One-and-a-halfth order logic

Murdoch J Gabbay, Heriot Watt University, UK
Joint work with Aad Mathijssen, TU/e, the Netherlands

PPDP, Venice, Italy 11/7/2006
Thanks...

...to the Computer Science department of the University of Venice, for making possible a visit to Venice in early 2006 (photos on my webpage).

...to the Technical University of Eindhoven for supporting this work.
Recall: First-Order Logic with equality (FOL)

Fix countably infinitely many variable symbols $a, b, c, \ldots$. Let terms be:

$$t ::= a$$

(More interesting syntactic universes of terms are possible!)

Formulae or predicates are:

$$\phi ::= \bot \mid \phi \Rightarrow \phi \mid \forall a.\phi \mid t \approx t'.$$

Write $\equiv$ for syntactic identity (identify formulae up to $\alpha$-equivalence).
A context \( \Phi \) and cocontext \( \Psi \) are finite and possibly empty sets of formulae. A judgement is a pair \( \Phi \vdash \Psi \). Valid judgements:

- (Axiom) \( \phi, \Phi \vdash \Psi, \phi \)
- (\( \Rightarrow R \)) \( \phi, \Phi \vdash \Psi, \psi \)
- (\( \Rightarrow L \)) \( \Phi \vdash \Psi, \phi \Rightarrow \psi \)
- (\( \forall R \)) \( \Phi \vdash \Psi, \forall a. \psi \)
- (\( \forall L \)) \( \phi[a \rightarrow t], \Phi \vdash \Psi \)
- (\( \approx L \)) \( \phi[a \rightarrow t], \Phi \vdash \Psi \)
- (\( \approx R \)) \( \Phi \vdash \Psi \)
- \( \perp, \Phi \vdash \Psi \)
What is the status of this definition?

What are $\phi$ and $\psi$?

Meta-variables ranging over formulae.

What are $t$ and $a$?

Meta-variables ranging over terms and variable symbols.

What is $\phi[a\mapsto t]$?

A meta-level operation defined which given a real predicate, variable symbol, and term, gives a predicate.

What is ‘$a$ fresh for $\Phi$ and $\Psi$’?

A meta-level condition only defined when given a real context and cocontext.
What does this definition serve to establish?

An entailment relation

\[ \Phi \vdash \Psi. \]

Thanks to the predicate part of FOL this can be internalised as

\[ '\Phi^\wedge \implies \Psi^\vee' \text{ holds}, \]

where \( \{\phi_1, \ldots, \phi_n\}^\wedge \equiv \phi_1 \wedge \cdots \wedge \phi_n \) and \( \{\}^\wedge = \top \), and

\( \{\phi_1, \ldots, \phi_n\}^\vee \equiv \phi_1 \vee \cdots \vee \phi_n \) and \( \{\}^\vee = \bot \).

So FOL is a syntax, and a set of valid formulae.

We’ll return to this later.
Quite a lot of things happen in the meta-level in FOL. For example

\[ \vdash \forall a. \forall b. \phi \iff \forall b. \forall a. \phi \]

is derivable for every value of the meta-variable \( \phi \):

\[
\begin{align*}
\phi \vdash \phi \\
\hline
\forall b. \phi \vdash \phi \\
\hline
\forall a. \forall b. \phi \vdash \phi \\
\hline
\forall a. \forall b. \phi \vdash \forall a. \phi \\
\hline
\forall a. \forall b. \phi \vdash \forall b. \forall a. \phi
\end{align*}
\]
However, the fact that this happens for all $\phi$ cannot be expressed in FOL.

Some nice example theorems:

- If $t \approx t'$ then $\phi[a \leftarrow t] \iff \phi[a \leftarrow t']$.
- If $a \not\in \text{fv}(\phi)$ then $\vdash (\forall a. \phi) \iff \phi$.
- $\forall a. \forall b. \phi$ if and only if $\forall b. \forall a. \phi$.

Normally you might go to higher-order logic to express universal properties ranging over all predicates. However, unification up to $\beta$-equality is undecidable, the models get more complex, and there are other prices for the convenience.
One-and-a-half proof schema

One-and-a-halfth order logic applies nominal terms to represent the meta-level.

Take (nominal) term-formers \( \approx, \forall, \Rightarrow, \bot, \) and \( \text{sub}. \)

Read these as ‘equals’, ‘forall’, ‘implies’, ‘false’ (or ‘bot’), and ‘substitute’.

Terms are \( t ::= a \) and formulae or predicates are:

\[
\phi ::= P \mid \bot \mid P \Rightarrow P \mid \forall[a]P \mid t \approx t' \mid P[a \mapsto t]
\]

Here we write \( P[a \mapsto t] \) for \( \text{sub}([a]P, t) \).
Sugar

Write

- $\neg \phi$ for $\phi \Rightarrow \bot$,
- $\phi \land \phi'$ for $\neg (\phi \Rightarrow \neg \phi')$,
- $\phi \iff \phi'$ for $(\phi \Rightarrow \phi') \land (\phi' \Rightarrow \phi)$,
- $\phi \lor \phi'$ for $\neg (\phi) \Rightarrow \phi'$,
- $\top$ for $\bot \Rightarrow \bot$.

Write $\Phi$, $\Psi$ for contexts, which are finite sets of formulae.

Let a primitive freshness assertion be $a \# P$, read it as ‘$a$ does not occur in $P$’. Write $\Delta$ for a freshness context, a finite set of primitive freshness assertions.
Sequent derivation rules

\[
\begin{align*}
\phi, \Phi \vdash \Psi, \phi & \quad (Axiom) \\
\Phi \vdash \Psi, \phi & \quad \phi \Rightarrow \psi, \Phi \vdash \Psi & \quad (\Rightarrow L) \\
\phi \Rightarrow \psi, \Phi \vdash \Psi & \quad \phi \Rightarrow \psi, \Phi \vdash \Psi & \quad (\Rightarrow R) \\
\phi', \Phi \vdash \Psi & \quad \Delta \vdash_{\text{SUB}} \phi' = \phi[a \mapsto t] & \quad (\forall L) \\
\forall[a] \phi, \Phi \vdash \Psi & \quad \phi \Rightarrow \psi, \Phi \vdash \Psi & \quad (\forall R) \\
\Phi \vdash \Psi, \psi & \quad \Delta \vdash a \# \Phi, \Psi & \\
\Phi \vdash \Psi, \forall[a] \psi & 
\end{align*}
\]
Em... just a few more sequent derivation rules

\[
\frac{\Phi \vdash \Psi, t \approx t}{(\approx R)}
\]

\[
\frac{\phi', \Phi \vdash \Psi \quad \Delta \models_{\text{SUB}} \phi' = \phi''[a \mapsto t'] \quad \Delta \models_{\text{SUB}} \phi = \phi''[a \mapsto t]}{t' \approx t, \phi, \Phi \vdash_{\Delta} \Psi}{(\approx L)}
\]

\[
\frac{\phi', \Phi \vdash_{\Delta} \Psi \quad \Delta \models_{\text{SUB}} \phi' = \phi}{\phi, \Phi \vdash_{\Delta} \Psi}{(StructL)}
\]

\[
\frac{\Phi \vdash_{\Delta} \Psi, \psi' \quad \Delta \models_{\text{SUB}} \psi' = \psi}{\Phi \vdash_{\Delta} \Psi, \psi}{(StructR)}
\]
Example derivations

\[ \forall [a] \forall [b] X \vdash X \quad a \# \forall [a] \forall [b] X \]
\[ \forall [a] \forall [b] X \vdash \forall [a] X \quad b \# \forall [a] \forall [b] X \)
\[ \forall [a] \forall [b] X \vdash \forall [b] \forall [a] X \]

\[ (Axiom) \]
\[ X \vdash X \quad \frac{X \vdash X}{\forall [b] X \vdash X} \quad \text{(\(\forall L\))} \]
\[ \vdash \forall [b] X = X[\text{b} \mapsto \text{b}] \quad \frac{\vdash \forall [b] X = (\forall [b] X)[\text{a} \mapsto \text{a}]}{\forall [a] \forall [b] X \vdash X} \quad \text{(\(\forall L\))} \]

Semantics in FOL: “For all \( \phi \), \( \forall a. \forall b. \phi \vdash \forall b. \forall a. \phi \).”
Freshness part of the derivation

\[
\begin{align*}
& b\#[b]X \quad (#\lfloor a \rfloor) \\
\frac{b\#\forall[b]X \quad (#f)}{b\#\forall[a]\forall[b]X \quad (#\lfloor a \rfloor)} \\
\frac{b\#\forall[a]\forall[b]X \quad (#f)}{b\#\forall[a]\forall[b]X}
\end{align*}
\]
Another example derivation

\[ X[a\mapsto T'] \vdash X[a\mapsto T'] \]
\[ (Axiom) \]
\[ \vdash_{\text{SUB}} X[a\mapsto a][a\mapsto T'] = X[a\mapsto T'], \]
\[ \vdash_{\text{SUB}} X[a\mapsto a][a\mapsto T] = X[a\mapsto T] \]  
\[ (\approx L) \]
\[ T' \approx T, \ X[a\mapsto T] \vdash X[a\mapsto T'] \]

Semantics in FOL:

“For all \( t \) and \( t' \) and \( \phi \), \( t' \approx t, \phi[a\mapsto t] \vdash \phi[a\mapsto t'] \).”
One more example derivation

\[
\begin{array}{c}
X \vdash_{a \# x} X \quad (Axiom) \\
\hline
a \# X \vdash a \# X \\
\hline
X \vdash_{a \# x} \forall[a] X \\
\end{array}
\]

(\forall R)

Semantics in FOL:

“For all \( \phi \) and \( a \), if \( a \not\in f\nu(\phi) \) then \( \phi \vdash \forall a.\phi \).”
A nice theorem:

\[
\begin{array}{c}
\Phi \vdash_{\Delta} \Psi, \phi \quad \phi, \Phi \vdash_{\Delta} \Psi \\
\hline
\Phi \vdash_{\Delta} \Psi
\end{array}
\]

(Cut)

**Theorem (cut-elimination):** Cut is eliminable.

The cut-elimination procedure is almost standard — but this is cut-elimination in the presence of unknown formulae.

Since the cut-elimination procedure is normally written parametrically over those formulae, this is no surprise really. However, the meta-level reasoning about substitution and \(\alpha\)-equivalence is now all completely explicit on the nominal terms.
Another nice theorem:

Say a nominal term is **closed** when it mentions no unknowns. So $a$ is closed but $X$ is not.

**Theorem:** First-order logic (and its derivations) correspond to sequents of closed terms (and their derivations); term-for-term up to $\vdash_{\text{SUB}}$, and proof-rule by proof-rule (up to $(\text{Struct})$).
Recall that **FOL** is just valid formulae

\[
\begin{align*}
    P \Rightarrow Q \Rightarrow P & \quad (P \Rightarrow Q) \Rightarrow (Q \Rightarrow R) \Rightarrow (P \Rightarrow R) \\
    \neg \neg P \Rightarrow P & \quad \forall[a](P \land Q) \iff \forall[a]P \land \forall[a]Q \\
    \bot \Rightarrow P & \quad a \# P \vdash \forall[a](P \Rightarrow Q) \iff P \Rightarrow \forall[a]Q \\
    T \approx T & \quad \forall[a]P \Rightarrow P[a \mapsto T] \\
    U \approx T \land P[a \mapsto T] \Rightarrow P[a \mapsto U]
\end{align*}
\]

This plus modus ponens gives the same valid formulae as the sequent system (but no proof-theory!).
In Higher-Order Logic (HOL), propositions have a type $o$ and $\forall_\sigma$ is a constant with type $(\sigma \rightarrow o) \rightarrow o$, write just $\forall$ or $\forall : (\sigma \rightarrow o) \rightarrow o$.

Then the two judgements express the same idea:

\[
\forall \lambda f. (\forall \lambda a. \forall \lambda b. f a b \leftrightarrow \forall \lambda b. \forall \lambda a. f a b) \text{ is valid'}
\]

\[
\forall[a] \forall[b] P \leftrightarrow \forall[b] \forall[a] P \text{ is valid'}. 
\]

$f$ has function type. If $a : \sigma$ and $b : \tau$ then $f : \sigma \rightarrow \tau \rightarrow o$ and ‘$f a b$ is $P$’.
Second/Higher-order logic

Similarly:

- ‘If $t \approx t'$ then $\phi[a \mapsto t] \iff \phi[a \mapsto t']$’ becomes

  $$t \approx t' \vdash \forall \lambda f. (ft \iff ft').$$

  in HOL.

  Note the types: $f$ has function type and if $t : \sigma$ then $f : \sigma \to o$ and $\forall : ((\sigma \to o) \to o) \to o$.

- ‘If $a \notin fv(\phi)$ then $\vdash \forall a. \phi \iff \phi$’ is not expressible in HOL.
Relation to HOL

One-and-a-halfth-order logic is not fully higher-order. We can write

\[ X \vdash Y \]

meaning in FOL “For all formulae \( \phi \) and \( \psi \), \( \phi \vdash \psi \).”

In HOL we can write this as

\[ \vdash \forall \phi. \forall \psi. (\phi \Rightarrow \psi). \]

However we can also write

\[ \vdash \forall \psi. ((\forall \phi. \phi) \Rightarrow \psi). \]

This is not possible in one-and-a-halfth-order logic: \( (\forall [X]X) \vdash Y \) is not syntax.
Relation to HOL

Not direct since we can express $a \not\equiv t$ and HOL cannot, but HOL can quantify over predicates to the left of an implication.

Also, suppose $X : o$ and $t : T$.

$X[a \mapsto t]$ corresponds to $ft$ and so $X$ corresponds to $f$ where $f : T \to o$.

But $X[a \mapsto t][a' \mapsto t']$ corresponds to $f'tt'$ and $X$ corresponds to $f' : T \to T \to o$.

But $X[a' \mapsto t'][a \mapsto t]$ corresponds to $f't't$ and $X$ corresponds to $f' : T \to T \to o$.

Similarly $X[a \mapsto t][a' \mapsto t'][a'' \mapsto t'']\ldots$
Relation to HOL

This is type raising.

In one-and-a-halfth-order logic, $X$ remains at $o$ and the universal quantification implicit in the use of $X$ allows this one symbol to represent a function of arbitrary arity — just like the meta-variable $\phi$, which we can write under substitutions $[a \mapsto t]$, $[a \mapsto t][a' \mapsto t]$, and so on, as we please.
Conclusions

Nominal Terms have ‘weak’ object-level variable symbols (atoms) with primitive facilities for abstraction and $\alpha$-renaming and ‘strong’ meta-level variable symbols (variables or unknowns).

We can use this to axiomatise/build sequent systems for logic-with-binding, like first-order logic.
Conclusions

Sequent and axiomatic presentations systems are possible:

‘$\Phi \vdash_\Delta \Psi$ is derivable’

translating to ‘$\Delta \vdash (\Phi^\wedge \Rightarrow \Psi^\vee)$ is valid’.

($^\wedge$ means ‘put $\wedge$ between the elements of $\Phi$', similarly for $^\vee$).
Conclusions

We get extra. E.g. one-and-a-halfth order logic has predicate unknowns; thus enabling us to reason universally on predicates in a new way.

This really new, because $a \not\# X$ is not expressible using other techniques (to our knowledge); not in full generality for a completely unknown $X$. 
Conclusions

For some further work, how about…

- Two-and-a-halfth-order logic, where you can abstract $P$, and a predicate can assert freshness properties $a \# P$ of its own unknowns?

- Implementation and automation?

- Semantics (aside from in FOL)?

Note that this work is based on nominal algebra, a theory of algebraic equality of nominal terms. Watch this space.

The end
Axioms of substitution $\vdash^\text{SUB}$

Write $t \vdash^\text{SUB}_\Delta u$ when $t = u$ is derivable from assumptions $\Delta$ using the following axioms:

\begin{align*}
(f \mapsto) & \quad f(u_1, \ldots, u_n)[a \mapsto t] = f(u_1[a \mapsto t], \ldots, u_n[a \mapsto t]) \\
([b] \mapsto) & \quad b \# t \Rightarrow ([b]u)[a \mapsto t] = [b](u[a \mapsto t]) \\
(var \mapsto) & \quad a[a \mapsto t] = t \\
(u \mapsto) & \quad a \# u \Rightarrow u[a \mapsto t] = u \\
(ren \mapsto) & \quad b \# u \Rightarrow u[a \mapsto b] = (b \ a) \cdot u \\
(perm) & \quad a, b \# t \Rightarrow (a \ b) \cdot t = t
\end{align*}
Equality is decidable in the theory of substitution (philosophically interesting fact, that!).

I do not know if unification is decidable.

The axioms above are sound and complete for a Herbrand-style model.

The category of all models of substitution is cartesian-closed. Very interesting programming and logic principles; one-and-a-halfth-order logic is one creature inhabiting this new and wonderful universe.

There is much more out there.
Permutation action

\[ \pi \cdot a \equiv \pi(a) \quad \pi \cdot (\pi' X) \equiv (\pi \circ \pi')X \]

\[ \pi \cdot [a]t \equiv [\pi(a)](\pi t) \]

\[ \pi \cdot f(t_1, \ldots, t_n) \equiv f(\pi t_1, \ldots, \pi t_n) \]
Nominal Terms

Nominal terms are a syntax inductively generated by

\[ t ::= a \mid \pi X \mid [a]t \mid f(t, \ldots, t). \]

Here:

- We fix \( a, b, c, \ldots \in \mathbb{A} \) a countably infinite set of atoms.
- We fix \( X, Y, Z, \ldots \in \mathbb{V} \) a countably infinite set of unknowns (disjoint from the atoms; everything’s disjoint).
- We fix \( f, g, \ldots \) some term-formers.
- Call \([a]t\) an abstraction.
Nominal Terms

\[ t ::= a \mid \pi X \mid [a]t \mid f(t, \ldots, t). \]

\(\pi\) is a permutation. A permutation is a finitely supported bijection on \(A\). Finitely supported means:

\[ \pi(a) = a \text{ for all } a \in A \text{ except for a finite set of atoms.} \]
Nominal Terms

For example permutations are:

\[(a \ b \ c) \quad \text{and} \quad \text{Id} \]

(a to b to c to a, and the identity function). Permutations are not:

\[(a_1 \ a_2)(a_3 \ a_4) \ldots \]

for \(A = \{a_1, a_2, \ldots\}\).
Freshness assertions \( a\#t \)

Read \( a\#X \) as ‘\( a \) does not occur in \( X \)’, or ‘\( a \) is fresh for \( X \)’.

Then we can characterise \( \alpha \)-equivalence as:

\[
b\#X \Rightarrow [b](b \ a)X = [a]X.
\]

For the moment I’m just telling you that this is the case.

Call a pair \( a\#t \) a freshness assertion. If \( t \equiv X \) call it primitive.
Freshness derivation rules (formally)

\[
\begin{align*}
\frac{a \# b}{a \# \Box} \quad &\quad & \frac{a \# t_1 \cdots a \# t_n}{a \# f(t_1, \ldots, t_n)} \quad &\quad & \frac{a \# \pi X}{a \# X} \\
\frac{a \# \Box t}{a \# a \Box t} \quad &\quad & \frac{a \# \Box b}{a \# b \Box t} \quad &\quad & \frac{\pi^{-1}(a) \# X}{a \# \pi X}
\end{align*}
\]
Core equality derivation rules (formally)

\[ t = t \] (refl)

\[ t = u \quad \text{(symm)} \]

\[ u = t \]

\[ t = u \quad u = v \] (tran)

\[ t = v \]

\[ t = u \]

\[ C[t] = C[u] \] (cong)

\[ a\#t \quad b\#t \] (perm)

\[ (a \ b) \cdot t = t \]
For example

\[
\begin{align*}
\frac{\#ab}{a\#b} \quad \frac{\#ab}{a\#b} \quad (\text{perm}) \\
\frac{[a]a = [b]b}{(b \ a) \cdot [b]b \equiv [a]a}
\end{align*}
\]

\[
\begin{align*}
\frac{\#X}{b\#[a]X} \quad \frac{\#X}{a\#[a]X} \quad (\text{perm}) \\
\frac{[b](b \ a)X = [a]X}{(b \ a) \cdot [a]X \equiv [b](b \ a)X}
\end{align*}
\]

Here \( \equiv \) is syntactic identity.
A freshness context $\Delta$ is a finite set of primitive freshness assertions.

An axiom $\Delta \vdash t = u$ is a pair of a freshness context and an equality assertion. If $\Delta$ is empty write it just $t = u$.

We can use axioms to enrich provable equality, which currently stands at some generalisation of $\alpha$-equivalence.
(Assume suitable term-formers $\lambda$, app and sugar.)

As an axiom, we instantiate $Y$ and $X$ to ‘any term’ when we enrich equality, generating a family of equalities for each instantiation (and each context). Thus, $Y$ and $X$ do represent ‘any term’, with universal quantification at top level. Instantiation is direct replacement of an unknown by a term (no capture avoidance).
Theory of first-order logic FOL

\[ P \Rightarrow Q \Rightarrow P = \top \]
\[ \neg \neg P \Rightarrow P = \top \]
\[ \bot \Rightarrow P = \top \]
\[ T \approx T = \top \]
\[ (P \Rightarrow Q) \Rightarrow (Q \Rightarrow R) \Rightarrow (P \Rightarrow R) = \top \]
\[ \forall[a](P \land Q) \iff \forall[a]P \land \forall[a]Q = \top \]
\[ a \# P \vdash \forall[a](P \Rightarrow Q) \iff (P \Rightarrow \forall[a]Q) = \top \]

\[ \forall[a]P \Rightarrow P[a \mapsto T] = \top \]
\[ U \approx T \land P[a \mapsto T] \Rightarrow P[a \mapsto U] = \top \]

(Assume suitable term-formers \( \approx, \forall, \Rightarrow, \bot \) and sugar.)

The ‘\( \models \)’ bit just converts a predicate into a nominal algebra judgement.
Theory of substitution SUB

\[ f(X_1, \ldots, X_n)[a \mapsto T] = f(X_1[a \mapsto T], \ldots, X_n[a \mapsto T]) \]

\[ b \# T \vdash ([b]X)[a \mapsto T] = [b](X[a \mapsto T]) \]

\[ a[a \mapsto T] = T \]

\[ a \# X \vdash X[a \mapsto T] = X \]

\[ b \# X \vdash X[a \mapsto b] = (b \ a)X \]
### Picture of what we have done

- **$\equiv$** is syntactic identity
  \[
  [a]a \neq [b]b
  \]
- **$\cong$** (no axioms) is $\alpha$-equivalence
  \[
  b \# X \vdash [b](b a) X = [a]x
  \]
- **$\cong_{\text{SUB}}$** is substitution
  \[
  b \# Y \vdash Y[b \mapsto X] = Y
  \]
- **$\cong_{\text{LAM}}$** is $\alpha\beta$-equivalence
  \[
  (\lambda[a]a)b = b
  \]
- **$\cong_{\text{FOL}}$** is logical equivalence
  \[
  (\forall[a](a \approx a)) = \top
  \]

All these theories are really very interesting beasts.