Towards MKM in the Large: Modular Representation and Scalable Software Architecture

Michael Kohlhase, Florian Rabe, Vyacheslav Zholudev

Jacobs University Bremen

Motivation

- Focus: computer-supported deduction, computation, representation of mathematical knowledge
- Well-studied in the small – but serious applications require large scale formalized mathematics, verification, ... distributed, collaborative, web-based
- Problems:
  - systems focus on in-memory processing
  - lack of formalized/annotated content
  - integration, extension of formal systems hard
- Here: infrastructure to separate the concerns
  - small scale systems when possible
  - dedicated systems for large scale aspects
  - flexible/transparent connection between them

Background

- Long term goal:
  - comprehensive framework to represent, integrate, translate, reason about logics
  - apply to all commonly used logics, generate large content base
  - cover model and proof theory
  - provide tool support: validation, browsing, editing, storage, ...
  - digital library of logics
- Observation: objectives highly inter-dependent, e.g.,
  - evaluation of framework requires case studies
  - case studies only feasible with strong module system, editor, etc.

Background

- Long term goal:
  - comprehensive framework to represent, integrate, translate, reason about logics
  - apply to all commonly used logics, generate large content base
  - cover model and proof theory
  - provide tool support: validation, browsing, editing, storage, ...
  - digital library of logics
- Observation: objectives highly inter-dependent, e.g.,
  - evaluation of framework requires case studies
  - case studies only feasible with strong module system, editor, etc.
  - therefore: make them generic

  become separate research projects
- Slogan: If you have 8 hours to chop down a tree, spend 6 sharpening your ax. most of the sharpening done now

Overview

- MMT: scalable representation language
  modular, foundation-independent, web-standard-compliant
  interface layer between formal and web systems
- TNTBase: scalable XML+SVN database [Balisage 2009]
  plugin for server-side MMT processing
- JOBAD: Javascript library for active documents [MKM 2009]
  plugin for client side MMT processing
- LF/Twelf: type theoretical logical framework
  extended for MMT generation [LFMTP 2009]
- MKMIDE: generic editing support [MKM 2010]
  to do: plugin for MMT/LF

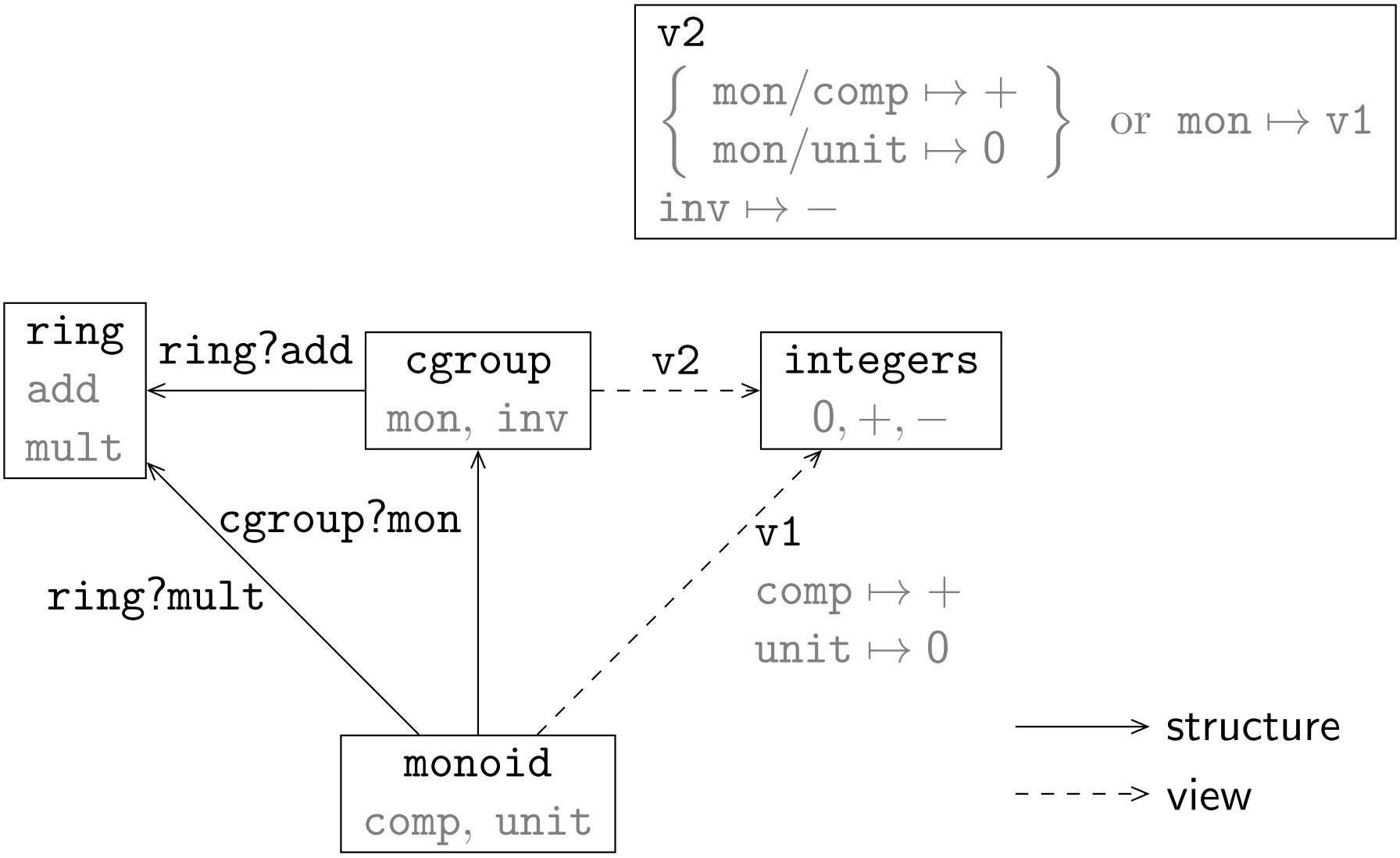
Representation language (MMT)

- MMT = module system for mathematical theories
- Formal syntax and semantics
  - needed for mathematical interface language
  - but how to avoid foundational commitment?
- Foundation-independence
  - identify aspects of underlying language that are necessary for large scale processing
  - formalize exactly those, be parametric in the rest
  - observation: most large scale operations need the same aspects
- Module system
  - preserve mathematical structure wherever possible
  - formal semantics for modularity
- Web-scalable
  - build on XML, OpenMath, OMDoc
  - URI-based logical identifiers for all declarations

Module System

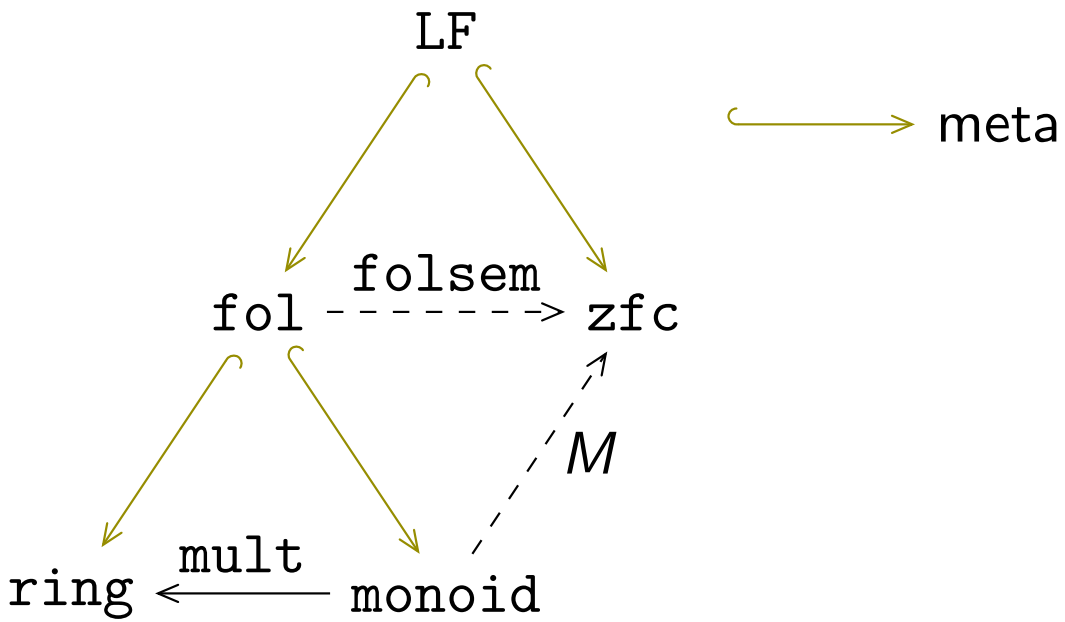
- Central notion: theory graph with theories as nodes and theory morphisms as edges
- Two kinds of theory morphisms
  - structures instantiate theories in a new context (also called: definitional link, import)
    import of theory S into theory T induces theory morphism $S \to T$
  - views translate between existing theories (also called: postulated link, theorem link)

Example


v2
mon/comp ↦ +
mon/unit ↦ 0
or mon ↦ v1
inv ↦ −
ring
add
mult
ring?add
cgroup
mon, inv
v2
integers
0, +, −
cgroup?mon
ring?mult
v1
comp ↦ +
unit ↦ 0
monoid
comp, unit
structure
view

Example (2)

- Logics and foundations represented as theories
  subject to the same module system
- Meta-relation between theories special case of inclusion
- Semantics of logics represented as theory morphisms into the foundation, e.g., `folsem`
- Similarly models M represented as theory morphisms



Module System: Case Studies

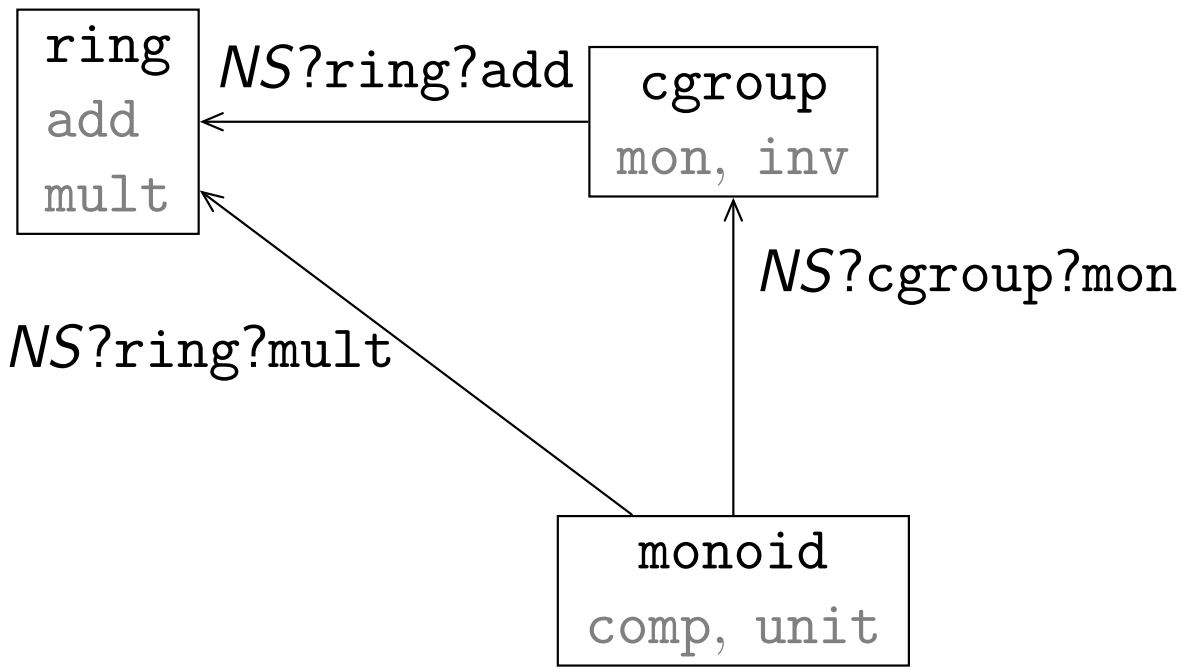
- Created using MMT instantiation of logical framework LF/Twelf
- Fully modular, highly interrelated
- Foundations: ZFC, Isabelle/HOL, Mizar
- Type theories: λ-cube
- Logics: FOL, SFOL, HOL, ML, DL, ...
- Logic translations: SFOL-FOL, FOL-SFOL, FOL-HOL, PL-IPL, ML-FOL, ...
- Algebra: 180 lines structured (1800 lines flattened)
- Lattices: 310 lines structured (3500 lines flattened)
- 100 files, 200 theories, 50 views, 5 authors

Web-Scalability

- Concrete syntax based on XML, OpenMath, OMDoc
- Crucial: good data structure for identifiers
  - all theories, views, constants, structures (including imported ones) addressable by URIs
  - abstract from physical locations
  - implementation needed everywhere: deduction system (SML, Haskell), database (XQuery), browser (Javascript), editor (Bean Shell), etc.
- Intertwined with language design!
  - understanding the identifiers means understanding the module system
  - e.g., MMT inference system based on ADD, GET requests

Example

namespace: NS = http://cds.omdoc.org/algebra.omdoc
theory: NS?monoid
symbol: NS?monoid?unit
imported symbols:
NS?ring?mult/unit
NS?ring?add/mon/unit



MMT Implementation

- Based on Scala — good language and Java-compatible
- Run as API, scriptable shell, local server, web server — use online or offline
- Structural validation: stronger than XML schema but still foundation-independent — incremental
- RDF extraction for indexing — invertible
- Catalog to translate URIs into URLs, hides physical locations
- Query engine for MMT-URIs and MMT-RDF
- Notation language for flexible rendering

Database: TNTBase

- Combines Berkeley DB XML and SVN backend storage
- Versioned XQueryUpdate
- RESTful interface
- Document format-specific abstraction layer
  pre/post-commit hooks, XQuery, Java
- Editable virtual documents corresponding to views in relational databases

MMT in TNTBase

- MMT plugin for TNTBase: MMT-aware querying, searching, indexing
- When committing: TNTBase calls MMT-API to
  - incrementally validate added/changed documents
  - compute and index RDF presentation
- When querying: XQuery functions provided to
  - dereference MMT-URIs
  - query indexed RDF
  - compute dependency closure

Example

- Render the theory *TypedZFC*
  - ZFC set theory + various type theories + views establishing typed reasoning in ZFC
  - depends on about 100 other modules
  - no need to retrieve them all but they may contain notations
- Without TNTBase: recursively retrieve, parse, scan all files, get notations — slow even when all files are local
- With TNTBase: a single XQuery — almost immediate

Conclusion

- MMT representation language
  - modular, foundation-independent, web-scalable
  - interface language between small scale formal systems and large scale web systems
  - to become part of OMDoc 2
- MMT reference implementation
  - incremental, structural validation
  - flexible integration as subsidiary service
  - flexible integration with storage backends
- TNTBase database
  - SVN, XML, XPath, XQueryUpdate
  - stable and released
  - plugin for MMT-aware indexing, querying, search
- Integration with editors, formal systems, web servers, web browsers