FM-HOL

Murdoch J. Gabbay, mjg1003@cl.cam.ac.uk
Computer Laboratory, Cambridge University, UK
April 2002

Binding

FM ="Frankel-Mostowski". Just a label. 60 ago developed a set theory which turned out to underlie the original FM work ([Gabbay-Pitts LICS'99]). Named in their honour.

FM-HOL is a Higher-Order Logic (HOL) for implementing and manipulating syntax-with-binding at object level. I.e. build datatypes of syntax, prove theorems on them.

For example?

Binding

Automath uses De Bruijn indeces:

(1) $$\Lambda \stackrel{\text{def}}{=} \textbf{Var of } \mathbb{N} + \textbf{App of } \Lambda \times \Lambda + \textbf{Lam of } \Lambda$$

Good: α-equivalence is literal equality, but

Less good: inductive principle derived from this is 'twisted'.

(2) $$\begin{aligned} &\big(\forall i.\ \phi(\textbf{Var}(i)) && \wedge \\ &\ \forall t_1, t_2.\ \phi(t_1) \wedge \phi(t_2) \rightarrow \phi(\textbf{App}(t_1, t_2)) && \wedge \\ &\ \forall t.\ \phi(t) \rightarrow \phi(\textbf{Lam}(t))\big) \\ &\ \rightarrow \forall t.\ \phi(t). \end{aligned}$$

Binding

Representation should re¤ect understanding.

Should observe that there are many forms of understanding: theoretical, implementational, ergonomical. Many forms of representation: De Bruijn, HOAS, na˝ve representation, sheaves, Isabelle terms. . . now FM.

Current development weighted towards theoretical/ergonomic. Implementation, “computationally ef£cient” FM, partly the object of FreshML research. Discussed here is the theoretical/ergonomic side of FM packaged in **FM-HOL**.

Theoretical/Ergonomical: $\mathbb{A}$, Tran

What is there to support? What do we do with syntax?

- Variable names. Call them ***atoms***, **FM-HOL** has type $\mathbb{A}$.
- Rename variable names. In terms t, but also predicates/proofs P. E.g. de facto we assume if $P(x)$ then $P(y)$ for x, y 'suitably fresh' variable names.

FM has constant symbol ***transposition***:

$$\textbf{Tran} : \mathbb{A} \to \mathbb{A} \to \alpha \to \alpha$$

written $(a\ b).t$, $(a\ b).P$. Structure of these elements?

Equivariance

$(a\ b).a = b$, $(a\ b).b = a$, $(a\ b).n = n$ for $n \neq a, b$.

Tran is ***equivariant***:

$$\begin{aligned}(a\ b).f(x) &= ((a\ b).f)((a\ b).x)\\ (a\ b).c &= c \qquad c\ \textbf{closed}.\end{aligned}$$

$$\begin{aligned}(a\ b).f(x,y,z,\ldots) &= f((a\ b).x,(a\ b).y,(a\ b).z,\ldots)\\ (a\ b).\textbf{Inl}(\textbf{Inr}(b)) &= \textbf{Inl}(\textbf{Inr}(a))\\ (a\ b).Proof_{x,y}(b) &\iff Proof_{(a\ b).x,(a\ b).y}(a)\end{aligned}$$

Application of Tran

Instead of $tm[z/y]$ use $(z\ y).tm$. Of course if z *and* y free in tm we need inductively de£ned $[z/y]$ function as before (e.g. π-calculus input $xy.P \stackrel{xz}{\rightarrow} P[z/y]$). In case where z is chosen 'fresh', the two coincide. Compare:

$$\textbf{Lam}(a.s) =_\alpha \textbf{Lam}(b.t) \leftarrow [c/a]s =_\alpha [c/b]t \quad c \notin \textbf{FV}(s), \textbf{FV}(t)$$

$$\textbf{Lam}(a.s) =_\alpha \textbf{Lam}(b.t) \leftarrow И c.\ (c\ a).s =_\alpha (c\ b).t$$

Being tied to **FV** pernicious & unnecessary.

Freshness

Every $X \subseteq \mathbb{A}$ (HOL sets) either ‘small’ or ‘large’:

$$\mathcal{P}(\mathbb{A}) = \mathsf{S} + \mathsf{И} \qquad \mathsf{S} = \left\{ \mathbb{A} \setminus L \;\middle|\; L \in \mathsf{И} \right\}$$

- Has a notion of ‘for fresh a’ absent in usual theories (“$c \notin FV(s)$”). De£ne quanti£er:

$$\mathsf{И}a.\, P(a) \stackrel{\text{def}}{=} P \in \mathsf{И}$$

Think ‘***small=£nite***’, ‘***large=co£nite***’. Recall previous slide.

Abstractions

И commutes with $\wedge$, $\vee$, $\neg$, $\rightarrow$. E.g.

$$И a.\ P(a) \wedge Q(a) \iff И a.\ P(a) \wedge И a.\ Q(a).$$

De facto assumption in practice anyway. Useful for automation. Actual de£nition of ‘small’ £nely crafted.

Using И and **Tran** construct ***abstraction types***:

$$x_* = a.x : \mathbb{A}.\alpha \qquad\qquad a.a = b.b$$

“$<a, x>$ with a bound”, also “function takes fresh a returns x_* concreted at a”. $\mathbb{A}.-$ functor with left & right adjoints. Commutes with $+$, $\times$, $\rightarrow$, 0, 1.

Syntax-with-binding

Inductive representations of syntax with bound variables,

$$\Lambda \stackrel{\text{def}}{=} \mathbb{A} + \Lambda \times \Lambda + \mathbb{A}.\Lambda,$$

$$\Pi \stackrel{\text{def}}{=} 1 + \Pi \times \Pi + \mathbb{A} \times \mathbb{A} \times \Pi + \mathbb{A} \times \mathbb{A}.\Pi + \mathbb{A}.\Pi.$$

Also ‘exotic’ entities: lists, trees, graphs. Exotic abstraction, $\mathbb{A}\text{-}lst.\alpha$. Analysis of binding ‘in nature’ my current excitement. Make relevant, implementable. Make abstractions and more sophisticated descendents ¤y. I seek proofs and defs in trouble because of binding. Seemingly no great shortage.

Really, I mean it!

(3) $$\frac{P_1 \stackrel{\overline{x}(y)}{\rightarrow} P_1' \quad P_2 \stackrel{xy}{\rightarrow} P_2'}{P_1 \mid P_2 \stackrel{\tau}{\rightarrow} \nu y.(P_1' \mid P_2')} \quad y \notin fn(P_1), fn(P_2)$$

becomes using simple convention

(4) $$\frac{P_1 \stackrel{\overline{x}(y)}{\rightarrow} P_1' \quad P_2 \stackrel{xy}{\rightarrow} P_2'}{P_1 \mid P_2 \stackrel{\tau}{\rightarrow} \nu y.(P_1' \mid P_2')}.$$

Note $y \notin fn(P_1)$.

Support for FM

- Theoreticians: **FM-HOL**, FM sets, others. Roll your own, e.g. Nominal Logic. И already in use outside immediate FM group. Abstraction types not yet so much, should be.
- Programmers: FreshML project. Augment ML with И-like constructor, need simple, decidable, useful, type system to prevent ‘escape’.
- Uni£cation: Current joint research with Pitts & Urban.
- Dependent types: See **FM-HOL** as £rst step.
- Automath & sisters: Isabelle/HOL/FM. Clunky?

FM-HOL in this paper

- Theoreticians: FM Universe
- Myself: Understand structure of theorems better, properties needed of S. Encoded in this paper my understanding (as of 2 months ago) of what FM*is* and how the de£nitions £t together. Ideological/artistic self-expression.
- Dependent types: With this understanding, build dependent type theory.
- Automath & sisters: Verify **FM-HOL** models abstractions in nature, then implement it. Isabelle/**FM-HOL**, slick!