Fresh Logic

Murdoch J. Gabbay
Work with James Cheney
Also including work with Lucian Wischik in the Appendix

May 14, 2004

Motivation

We want to specify, and reason about abstract syntax with variable symbols, and its operational behaviour (and its denotation too if we're feeling brave!).

Fresh Logic is First-Order Logic (with equality) enriched with, a sort of atoms $\mathtt{A}$, a swapping term-former $\mathtt{swap}_\tau : \mathtt{A} \longrightarrow \mathtt{A} \longrightarrow \tau \longrightarrow \tau$, a freshness predicate symbol $\# : \mathtt{A} \times \tau$, and a И-quantifier quantifying over variables of sort $\mathtt{A}$.

We use И to create fresh atoms, $\mathtt{swap}$ to rename 'stale' atoms, and $\#$ to say when a 'stale' atom is actually fresh.

That's it. We write $\#$ infix and write $\mathtt{swap} x y t$ as $(x\ y)t$.

Examples

1. $x_{\mathtt{A}}$ an atom.
2. $\vdash (x\ y)x = y$ a derivable equality judgement.
3. $x\#x \vdash \bot$ a derivable judgement.
4. $x\#y \vdash x\#y$ another derivable judgement.
5. $\vdash \text{И}x.\, x = x$ ditto.
6. $\vdash \text{И}x.\, x\#z$ you guessed it.
7. $\vdash (x\ y)z = x$ not derivable; we don't use atoms-as-constants (aac) but atoms-as-variables (aav).

Motivation

Terms are defined by the following grammar:

$$s, t, a, b ::= x \mid \mathsf{c} \mid \lambda x.t \mid tt'$$

Here c are constructors, for example $\mathtt{swap}$, or perhaps $\langle\text{-},\text{-}\rangle$ for pairing; the sort system puts terms in the right type, variables x are assumed sorted à la Church.

Predicates are defined by the following grammar:

$$P, Q, R ::= p(ts) \mid P \wedge P \mid P \vee P \mid P \supset P \mid \\ \top \mid \bot \mid \forall x.\, P \mid \exists x.\, P \mid \mathsf{И} x.\, P$$

Equality $=$ and freshness $\#$ are predicate constant symbols (the ps).

Deduction Rules i

$$\frac{}{\Gamma, P \vdash P}(Ax) \qquad \frac{}{\Gamma, \bot \vdash C}(\bot L) \qquad \frac{}{\Gamma \vdash \top}(\top R)$$

$$\frac{\Gamma \vdash P \, \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}(\wedge R) \qquad \frac{\Gamma, P, Q \vdash C}{\Gamma, P \wedge Q \vdash C}(\wedge L)$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q}(\vee R_1) \qquad \frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q}(\vee R_2) \qquad \frac{\Gamma, P \vdash C \quad \Gamma, Q \vdash C}{\Gamma, P \vee Q \vdash C}(\vee L$$

Deduction Rules ii

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}(\supset R) \qquad \frac{\Gamma \vdash P \quad \Gamma, Q \vdash C}{\Gamma, P \Rightarrow Q \vdash C}(\supset L)$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash \forall x.\, P}(\forall R) \qquad \frac{\Gamma, P\{t/x\} \vdash C}{\Gamma, \forall x.\, P \vdash C}(\forall L)$$

$$\frac{\Gamma \vdash P\{t/x\}}{\Gamma \vdash \exists x.\, P}(\exists R) \qquad \frac{\Gamma, P \vdash C}{\Gamma, \exists x.\, P \vdash C}(\exists L)$$

Deduction Rules iii

$$\frac{\Gamma \vdash P \quad \Gamma, P \vdash Q}{\Gamma \vdash Q}(Cut) \qquad \frac{\Gamma, P, P \vdash C}{\Gamma, P \vdash C}(Ctrct)$$

$$\frac{\Gamma, t = t \vdash C}{\Gamma \vdash C}(=Ref) \qquad \frac{\Gamma, t' = t, P\{t'/x\} \vdash C}{\Gamma, t' = t, P\{t/x\} \vdash C}(=Sub)$$

Deduction Rules iv

$$\frac{\Gamma, a\#ts \;\vdash\; P\{a/n\}}{\Gamma, a\#ts \;\vdash\; \text{И}n.\, P}\,(\text{И}R) \quad (P \equiv P'[n, ts])$$

$$\frac{\Gamma, a\#ts, P\{a/n\} \;\vdash\; C}{\Gamma, a\#ts, \text{И}n.\, P \;\vdash\; C}\,(\text{И}L) \quad (P \equiv P'[n, ts])$$

Deduction Rules v

$$\frac{\Gamma, n\#ts \vdash C}{\Gamma \vdash C}(new\mathbb{A}) \quad (n \notin V(\Gamma, C, ts))$$

$$\frac{\Gamma, a\#b \vdash C \quad \Gamma, a = b \vdash C}{\Gamma \vdash C}(case\mathbb{A}) \qquad \frac{}{\Gamma, a\#a \vdash C}(\#\mathbb{A})$$

$$\frac{\Gamma, (a\ b)\cdot t = t \vdash P}{\Gamma,\ a\#t, b\#t \vdash P}(\pi\#) \qquad \frac{\Gamma, P \vdash C}{\Gamma, (a\ b)\cdot P \vdash C}(\pi L)$$

Deduction Rules vi

$$\frac{\Gamma, A \ \vdash \ C}{\Gamma \ \vdash \ C}\,(\mathcal{A}L) \quad (A \in \mathcal{A})$$

$$\mathcal{A} = \left\{ \begin{array}{l} (a\,a)\cdot t = t, \qquad (a\,b)\cdot(a\,b)\cdot t = t \\ (a\,b)\cdot a = b, \quad (a\,b)\cdot \mathsf{c} = \mathsf{c}, \\ (a\,b)\cdot[t\,u] = [(a\,b)\cdot t]\,[(a\,b)\cdot u], \\ (a\,b)\cdot \lambda x.t = \lambda x.(a\,b)\cdot[t\{(a\,b)\cdot x/x\}] \end{array} \right\}$$

Example deductions $(D1)$ and $(D2)$

$$(D1) \quad \dfrac{\dfrac{\overline{n \# x \vdash n \# x}\,(Ax)}{n \# x \vdash \text{И}n.\, n \# x}\,(\text{И}R)}{\vdash \text{И}n.\, n \# x}\,(new\mathbb{A}).$$

$$(D2) \quad \dfrac{\dfrac{\dfrac{\dfrac{\overline{n \# x, m \# x, n \# m \vdash x = x}\,(=Ref), (Ax)}{n \# x, m \# x, n \# m \vdash (n\ m) \cdot x = x}\,(\pi L)}{n \# x, m \# x, n \# m \vdash \text{И}m.\, (n\ m) \cdot x = x}\,(\text{И}R)}{n \# x, m \# x, n \# m \vdash \text{И}n.\, \text{И}m.\, (n\ m) \cdot x = x}\,(\text{И}R)}{\vdash \text{И}n.\, \text{И}m.\, (n\ m) \cdot x = x}\,(new\mathbb{A}), (new\mathbb{A}).$$

Interlude: Slices

A slice of P over n is a tuple (P, n, ys, P', ts) of P, a variable symbol n, and:

1. Variable symbols $y_1, \ldots, y_k$ which we write ys, not appearing in P
2. A proposition P' with $V(P') = \{n, y_1, \ldots, y_k\}$.
3. Terms $t_1, \ldots, t_k$ which we write ts, such that $n \notin \bigcup_1^k V(t_i)$ and $P'\{t_1/y_1\} \ldots \{t_k/y_k\} \equiv P$.

Slices (examples)

1. $p(f(x,n),m)$ sliced over n is
   $(p(f(x,n),m), n, (y_1, y_2), p(f(y_1,n),y_2), (x,m))$.

2. $p(f(x,n),m)$ sliced over m is
   $(p(f(x,n),m), m, (y_1), p(y_1,m), f(x,n))$.

There is a natural notion of minimal slice $P \equiv P'[n,ts]$, the (unique up to renaming the ys) slice such that P' is as small and the ts are as large as possible. Both slices above are minimal.

Lemma: If $P \equiv P'[n,ts]$ then for any term s,
$P\{s/n\} \equiv P'\{s/n\}\{ts/ys\}$. Also, for any n and s such that
$n \notin V(s)$, $P\{s/x\} \equiv P'[n, ts\{s/x\}]$.

Thus a substitution for n in P does not affect the ts, and minimality of slices over n is not affected by substitutions that do not introduce free occurrences of n.

Example deductions $(D3)$

$$
(D3)\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,(Ax) \quad \dfrac{}{B \vdash B}\,(Ax)}{n \mathrel{\#} V(A,B),\ A, B \vdash A \wedge B}\,(\wedge R)}{n \mathrel{\#} V(A,B),\ A, \mathsf{N}n.\,B \vdash A \wedge B}\,(\mathsf{N}L)}{n \mathrel{\#} V(A,B),\ \mathsf{N}n.\,A, \mathsf{N}n.\,B \vdash A \wedge B}\,(\mathsf{N}L)}{n \mathrel{\#} V(A,B),\ \mathsf{N}n.\,A, \mathsf{N}n.\,B \vdash A \wedge B}\,(\wedge L)}{n \mathrel{\#} V(A,B),\ \mathsf{N}n.\,A, \mathsf{N}n.\,B \vdash \mathsf{N}n.\,(A \wedge B)}\,(\mathsf{N}R)}{\mathsf{N}n.\,A, \mathsf{N}n.\,B \vdash \mathsf{N}n.\,(A \wedge B)}\,(new\mathbb{A})}{\mathsf{N}n.\,A \wedge \mathsf{N}n.\,B \vdash \mathsf{N}n.\,(A \wedge B)}\,(\wedge L).
$$

Example deductions $(D4)$ and $(D5)$

$$
(D4)\qquad
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{n \mathrel{\#} x \;\vdash\; n \mathrel{\#} x}\,(Ax)
        \qquad
        \dfrac{}{P \;\vdash\; P}\,(Ax)
      }{n \mathrel{\#} x \Rightarrow P,\; n \mathrel{\#} x, vs \;\vdash\; P}\,(\supset L)
    }{\text{И}n.\,\forall x.\, n \mathrel{\#} x \Rightarrow P,\; n \mathrel{\#} x, vs \;\vdash\; P}\,(\text{И}L), (\forall L)
  }{\text{И}n.\,\forall x.\, n \mathrel{\#} x \Rightarrow P \;\vdash\; \text{И}n.\,P}\,(new\mathbb{A}), (\text{И}R)
}{\text{И}n.\,\forall x.\, n \mathrel{\#} x \Rightarrow P \;\vdash\; \forall x.\,\text{И}n.\,P}\,(\forall R)
$$

$$
(D5)\qquad
\dfrac{
  \dfrac{
    \dfrac{}{(n\ a)\cdot x = x \;\vdash\; (n\ a)\cdot x = x}\,(Ax)
  }{a \mathrel{\#} x, n \mathrel{\#} x \;\vdash\; (n\ a)\cdot x = x}\,(\pi\#)
}{a \mathrel{\#} x \;\vdash\; \text{И}n.\,(n\ a)\cdot x = x}\,(\text{И}R), (new\mathbb{A})
$$

Example deductions $(D6)$

$$
(D6) \qquad
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{a \# x \vdash a \# x}\,(Ax)}{(n\;a)\cdot a \# (n\;a)\cdot(n\;a)\cdot x \vdash a \# x}\,(\leadsto)}{n \# (n\;a)\cdot x \vdash a \# x}\,(\pi L)}{n \# x, (n\;a)\cdot x = x \vdash a \# x}\,({=}Sub)}{\cfrac{n \# x, a, \reflectbox{N} n.\,(n\;a)\cdot x = x \vdash a \# x}{\reflectbox{N} n.\,(n\;a)\cdot x = x \vdash a \# x}\,(new\mathbb{A})}\,(\reflectbox{N} L)
$$

Uniform Derivation (Uniform Proof)

Logic programming can be viewed as a form of uniform derivation. A derivation is uniform if for all subderivations of judgements whose conclusion is not atomic ($\Gamma \vdash p(ts)$ is not such), the final rule is a right-rule or $(new\mathbb{A})$.

A logic programming language based on Fresh Logic is a subset of the judgements such that an element of that subset is derivable if and only if it has a uniform derivation.

Uniform logic programming langauge

For example a hereditarily Horn clause language

$$
\begin{array}{rcl}
G & ::= & \top \mid p(ts) \mid G \wedge G \mid \exists x.G \mid \text{И} n.G \\
D & ::= & \top \mid p(ts) \mid D \wedge D \mid G \supset D \\
 & \mid & \forall x.D \mid \text{И} n.D
\end{array}
$$

and a hereditarily Harrop clause language

$$
\begin{array}{rcl}
G & ::= & \top \mid p(ts) \mid G \wedge G \mid \exists x.G \mid \text{И} n.G \\
 & \mid & \forall x.G \mid G \vee G \mid D \supset G \\
D & ::= & \top \mid p(ts) \mid D \wedge D \mid G \supset D \\
 & \mid & \forall x.D \mid \text{И} n.D
\end{array}
$$

$$\frac{(\Sigma, h); \Gamma \;\vdash\; \sigma \triangleright A[h\sigma/x]}{\Sigma : \Gamma \;\vdash\; \sigma \triangleright \forall x.A}\;(\forall R)\quad (h \notin \Sigma)$$

$$\frac{\Sigma, \sigma \;\vdash\; t : \tau \quad \Sigma : \Gamma, \sigma \triangleright A[t/x] \;\vdash\; C}{\Sigma : \Gamma, \sigma \triangleright \forall x{:}\tau.A \;\vdash\; C}\;(\forall L)$$

$$\frac{\Sigma, \sigma \;\vdash\; t : \tau \quad \Sigma : \Gamma \;\vdash\; \sigma \triangleright A[t/x]}{\Sigma : \Gamma \;\vdash\; \sigma \triangleright \exists x{:}\tau.A}\;(\exists R)$$

$$\frac{(\Sigma, h); \Gamma, \sigma \triangleright A[h\sigma/x] \;\vdash\; C}{\Sigma : \Gamma, \sigma \triangleright \exists x.A \;\vdash\; C}\;(\exists L)\quad (h \notin \Sigma)$$

$$\frac{\Sigma : \Gamma \;\vdash\; (\sigma, y) \triangleright A[y/x]}{\Sigma : \Gamma \;\vdash\; \sigma \triangleright \nabla x.A}\;(\nabla R)\quad (y \notin \sigma)$$

$$\frac{\Sigma : \Gamma, (\sigma, y) \triangleright A[y/x] \;\vdash\; C}{\Sigma : \Gamma, \sigma \triangleright \nabla x.A \;\vdash\; C}\;(\nabla L)\quad (y \notin \sigma)$$

Translation into Fresh Logic

We enrich the signature with constants $\mathtt{n}_\tau : \mathtt{A}{\longrightarrow}\tau$ for each τ, and we write $ev(h)$ for $\forall n.\ n\#h$.

$$
\begin{array}{rcl}
[\![t]\!]_\sigma & = & t \\
[\![P \otimes Q]\!]_\sigma & = & [\![P]\!]_\sigma \otimes [\![Q]\!]_\sigma \quad (\otimes \in \{\wedge, \vee, \supset\}) \\
[\![\forall x{:}\tau.P]\!]_\sigma & = & \forall h{:}\tau_\sigma{\longrightarrow}\tau.ev(h) \supset [\![P]\!]_\sigma\{h\sigma/x\} \\
[\![\exists x{:}\tau.P]\!]_\sigma & = & \exists h{:}\tau_\sigma{\longrightarrow}\tau.ev(h) \wedge [\![P]\!]_\sigma\{h\sigma/x\} \\
[\![\nabla x{:}\tau.P]\!]_\sigma & = & \text{И} x{:}\mathtt{A}.[\![P]\!]_{(\sigma,x)}\{\mathtt{n}_\tau x/x\} \\
[\![\sigma \triangleright P]\!] & = & \text{И} \sigma{:}\mathtt{A}.[\![P]\!]_\sigma\{\mathtt{n}\sigma/\sigma\}
\end{array}
$$

Conclusions

Fresh Logic could be a valid logic programming environment. The translation of FOLN suggests a relationship between HOAS and FM techniques, as well as (automatically) giving one semantics to the former. How about a HOAS-type logic with ∇ and $\#$ used intead of ∇ and local contexts.

Appendix: more on the translation

```
(* pi-calculus a la Miller and Tiu,
   delegating function types directly to FreshOCaml *)

type atom = unit name;;

type proc = Nil
| Snd of atom*atom*proc | Rcv of atom*(atom->proc)
| Par of proc*proc | New of (atom->proc)
| Rep of proc;;

let rec subs a b p = match p with
  Snd(x,y,p)     ->  Snd(subsA a b x,subsA a b y,subs a b p)
| Rcv(x,f)       -> Rcv(subsA a b x,
    function n -> let m=fresh in subs m b ((swap m and a in f) n))
           (* Here we ensure that n is nabla-quantified by
              explicitly renaming a to m to avoid clash *)
| Par(p1,p2)     -> Par(subs a b p1,subs a b p2)
| New(f)         -> New(function n ->
    let m=fresh in subs m b ((swap m and a in f) n))
   (* Here we ensure that n is nabla-quantified by
      explicitly renaming a to m to avoid clash *)
| Rep(p)         -> Rep(subs a b p)
| Nil            -> Nil
and subsA a b x = if x=a then b else x;;
```

Appendix: more on the translation

In symbols,

$$New(f)[a \mapsto b] = New\Big(\lambda n.\,\text{И} m.\ (((m\,a)f)n)[m \mapsto b]\Big).$$