isogeny_matrix field
  - $$\begin{bmatrix} 1 & 5 & 25 \\ 5 & 1 & 5 \\ 25 & 5 & 1 \end{bmatrix}$$
  - represented as [[1,5,25],[5,1,5],[25,5,1]]

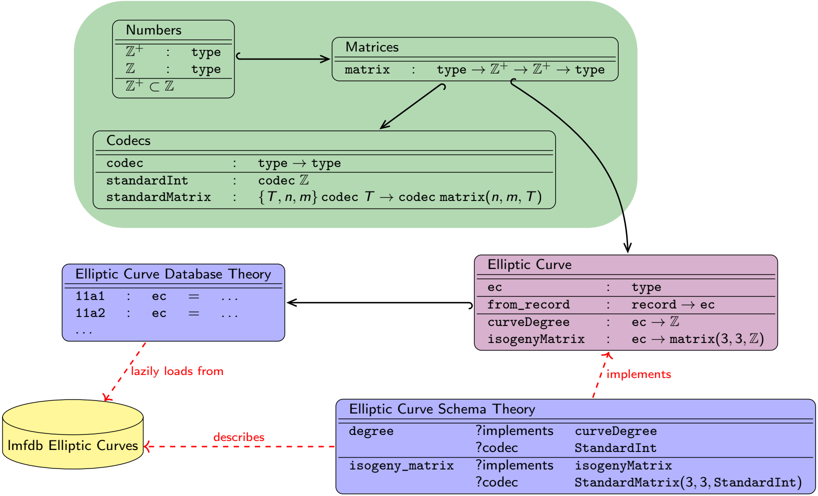
Codec Operator Examples

- **Definition 4.3 (Codec Operator).** A codec operator is a function which takes a codec, a set of parameters, and returns a codec.
- Codecs (continued)

StandardList : codec $T \to$ codec $\mathrm{List}(T)$	JSON list, recursively coding each element of the list
StandardVector : codec $T \to$ codec $\mathrm{Vector}(n, T)$	JSON list of fixed length n
StandardMatrix : codec $T \to$ codec $\mathrm{Matrix}(n, m, T)$	JSON list of n lists of length m

- StandardMatrix(StandardInt, $3, 3$) generates the codec we used for the isogeny matrix

Our approach: Virtual Theories



An Example of a Query

- **Example 4.4.** Finding all cyclic transitive groups in LMFDB (recall from above)

```
x in (related to ( literal ‘lmfdb:db/transitivegroups?group ) by (object declares))
| holds x (x cyclic x *=* true)
```

- This example does not rely on the internal structure of LMFDB
- can be translated into an LMFDB query using the just-defined codecs theory
- `http://www.lmfdb.org/api/transitivegroups/groups/?cyc=1`

Conclusion

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.

Conclusion and Future Work

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.
- Implementation: Docker with ODK systems and Jupyter front-end at `https://github.com/vv20/mitm_proof_of_concept` (deploy publically soon)

Conclusion and Future Work

- For a VRE from Open Source Systems we need a uniform meaning space. (promise/danger in the communication)
- Idea: Center system API theories around the shared math knowledge (Math-in-the-Middle Ontology)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system and SCSCP)
  Use MMT alignments to specify MitM-pivoting translations.
- Implementation: Docker with ODK systems and Jupyter front-end at `https://github.com/vv20/mitm_proof_of_concept` (deploy publically soon)
- MitM Economics: these will decide on the utility!
  - MitM network costs $= \mathcal{O}(3k(n+1))$, where $k \widehat{=} \#$ (constr. + API ops.) instead of $\mathcal{O}(nk^2)$ (6 vs. 9 for three systems)
  - MitM joining costs linear in API size. (interoperability workflows star-shaped)
- What can you do?: Connect your system to MitM $\rightsquigarrow$ API theories/Phrasebook
- What will we do?: OpenDreamKit still runs 13 months
  - compiling MitM pivoting translations into P2P translations (eliminate SCSCP too)
  - provide MitM-based documentation for all systems (translate docs not terms)
  - math service discovery (via alignment paths $\leftrightsquigarrow$ priority?)

Sebastian Freundt et al. *Symbolic Computation Software Composability Protocol (SCSCP)*. Version 1.3. URL: https://github.com/OpenMath/scscp/blob/master/revisions/SCSCP_1_3.pdf (visited on 08/27/2017).

MMT – Language and System for the Uniform Representation of Knowledge. project web site. URL: https://uniformal.github.io/ (visited on 08/30/2016).

Dennis Müller et al. "Alignment-based Translations Across Formal Systems Using Interface Theories". In: *Fifth Workshop on Proof eXchange for Theorem Proving - PxTP 2017*. 2017. URL: http://jazzpirate.com/Math/AlignmentTranslation.pdf.

Florian Rabe. "The MMT API: A Generic MKM System". In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (Bath, UK, July 8, 2013–July 12, 2013). Ed. by Jacques Carette et al. Lecture Notes in Computer Science 7961. Springer, 2013, pp. 339–343. ISBN: 978-3-642-39319-8. DOI: 10.1007/978-3-642-39320-4.