HOL vs ALF

Sten Agerholm, Ilya Beylin, Peter Dybjer

August 26, 1996

"actually, this talk has no theoretical interest ..."

HOL vs ALF

HOL = Higher Order Logic: implementation of Church simple type theory + ML-polymorphism (Gordon).

full-scale framework for classical mathematics

ALF = A Logical Framework: an implementation of a basic framework for dependent types; support for inductive definitions and function definition by pattern matching; hence supports Martin-Löf type theory. (Coquand, Magnusson, Nordlander, Nordström, ... 1991, ...)

full-scale framework for constructive mathematics and simultaneously a functional programming language

The logics

both typed λ-calculus based, but:

HOL	ALF
simple types	dependent types
external logic	integrated logic
classical logic	intuitionistic logic
impredicative higher-order logic	predicative higher-order logic
closed (safe) system	open system
inductive definition package	primitive inductive definitions
extensional equality	intensional equality
ML-polymorphic	monomorphic

both have naive set-theoretic semantics!

The systems

HOL:

- tactics in ML

ALF:

- proving = programming by defining new inductive datatypes and recursive functions. Like functional programming but with dependent types and using only terminating “structural” recursion (“strong” functional programming). Built in normalization during type-checking.

- explicit representation of proof term on the screeen; proof by pointing and clicking.

Show an example of an ALF-screen

"actually, this problem has only theoretical interest ..."

Coherence for monoidal categories (Mac Lane 1963)

more generally: formalization of category theory (cf Huet and Saibi 1995: Constructive Category Theory (in Coq))

Coherence problems

A *monoidal category* is a category where the objects form a monoid *up to isomorphism*. This means that there are arrows (natural isos)

$$\begin{aligned} \alpha_{a,b,c} &: a \otimes (b \otimes c) \longrightarrow (a \otimes b) \otimes c \\ \lambda_a &: e \otimes a \longrightarrow a \\ \rho_a &: a \otimes e \longrightarrow a \end{aligned}$$

Question: Under what conditions are two "canonical" arrows from a to b equal? A canonical arrow is built up by the operations of a monoidal category starting from α, λ, ρ and witnesses "equality" of objects?

Fundamental question for Martin-Löf type theory: are all proofs of an equality equal?

$$p, p' \in I(A, a, b) \;\rightarrow\; p = p'?$$

The solution

Use proof = program = arrow:

1. **Normalization in free monoid. (Flatten binary trees to list!)**

2. **Proof objects witnessing normalization.**

3. **These come out as arrows in a free monoidal category.**

4. **Check equalities of these arrows: induction + diagram-chasing.**

ALF: free monoid can be used for free monoidal category.

HOL: reimplement free monoidal category.

Binary words

HOL: inductive datatype package:

```
bw = e | Var of X | Ox of bw => bw
```

ALF: inductive definition of "set":

$$\begin{aligned} bw \;\in\; & Set \\ & e \in bw \\ & Var \in (x \in X)bw \\ & \otimes \in (a, b \in bw)bw \end{aligned}$$

Equality of binary words

HOL: inductive relation definition package

```
a Ox (b Ox c) cbw (a Ox b) Ox c,
e Ox a cbw a,  a Ox e cbw a, ...
```

ALF: inductive definition of dependent set

$$cbw \in (a, b \in bw)Set$$

$$\alpha \in (a, b, c \in bw)cbw(\otimes(a, \otimes(b, c)), \otimes(\otimes(a, b), c))$$
$$\lambda \in (a \in bw)cbw(\otimes(e, a), a)$$
$$\rho \in (a \in bw)cbw(\otimes(a, e), a)$$
$$\vdots$$

The word problem for monoids

HOL:

```
|- !a b. a cbw b = (Nf a = Nf b),
```

follows from

```
|- !a b. a cbw b ==> (Nf a = Nf b),
|- !a. a cbw (Nf a)
```

ALF:

$$nf \in (a, b \in bw; f \in cbw(a, b))I(Nf(a), Nf(b))$$
$$\nu \in (a \in bw)cbw(a, Nf(a))$$

Formalization of the free monoidal category

E-category is like category but there is an explicit equivalence relation on arrows. (Non-standard notion of category)

ALF:

- bw **is the set of objects**
- $cbw(a, b)$ **is the set of arrows from** a **to** b
- $==\in (f, g \in Hom(a, b))Set$ **is inductively defined dependent set**

HOL:

- `bw` **is the type of objects**
- `arr` **is a new inductive datatype of raw arrows**
- `dom:arr->bw` **and** `cod:arr->bw` **are the source and target functions.**
- `==:arr->arr->bool` **inductively defined relation**

The coherence theorem

ALF

$$coherence \in (a, b \in bw; f, g \in cbw(a, b)) == (f, g)$$

Follows from ν is a natural iso:

$$\begin{array}{ccc} a & \xrightarrow{f} & b \\ {\scriptstyle \nu_a}\downarrow & & \downarrow{\scriptstyle \nu_b} \\ N\,f\,a & \xrightarrow[id]{} & N\,f\,b \end{array}$$

Proof by induction on f (essentially). Each case proved by diagram-chasing.

HOL support for diagram chasing

- **use a theorem stating that two arrows are equal as a rewrite rule (cf Paulson 1983)**

- **congruence, transitivity, reflexivity performed automatically**

- **associative rewriting (first move parentheses right)**

- **side conditions using** `dom` **and** `cod` **proved automatically**

As a result the HOL-proof does not mention many of the things subsumed by the informal diagram notation (there are minor exceptions).

“Metacoherence” in ALF

We have

$$nf \in (a, b \in bw)I(Nf(a), Nf(b))$$

in ALF.

Unfortunately this doesn't mean that we can substitute $Nf(a)$ for $Nf(b)$ everywhere. Hence

$$id \in (a \in bw)cbw(Nf(a), Nf(b))$$

does not type-check!

Instead we have to reason about “identity” arrows which depend on the proof nf that two objects are equal, and use that such proofs are unique.

Conclusion

Classical reasoning played no role.

The treatment of equality in HOL simplified matters

- **theoretically: extensional equality and substitutivity of equality**

- **practically: ML-tool for diagram chasing**

Potential advantages of ALF much less significant for this case

- **primitive inductive definitions and dependent types**

- **built-in proof-normalization**