Proof Engineering of Higher Order Logic

Collaboration, Translation, Checking and Retrieval

Robert White (Shuai Wang)

ILLC/INRIA → CNRS

AITP'16

Outline

Higher Order Logic

- Simple type theory (STT) is also known as Higher order logic (HOL).
- HOL = simply typed λ-Calculus + boolean types + axioms + inference rules.
- Most mathematical objects/theories can be expressed in HOL.
- Interactive and automatic theorem provers & proof checkers for formal mathematics.
- HOL Light, ProofPower, HOL4, HOL Zero . . . [HOL family].
- The first talk by Tom Hales: Kepler Conjecture

Kernel

HOL syntax:

type variables	α, β
type operators	p
types	$A, B ::= \alpha \mid p(A_1, \ldots, A_n)$
term variables	x, y
term constants	c
terms	$M, N ::= x \mid \lambda x : A.M \mid MN \mid c$

Polymorphic Type:

$$= : \alpha \to \alpha \to o$$

Primitive Inference Rules

Structural	$\dfrac{}{\{A\} \vdash A}\ ASSUME$
λ Calculus	$\dfrac{\Gamma \vdash A = B}{\Gamma \vdash \lambda x.A = \lambda x.B}\ ABS$ $\dfrac{}{(\lambda x.A)x = A}\ BETA$
Instantiation	$\dfrac{\Gamma[x_1, \ldots, x_n] \vdash A[x_1, \ldots, x_n]}{\Gamma[t_1, \ldots, t_n] \vdash A[t_1, \ldots, t_n]}\ INST$ $\dfrac{\Gamma[\alpha_1, \ldots, \alpha_n] \vdash A[\alpha_1, \ldots, \alpha_n]}{\Gamma[\gamma_1, \ldots, \gamma_n] \vdash A[\gamma_1, \ldots, \gamma_n]}\ INST\_TYPE$
Bi-implication	$\dfrac{\Gamma \vdash A = B \qquad \Delta \vdash A}{\Gamma \cup \Delta \vdash B}\ EQ\_MP$ $\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{(\Gamma \setminus \{B\}) \cup \Delta \setminus \{A\}) \vdash A = B}\ DEDUCTANTISYMRULE$
Equality	$\dfrac{}{\vdash A = A}\ REFL$ $\dfrac{\Gamma \vdash A = B \qquad \Delta \vdash C = D}{\Gamma \cup \Delta \vdash A(C) = B(D)}\ MK\_COMB$ $\dfrac{\Gamma \vdash A = B \qquad \Delta \vdash B = C}{\Gamma \cup \Delta \vdash A = C}\ TRANS$

Dependency

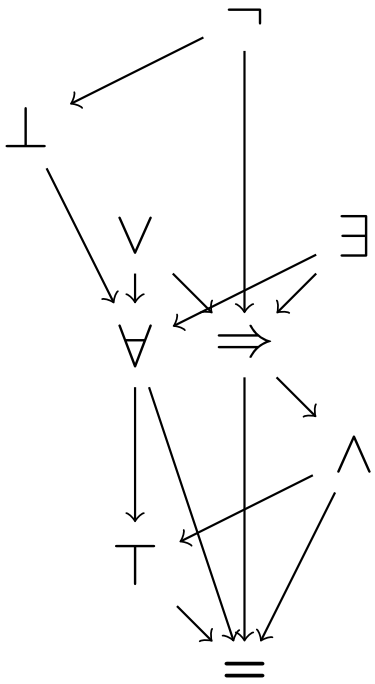

Figure: Constants and Connectives Dependency Analysis

OpenTheory

- HOL family: HOL Light, ProofPower, HOL4, Isabelle ...
- Need a platform to reuse proofs from different systems.
- OpenTheory has a standard format of proofs (*.art).
- Export proofs and import proofs (in article files).
- OpenTheory HOL Light:
  a modified version of HOL Light which allows import and export of proofs.
- `http://www.gilith.com/research/opentheory/`

Packages and Dependency

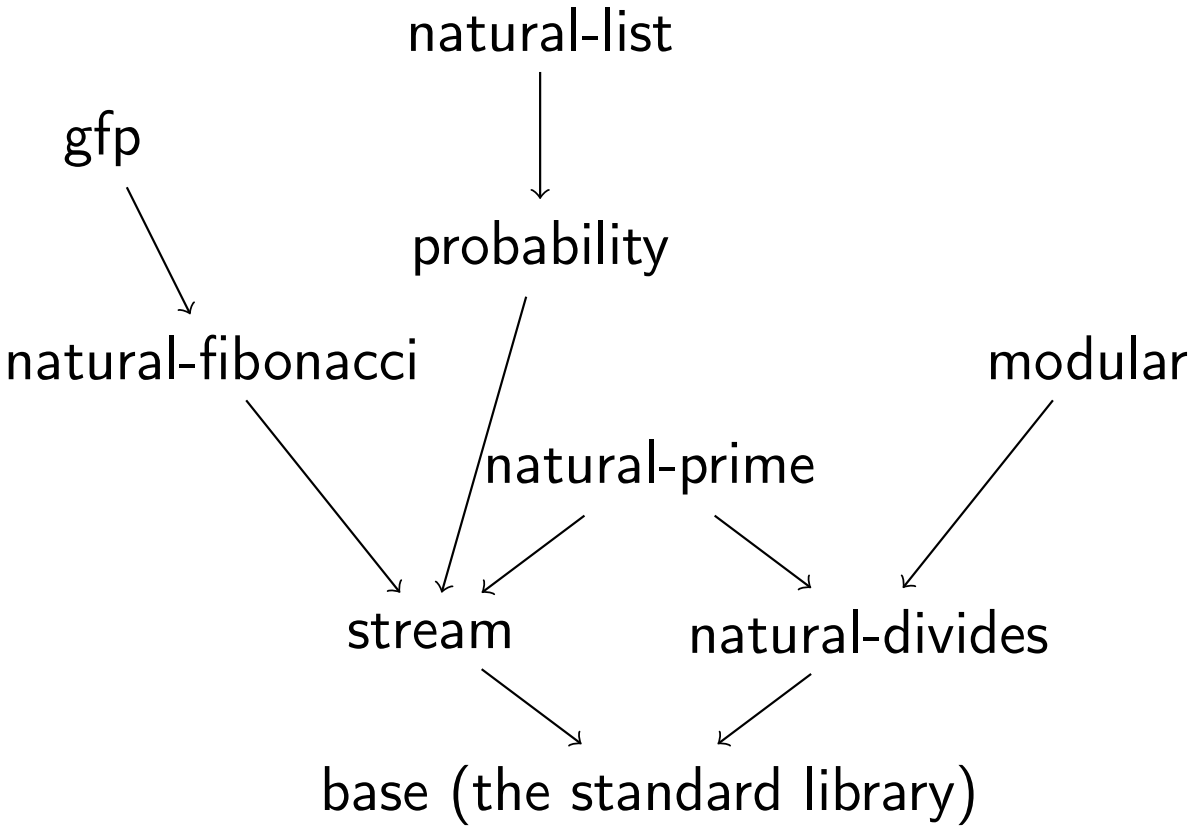

Figure: Dependency of Packages of OpenTheory

OpenTheory Kernel

$$\frac{}{\vdash t = t}\ refl\ t \qquad \frac{}{\{t\} \vdash t}\ assume\ \varphi \qquad \frac{\Delta \vdash \varphi \qquad \Gamma \vdash \varphi = \psi}{\Gamma \cup \Delta \vdash \psi}\ eqMp$$

$$\frac{\Gamma \vdash t = u}{\Gamma \vdash (\lambda v.t) = (\lambda v.u)}\ absThm\ v \qquad \frac{\Gamma \vdash f = g \qquad \Delta \vdash x = y}{\Gamma \cup \Delta \vdash fx = gy}\ appThm$$

$$\frac{\Gamma \vdash \varphi \qquad \Delta \vdash \psi}{(\Gamma \setminus \{\psi\}) \cup (\Delta \setminus \{\varphi\}) \vdash fx = \varphi = \psi}\ deductAntisym \qquad \frac{\Gamma \vdash \varphi}{\Gamma[\sigma] \vdash \varphi[\sigma]}\ subst\sigma$$

$$\frac{}{\vdash (\lambda v.t)u = t[u/v]}\ betaConv((\lambda v.t)u) \qquad \frac{}{c = t}\ defineConstct$$

$$\frac{\vdash \varphi\, t}{\vdash abs(rep\, a) = a \qquad \vdash (rep(abs\, r) = r) = \varphi\, r}\ defineTypeOp\ \ n\ \ abs\ \ rep\ \ vs$$

Figure: Primitive Inference Rules of OpenTheory 5

OpenTheory 5 & 6

Commands	Updated/New
proveHyp	new
trans	new
sym	new
defineConstList	new
hdTl	new
defineTypeOp	updated
pragma	new

Table: Differences between commands of version 5 and version 6

OpenTheory 5 & 6

$$\frac{\Gamma \vdash \varphi \qquad \Delta \vdash \psi}{\Gamma \cup (\Delta \setminus \{\varphi\}) \vdash \psi}\ proveHyp$$

$$\frac{\Gamma \vdash s = t \qquad \Delta \vdash t = u}{\Gamma \cup \Delta \vdash s = u}\ trans$$

$$\frac{\Gamma \vdash \varphi = \psi}{\Gamma \vdash \psi = \varphi}\ sym$$

$$\frac{\vdash \varphi\, t}{\vdash abs(rep\, a) = a \qquad \vdash \lambda r.\varphi\, r = (\lambda r.(rep(abs\, r) = r))}\ \text{defineTypeOp n abs rep vs}$$

Figure: Additional Primitive Inference Rules in OpenTheory 6

News from OpenTheory

OpenTheory is updating constantly, so will Holide and Dedukti.

Today's Topics:

1. Re: package upload failed (Joe Leslie-Hurd)
2. Re: interpretation in separate file (Joe Leslie-Hurd)

Holide and Dedukti

- OpenTheory has a repository of proof packages (articles).
- Holide translates proofs from OpenTheory articles to Dedukti.
- Dedukti[1] is a proof checker (for proof checking).
- from a reviewer: It is not clear from the abstract whether the author has contributed to the development of Holide or Dedukti

Workflow of OpenTheory, Holide and Dedukti

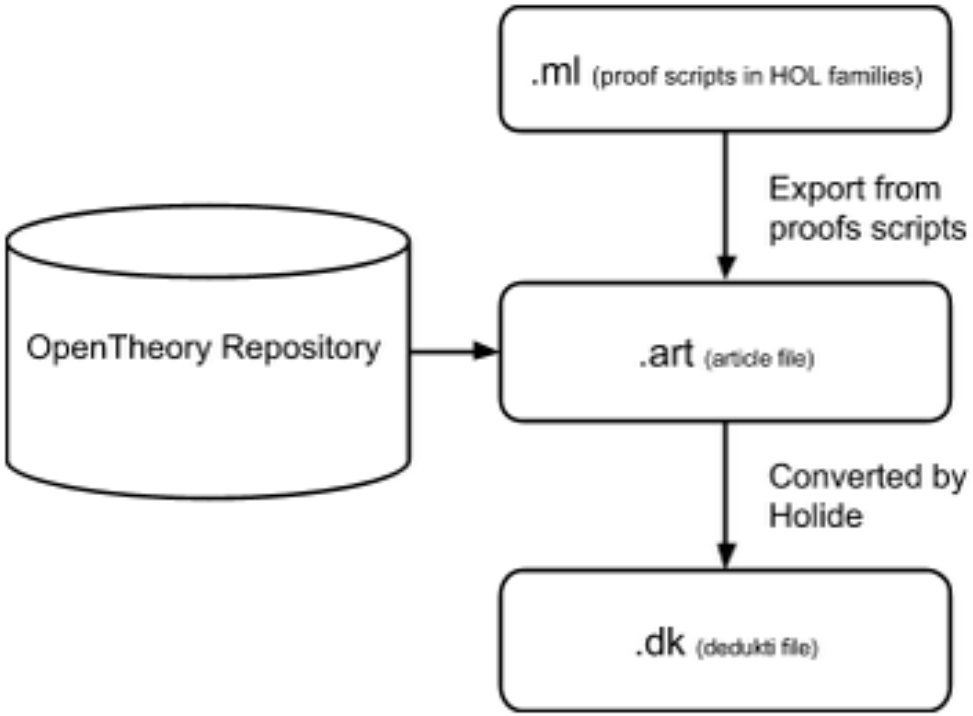

Figure: Work Flow of OpenTheory, Holide and Dedukti

Translation

translate $\left(\dfrac{\Gamma \vdash \varphi = \psi}{\Gamma \vdash \psi = \varphi}\, sym \right) = Sym|A||t_1||t_2|$

translate$\left(\dfrac{\Gamma \vdash s = t \qquad \Delta \vdash t = u}{\Gamma \cup \Delta \vdash s = u}\, trans \right) =$
$Trans|A||x||y||z||D_1||D_2|$, where D_1 is the proof of $s = t$ and D_1 is the proof of $t = u$

translate$\left(\dfrac{\Gamma \vdash \varphi \qquad \Delta \vdash \psi}{\Gamma \cup (\Delta \setminus \{\varphi\}) \vdash \psi}\, proveHyp \right) =$
$ProveHyp|x||y||D_1|(\lambda x : ||\psi||.|D_2|)$, where D_1 is the proof of φ and D_2 is the proof of ψ.

Translation (Continued)

Sym, *Trans* and *ProveHyp* have types as follows:

$$Sym : \Pi\alpha : type.\Pi x, y : term\,\alpha.\ proof\,(eq\ bool\ x\ y) \rightarrow proof\,(eq\ bool\ y\ x)$$

$$Trans : \Pi\alpha : type.\Pi x, y, z : term\,\alpha\ proof\,(eq\,\alpha\ x\ y) \rightarrow proof\,(eq\,\alpha\ y\ z) \rightarrow proof\,(eq\,\alpha\ x\ z)$$

$$ProveHyp : \Pi x, y : term\ bool.proof\ x \rightarrow (proof\ x \rightarrow proof\ y) \rightarrow proof\ y$$

Benchmarks: Translation

Package	Holide 1		Holide 2	
	Size (KB)	Time (s)	Size (KB)	Time (s)
base	1,436	19.35	1,194	19.42
cl	313	5.77	313	5.56
empty	0	0.20	0	0.00
gfp	136	1.42	112	1.35
lazy-list	1,390	31.43	1,391	31.78
modular	45	1.13	37	0.37
natural-bits	162	1.43	132	1.39
natural-divides	193	2.10	157	1.94
natural-fibonacci	130	1.31	108	1.24
natural-prime	140	1.46	116	1.34
parser	240	3.22	204	3.15
probability	26	0.30	23	0.23
stream	75	0.75	63	0.73
word10	86	0.76	71	0.62
word12	88	0.79	72	0.75
word16	131	1.60	107	0.77
word5	77	0.70	64	1.56
Total	**4,668**	**73.73**	**4,377**	**72.21**

Figure: The Size of Article Files and Translation Time

Benchmarks: Checking

Package	Dedukti (Holide 1)		Dedukti (Holide 2)	
	Size (KB)	Time (s)	Size (KB)	Time (s)
base	4,681	10.63	4,440	9.74
cl	1,219	2.42	1,219	2.46
empty	0	0.00	0	0.00
gfp	400	0.73	375	0.65
lazy-list	5,718	13.31	5,717	13.11
modular	120	0.19	111	0.17
natural-bits	452	0.74	419	0.68
natural-divides	599	1.11	566	0.99
natural-fibonacci	378	0.67	354	0.60
natural-prime	408	0.72	388	0.65
parser	802	1.87	776	1.69
probability	72	0.12	69	0.11
stream	221	0.41	211	0.38
word10	234	0.38	216	0.29
word12	239	0.40	220	0.35
word16	396	0.80	364	0.36
word5	207	0.33	192	0.72
Total	**16,146**	**34.83**	**15,637**	**32.95**

Figure: Th Size of Article Files and Proof Checking Time

Evaluation

- Fully verified all the libraries in OpenTheory.
- Little difference between version 5 and version 6.
- The size of proof articles got reduced by around 7%. The proof checking time reduced by around 5%.
- These benchmarks were generated on a 64-bit Intel Core i5-4590 CPU @3.30GHz $\times$4 PC with 3.8GB RAM.

Kepler Conjecture

- 20 years to complete
- The formal proof of Kepler Conjecture takes about 5000 hours to check (in HOL Light) in 2014.
- What if we check by Dedukti?

ProofCloud

1. A Proof Retrievel Engine: `http://airobert.github.io/proofcloud/`
2. 1700+ pages of proofs with analysis.
3. A representation of proof checking results by Holide and Dedukti.
4. Which proofs are constructive?

Version 2

Proof Search Engine which represents the analysis and proof checking results.

Figure: Index Page of ProofCloud (version 1)

It's version 2 now!!!

Demo

- So far there are 6 packages: base, stream, probability, natural-bits, natural-divides and natural-prime.
- For the natural-prime package: `http://airobert.github.io/proofcloud/natural-prime.html`
- ADD_EQ_0
- $!mn.m + n = 0 \Leftrightarrow m = 0 \wedge n = 0$

Structural Analysis

the combination of the *subst* and *eqmp* rule takes over 45% of all the inferences rules.

Inference Rules	**Count**
subst	93667
eqmp	92617
appthm	53155
proveHyp	47728
betaConv	21485
absThm	15096
...	...
assume	16986
Overall	**413207**

Structural Results

The frequency of subst and eqmp combined is over 45%

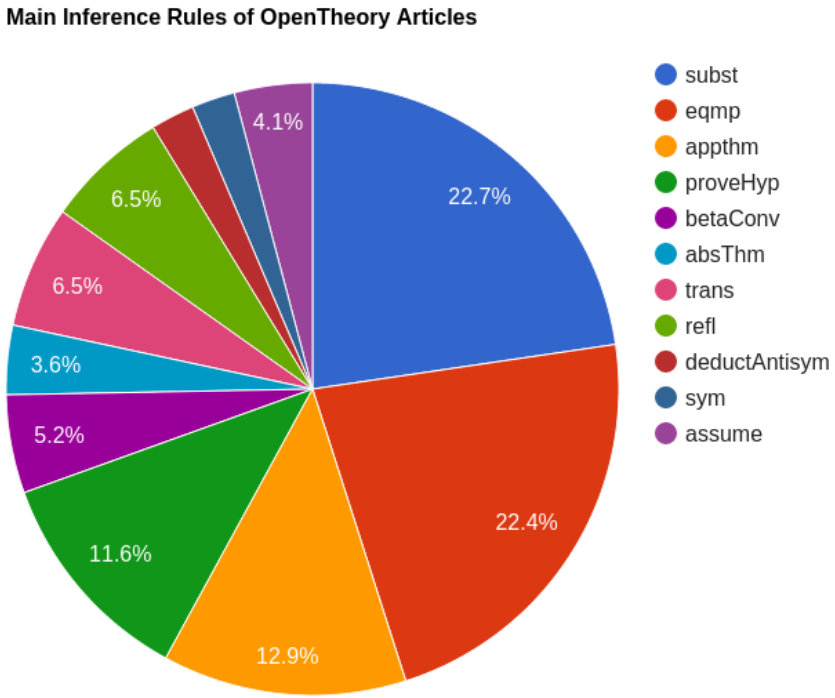

Figure: Frequency of Main Inference Rules of OpenTheory Articles

And more for each individual theorem or combine with Machine Learning?

Statistical Results

1209 proofs in the standard library.
541 constructive proofs v.s. 668 classical proofs
44.75% of them constructive proofs.

In contrast, the *natural-divides* package has only 10 constructive proofs out of 136 proofs, making only 7.35% of them constructive.

Next, these 668 proofs to their constructive form?

ProofCloud 3?

- More OpenTheory packages and even Coq, Mizar ... libraries?
- Better GUI
- Better searching accuracy
- More text description of theorems (at least a proper name)
- A proof analysis engine combined with machine learning as a feature suggestion engine after comparing thousands of proofs?

Conclusion and Acknowledgement

- An overview of recent updates of OpenTheory, Holide, Dedukti, ProofCloud
- Proof checking benchmarks
- Thanks to Dr. Gilles Dowek, Dr. Ali Assaf, Dr. Joe Hurd, Mr. Frédéric Gilbert and Mr. Nigel Sham and Mr. Dianlin Shen.

Thank you!

I am looking for a master thesis supervisor and opportunities for doctoral studies related but not limited to those provers!