Towards Knowledge Management for HOL Light

Cezary Kaliszyk Florian Rabe

University of Innsbruck, Austria

Jacobs University, Bremen, Germany

MKM 2014

Motivation 1: System Compatibility

- Developments in one system can be used in another
  - shared library / library translations
- In practice: Most systems are not compatible
  - Typically only the system can parse its library
- Positive exceptions
  - Matita and Coq shared the format once
  - Various Translations (HOL Light $\rightarrow$ Isabelle/HOL)

Motivation 2: System Interoperability

- One system can be called while working in another one
- In practice: common for a main system to outsource . . .
  - specialized tools
    e.g., decision procedures, theory exploration
  - automated provers, model finders
    e.g., use ATPs in proof assistant
  - computation systems
    use computer algebra system in deduction system
- . . . but mostly . . .
  - restricted to individual system pairs
  - brittle ad hoc connections
  - no symmetric interoperability

Motivation 3: Library compatibility

$\mathbb{R}$ defined using:

- Cauchy sequences
- Dedekind cuts
- ...

Next talk

Motivation 4: Library Management

- Same functionality needed in every system
  - browsing, navigation
  - distribution, versioning
  - search, querying
  - refactoring, change management
- Dilemma
  - typically not interesting for proof assistant developers
  - but necessary for large scale case studies
- Could be realized generically
- In practice: only system-specific ad hoc solutions (if any)

The HOL Light System

- HOL Logic
  - Church simple type theory
  - Shallow polymorphism
  - Small inference system
    - 10 basic rules
    - 3 extension principles
- HOL Light kernel
  - LCF style
  - Private OCaml types for HOL types, terms, theorems
  - List references to store results of extensions

The HOL Light Library

- HOL Light core system
  - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, lists, sets
  - And their basic properties
  - $\approx$ 2,000 theorems
- HOL Light standard library
  - $\mathbb{R}^{\mathbb{N}}$, 100 theorems, $\binom{n}{k}$, $\lfloor x \rfloor$, ...
  - $\approx$ 17,000 theorems
- Flyspeck
  - Fans, Graphs, Packings, ...
  - $\approx$ 14,000 theorems

MMT

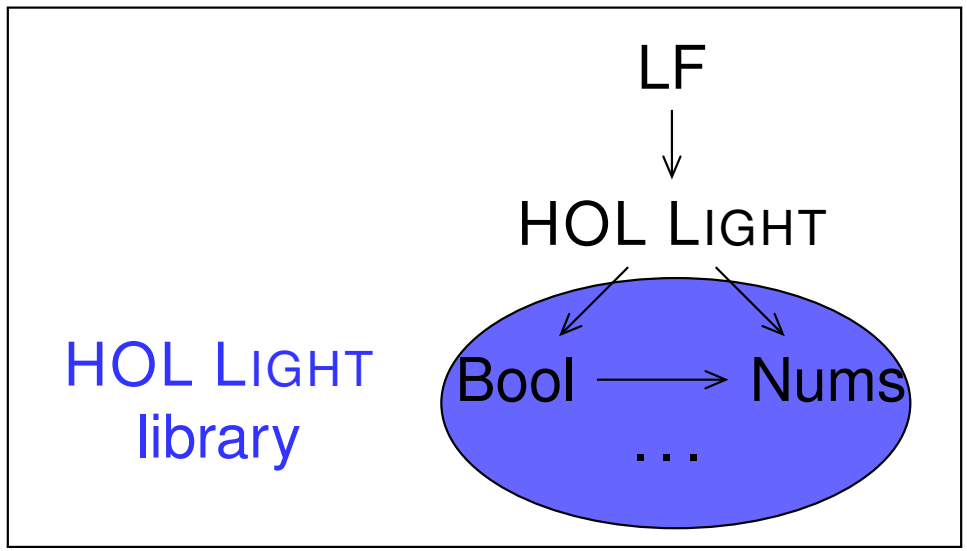
- Representation language for formal mathematical content
  - Foundation-independent
- Heterogeneous
  - Defining logical frameworks, logics, theorems in one syntax
- Implementation with generic
  - module system
  - parsing + type reconstruction
  - IDE
  - change management
- Category theory semantics
  - theories, morphisms, declarations, expressions
- Developed since 2007, $>$ 30000 lines of Scala code
  - OMDoc/OpenMath-based XML syntax with Scala-based API
- Close relatives:
  - Fixed logical framework: LF, Isabelle, Dedukti
  - Hets: but declarative logic definitions

Exporting the HOL Light Library

- Popular integration test case due to
  - simplicity of logic
  - size of the kernel
- Examples exports
  - to Isabelle/HOL — Obua, 2006; Kaliszyk 2013
  - to OpenTheory — Hurd, 2011
  - to Coq — Keller, Werner, 2010
  - to Dedukti — Dowek et al., 2013
- But: already the exports are adapted to the target system

Approach

- Use MMT for logic, data and infrastructure
  1. LF represented and implemented within MMT
  2. HOL Light logic (kernel) represented as LF theory
     - (formalization of HOL Light kernel)
  3. Automatically exported HOL Light library
     - OMDoc theories
- All part of the same MMT theory graph



The HOL LIGHT Logic in MMT/LF (1)

- One LF-type per concept
  - types, term, theorems
- Constructors for primitive operators
  - booleans, equality, λ-calculus

```
holtype : type
term    : holtype → type
thm     : term bool → type

bool    : holtype
fun     : holtype → holtype → holtype
Abs     : {A,B} (term A → term B) → term (A ⇒ B)
Comb    : {A,B} term (A ⇒ B) → term A → term B
equal   : {A} term A ⇒ (A ⇒ bool)
```

The HOL LIGHT Logic in MMT/LF (2)

- Curry-Howard: proofs as terms
- A constructor for each primitive proof rule

```
REFL     : {A,X:term A} ⊢ X = X
TRANS    : {A,X,Y,Z:term A}
  ⊢ X = Y → ⊢ Y = Z → ⊢ X = Z
MP       : p,q ⊢ p = q → ⊢ p → ⊢ q
BETA     : {A,B,F:term A → term B,X:term A}
           ⊢ (λ F)'X = (F X)
MK_COMB  : {A,B, F,G:term A⇒B, X,Y:term A}
  ⊢ F = G → ⊢ X = Y → ⊢ F'X = G'Y
ABS      : {A,B, S,T:term A → term B}
  ({x: term A} ⊢ (S x) = (T x)) → ⊢ λ S = λ T
DEDUCT_ANTISYM_RULE
         : {p,q} (⊢ p →⊢ q) → (⊢ q →⊢ p) →⊢ p = q
```

The HOL LIGHT Logic in MMT/LF (3)

One pattern declaration for each primitive extension principle

- Definition
- Type definitions
- (HOL axioms can be realized by meta axioms)

[Horozal, Kohlhase, Rabe, MKM 2012]

```
extension definition =
   [n: nat] [A: holtype^n → holtype]
   [a: {T: holtype^n} term (A T)]
      c   : {T} term (A T)
      DEF : {T} ⊢ (c T) = (a T)

extension new_basic_type_definition =
   ...
```

Exporting the Library

- Gathering an export list
  - Theories
  - Types, Constants, Definitions
  - Notations
- OMDoc theory file for each HOL Light file
  - MMT constants for types, constants, theorems

```
<constant name="PRE"><type>
  <om:OMOBJ xmlns:om="http://www.openmath.org/OpenMath">
    <om:OMS module="LF" name="apply"></om:OMS>
    <om:OMS module="Kernel" name="term"></om:OMS>
    <om:OMA>
      <om:OMS module="LF" name="apply"></om:OMS>
      <om:OMS module="Kernel" name="fun"></om:OMS>
      <om:OMS module="nums" name="num"></om:OMS>
      <om:OMS module="nums" name="num"></om:OMS>
    </om:OMA>
  </om:OMA></om:OMOBJ>
</type></constant>
```

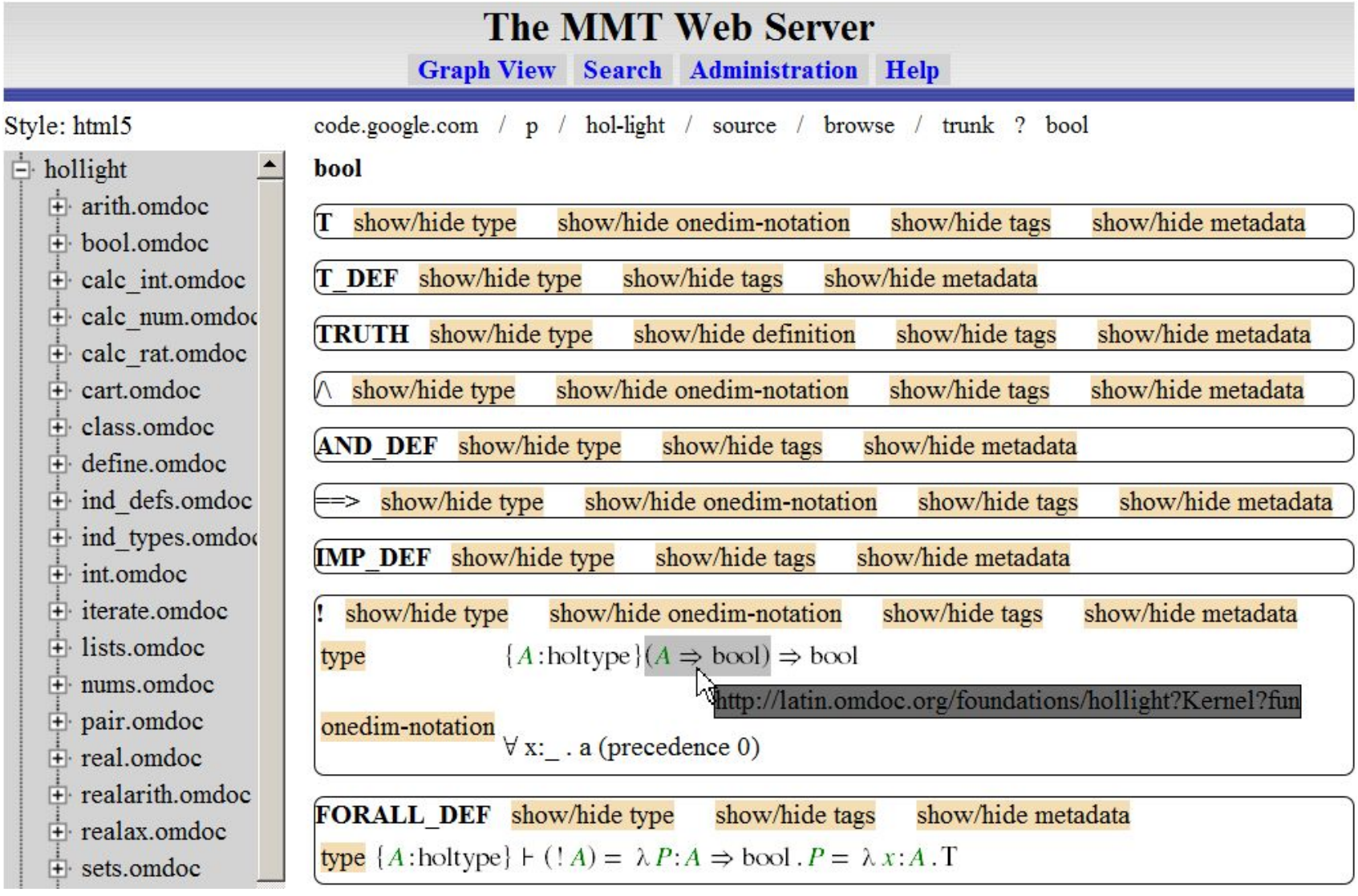
Goal: Generic Library Management

- Library browser
  - MMT generates HTML (Presentation MathML)
  - interactive (JavaScript)
  - semantics-aware

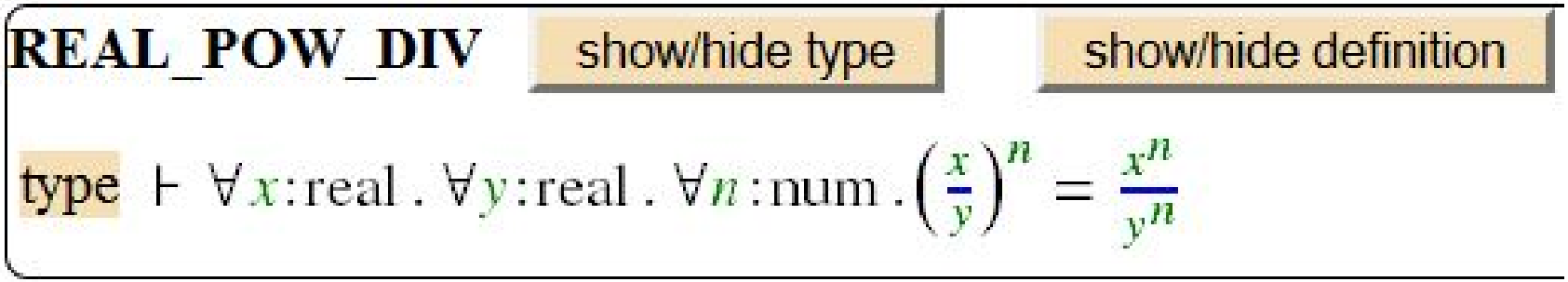
    e.g., dynamic type inference of subterms
  - cross-library browsing
- Search
  - MMT generates index for MathWebSearch

    [Kohlhase et al.]
- Change management
  - export/detect dependencies between library items
  - detect changes between library versions
  - propagate changes along dependencies

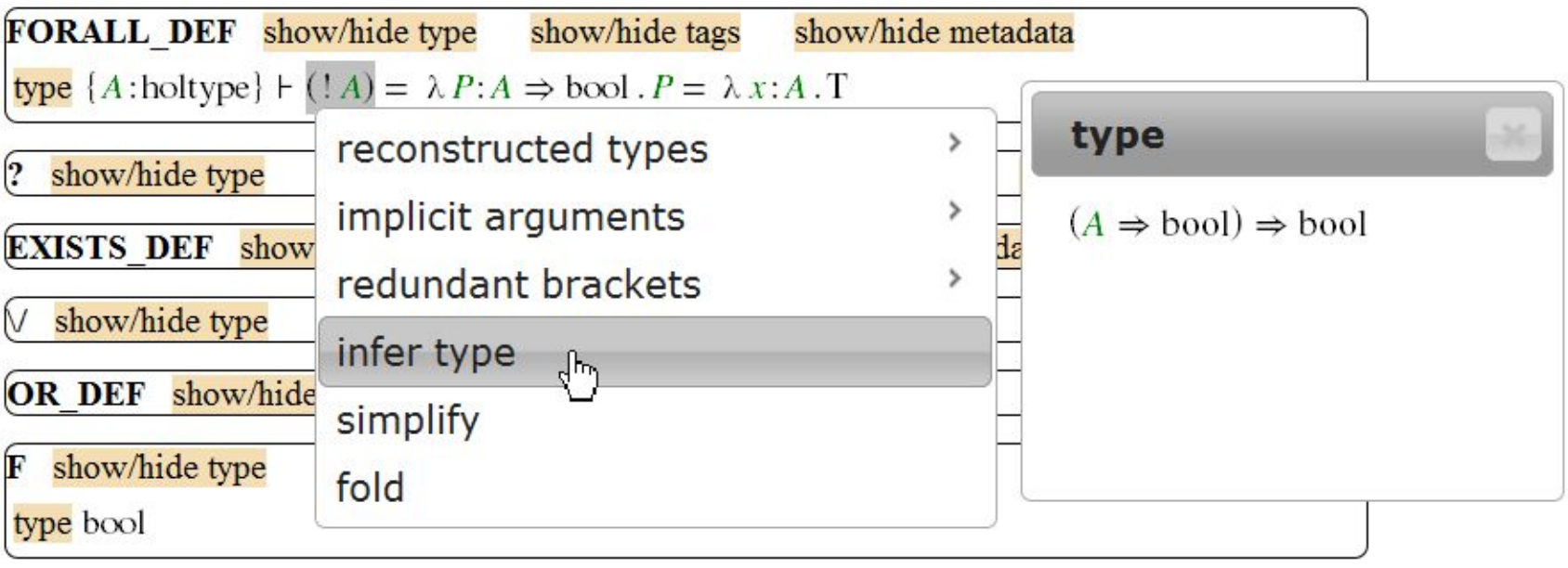
Example Service: The MMT Browser



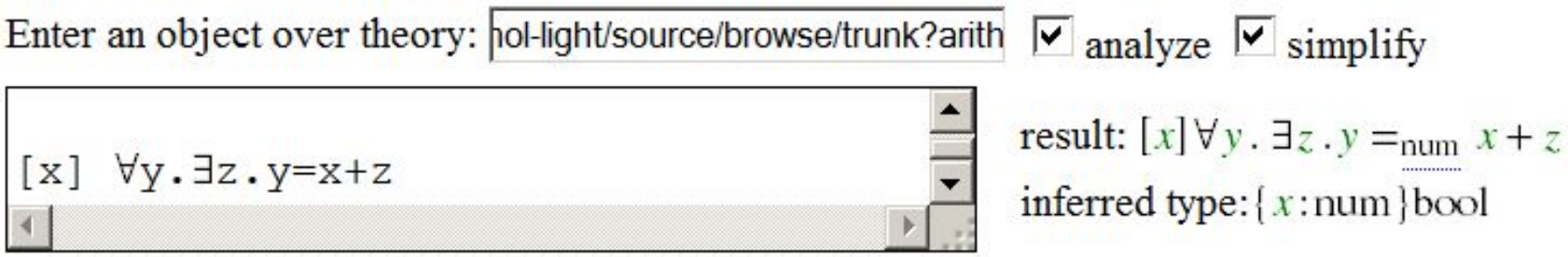
Browser Features: 2-dimensional Notations



Browser Features: Type Inferece



Browser Features: Parsing



Example Service: Search

Enter Java regular expressions to filter based on the URI of a declaration

Namespace

Theory

Name

Enter an expression over theory http://code.google.com/p/hol-light/source/browse/trunk

$x,y,p: x MOD p = y MOD p

Use $x,y,z:query to enter unification variables.

Search

type of MOD_EQ

$$\vdash \forall m{:}\text{num} . \forall n{:}\text{num} . \forall p{:}\text{num} . \forall q{:}\text{num} . m = n + q * p \Longrightarrow m \text{ MOD } p = n \text{ MOD } p$$

type of MOD_MULT_ADD

$$\vdash \forall m{:}\text{num} . \forall n{:}\text{num} . \forall p{:}\text{num} . (m * n + p) \text{ MOD } n = p \text{ MOD } n$$

Conclusion

- Complete export of HOL LIGHT
  - Kernel
  - Library files as independent theories
- MMT services avaliable for HOL LIGHT users
  - Interactive browsing, search, parsing
- Future work
  - Refactoring (to introduce heterogeneity)
  - Correspondences between concepts in different libraries
    - as an MKM concept
    - partial morphisms?