Integration of Formal Mathematical Theories

A Challenge for NA-MKM

MICHAEL KOHLHASE

School of Computer Science

Carnegie Mellon University

http://www.cs.cmu.edu/~kohlhase

Demo at http://mbase.mathweb.org:8080/mbase

MBase, a Knowledge Base of Math. Theories

- This has been attempted before! (Principia Math., Bourbaki,...)
- This time stress the infrastructure aspect (Open Source Model)
  - enable easy and powerful browsing (personalization, MathML)
  - high-level (semantic) search (commutativity: $X(Y,Z) = X(Z,Y)$)
  - distributed Internet support: (e.g. local working KB vs. archive KB)
  - version management and concurrent access (like CVS for cooperation)
  - offer added-value inference services (enlist MathWeb)
  - large-scale structure for navigation & reuse (theory graph, inheritance)
- MBase as a MathWeb component. (not only for human consumption)
  - situated vs. stateless communication of mathematical services

MBase and the Semantics of communicated Objects

- Why is this an issue? (preserving the meaning across transport)
  - e.g. the Ricci-Tensor $\mathcal{R}^{ij}$ differs between schools of physicists by a factor of 2!
  - Is this unit in psi or erg? (Remember the Mars orbiter †1999)
- OpenMath/C-MathML approach: Objects as logical formulae
  - sufficient: semantics of constants (that of var, appl, bind is well-known)
  - specify semantics by reference to joint ontology ($\Longrightarrow$ OMDoc)
- MBase as an ontology-server for MathWeb
  - establishes unique reference for objects (distinguished URI)
  - serves knowledge on demand (just-in-time math)
  - offers dynamic communication caches (local context)
  - seamless integration of local and global context
  - Problems: Caching, version management,distributed garbage collection

The MBase System: Architecture

- MathWeb for distribution (mOZart-internal and XML-RPC)
- OMDoc for Communication (XML/OpenMath/MathML-based)
  - Precise document model allows to decouple interface development
- RDBMS for persistence (large amount of data)

  currently: MySQL (archive server) JDom impl. (scratch-pad/cache)
  - encode mathematical structure in database model
  - use Oz pickling to store mOZart data structures as strings in DB.
  - use concurrent constraints in mOZart to manipulate logic terms.
- InKa for management of theory change
  - management of inheritance structure, proof obligations
  - (concurrent access and update) (like CVS, but object-level diff/patch)

An Experiment in Data/Knowledge Integration

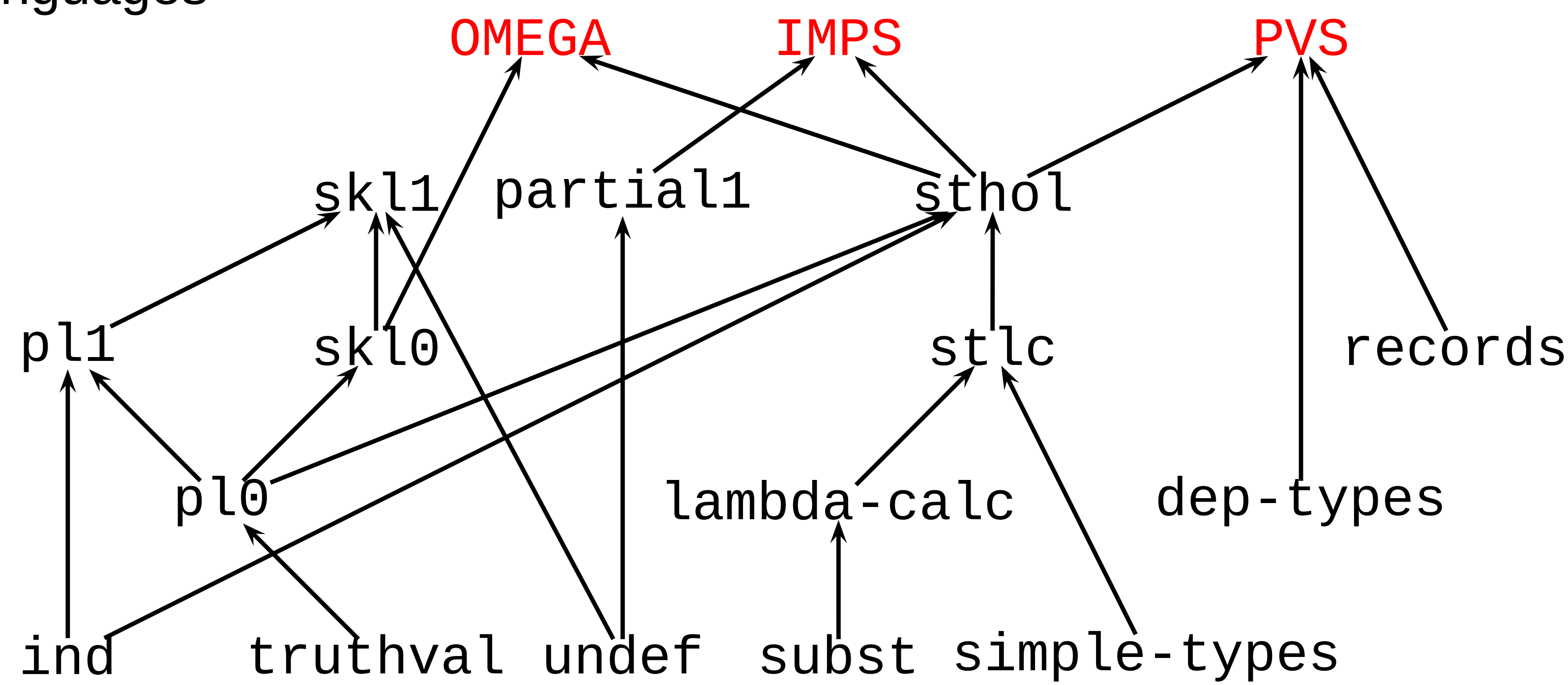
- Goal: connect various theorem proving systems to MBASE
- Systems: ΩMEGA, INKA, PVS, λ*Clam*, TPS, IMPS and COQ
- Similarities: (all descendents of AUTOMATH)
  - TPS, PVS, ΩMEGA, λ*Clam*, and IMPS based on simply typed λ-calculus.
  - ΩMEGA, INKA, IMPS and PVS have higher sort concepts.
  - PVS and COQ allow dependent- and record types
- Differences: (serving different purposes)
  - INKA, PVS, and COQ support inductive definitions, (but by very different mechanisms and on differing levels)
  - IMPS supports partial functions, theory interpretations
  - COQ is constructive, rest classical.

The Experiment (continued)

- Formula level (system-specific representation languages)
  - Use `<OMS cd="sys" name="op">` for logical operators
  - specify the representation language of `sys` by `<theory id="sys">`
  - use similarity among systems to define common language cores
  - communication immediate in common fragments!
- Statement level (Def,Thm,Proof,...)
  - OMDoc1.1 sufficient for ΩMEGA, INKA, PVS, λ*Clam*, TPS, IMPS
  - `<definition>` too weak for COQ (mutuality) (OMDoc2.0)
- Theory level (Development Graph)
  - OMDoc1.1 sufficient for ΩMEGA, INKA, COQ, λ*Clam*, TPS, IMPS
  - PVS has param. theories, quantification over parameters (breaks OPENMATH)

A standardized Hierarchy of logical languages

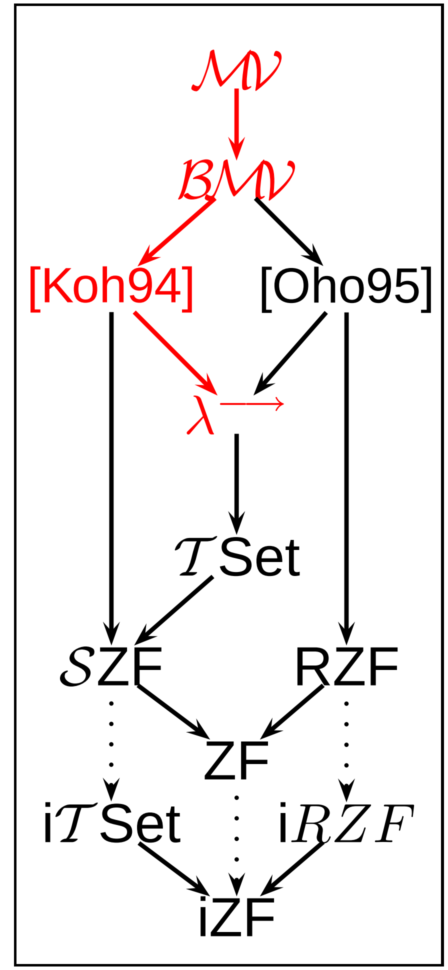
- Idea: Provide a standardized, well-documented set of "names" for logical languages



- This hierarchy is based on literal inclusion (can we wo better?)
- MBASE: Conservative Extension Principle with Logic Morphisms (Extend the Hierarchy with a level of proofs.)

Logic Morphisms

- Definition: Logical System $\mathcal{S} = (\mathcal{L}, \mathcal{C})$,
  - $\mathcal{L}$ language (set of well-formed formulae)
  - $\mathcal{C}$ calculus (set of inference rules)
  - $\mathcal{D}: \mathcal{H} \vdash_{\mathcal{C}} \mathbf{A}$ is a $\mathcal{C}$-derivation of $\mathbf{A}$ from $\mathcal{H}$
- Definition: Logic Morphism $\mathcal{F}: \mathcal{S} \longrightarrow \mathcal{S}'$,
  - Language Morphism $\mathcal{F}^{\mathcal{L}}: \mathcal{L} \longrightarrow \mathcal{L}'$
  - Calculus Morphism $\mathcal{F}^{\mathcal{D}}$ from $\mathcal{C}$-derivations to $\mathcal{C}'$-derivations, such that for any $\mathcal{C}$-derivation $\mathcal{D}: \mathcal{H} \vdash_{\mathcal{C}} \mathbf{A}$, we have $\mathcal{F}^{\mathcal{D}}(\mathcal{D}): \mathcal{F}^{\mathcal{L}}(\mathcal{H}) \vdash_{\mathcal{C}'} \mathcal{F}^{\mathcal{L}}(\mathbf{A})$.
- Logic morphisms transport proofs!



Relativization = Morphism from $\mathbb{S}FOL$ to FOL

- Signature: $\mathcal{R}([+ :: \mathbb{N} \to \mathbb{N} \to \mathbb{N}]) = \forall X, Y. \mathbb{N}(X) \wedge \mathbb{N}(Y) \Rightarrow \mathbb{N}(X + Y)$.
- Formulae: $\mathcal{R}(\forall X_{\mathbb{B}}. \mathbf{A}) = \forall X. \mathbb{B}(X) \Rightarrow \mathcal{R}(\mathbf{A})$
- Sorts: $\mathcal{R}\left(\dfrac{\mathbf{A} :: \mathbb{B} \to \mathbb{C} \quad \mathbf{B} :: \mathbb{B}}{\mathbf{AB} :: \mathbb{C}}\right) = \dfrac{\dfrac{\forall X. \mathbb{B}(X) \Rightarrow \mathbb{C}(\mathbf{A}X)}{\mathbb{B}(\mathbf{B}) \Rightarrow \mathbb{C}(\mathbf{AB})} \quad \mathbb{B}(\mathbf{B})}{\mathbb{C}(\mathbf{AB})}$
- Proofs: $\mathcal{R}\left(\dfrac{\forall X_{\mathbb{B}}. \mathbf{A} \quad \mathbf{B} : \mathbb{B}}{[\mathbf{B}/X]\mathbf{A}}\right) = \dfrac{\dfrac{\forall X. \mathbb{B}(X) \Rightarrow \mathcal{R}(\mathbf{A})}{\mathbb{B}(\mathcal{R}(\mathbf{B})) \Rightarrow \mathcal{R}([\mathbf{B}/X]\mathbf{A})} \quad \mathbb{B}(\mathbf{B})}{\mathcal{R}([\mathbf{B}/X]\mathbf{A})}$

Integrating Theories: e.g. sets in PVS vs ΩMEGA

- Which symbols are defined? How do we map them?

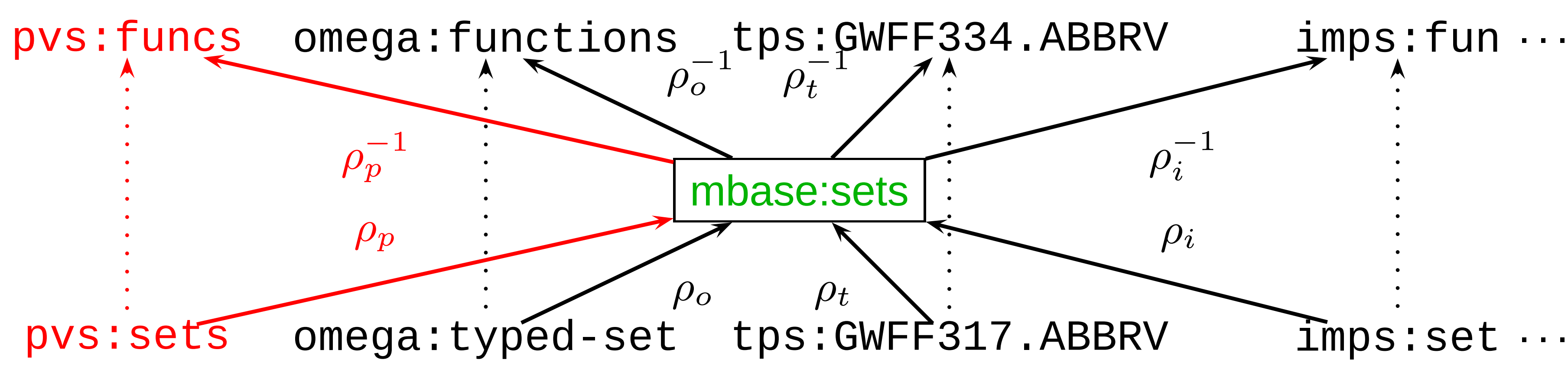
PVS	ΩMEGA	PVS	ΩMEGA
set		subset?	subset
member	in		subset2
empty?	empty	strict_subset?	proper-subset
emptyset	emptyset		superset
nonempty?	not-empty	union	union
full?			union2
fullset			union-over-collection
singleton?	singleton	intersection	intersection
singleton			intersection-over-coll.
complement	set-complement	disjoint?	misses
difference	setminus	meets	
symmetric_difference		add	add-one
	exclunion	remove	

Integrating Theories: PVS vs ΩMEGA

- Further PVS symbols: `every`, `some`, `the`, `choose`, `rest` (defined in `base.thy` in ΩMEGA)
- Further OMEGA symbols: `powerset`, `set=`, `strange-ho-abbr`, `pair`, `cartesian-product`, `first-of-pair`, `second-of-pair`, `pair-operation`, (in type system)
  `has-fixpoint`, `is-constant-map`, `is-identity`, (with functions?)
  `finite-set` (`finite_set.pvs`)
- Problems: (all of the systems have grown organically)
  - differing names, definitions, theory boundaries (differing intuitions?)
  - artefacts of constraints imposed by the system (file lengths, etc.)
  - part of the theories hard-coded into the system (base logic, notations)
  - no theory organization at all (e.g. TPS has flat library with abbrevs.)

Strategies for Theory Integration

- Idea: Use theory interpretations to establish inclusion into union theory



- move all the results to the union theory via theory-interpretations ρ_*, collect the results there. (may need to simplify)
- repeat recursively, until done
- Problems: needs a lot of manual labor, users must rel-learn? (expensive)
- Benefits: true interoperability of math software systems re-use the work of others (let's stand on the shoulders of giants)

Challenge: It is time for another stab at QED

- A mathematical Infrastructure based on Communication and Knowledge
  - MATHWEB as a framework for distributed web-based math. services
  - MBASE as a knowledge base with added-value services
  - Many other systems I have not mentioned (but not forgotten)
  - Standardized communication languages (based on XML)

  gives us the beginning of an integrated framework for MKM.
- Challenge: Avoid fragmentation!
  - Concentrate on Interoperability (translation + migration vs. prescription)
  - "Open-Knowledge" model (instead of local fiefdoms)
  - Embrace "Web Services" idea (distribution as a way to unite)
- Let us work together to create a great resource!