One-and-a-halfth-order Logic

Aad Mathijssen        Murdoch J. Gabbay

24th May 2006
Motivation

Consider the following valid assertions in first-order logic:

- $\phi \supset \psi \supset \phi$
- if $a \notin fn(\phi)$ then $\phi \supset \forall a.\phi$
- if $a \notin fn(\phi)$ then $\phi \supset \phi[a \mapsto t]$
- if $b \notin fn(\phi)$ then $\forall a.\phi \supset \forall b.\phi[a \mapsto b]$
Motivation

Consider the following valid assertions in first-order logic:

- $\phi \supset \psi \supset \phi$
- if $a \notin fn(\phi)$ then $\phi \supset \forall a.\phi$
- if $a \notin fn(\phi)$ then $\phi \supset \phi[a \mapsto t]$
- if $b \notin fn(\phi)$ then $\forall a.\phi \supset \forall b.\phi[a \mapsto b]$

These are not valid syntax in first-order logic, because of meta-level concepts:

- meta-variables varying over syntax: $\phi, \psi, a, b, t$
- properties of syntax: $a \notin fn(\phi), \phi[a \mapsto t], \alpha$-equivalence
Motivation

Consider the following valid assertions in first-order logic:

- $\phi \supset \psi \supset \phi$
- if $a \notin fn(\phi)$ then $\phi \supset \forall a.\phi$
- if $a \notin fn(\phi)$ then $\phi \supset \phi[a \mapsto t]$
- if $b \notin fn(\phi)$ then $\forall a.\phi \supset \forall b.\phi[a \mapsto b]$

These are not valid syntax in first-order logic, because of meta-level concepts:

- meta-variables varying over syntax: $\phi, \psi, a, b, t$
- properties of syntax: $a \notin fn(\phi), \phi[a \mapsto t], \alpha$-equivalence

Is there a logic in which the above assertions can be expressed directly in the syntax?
Motivation (2)

Consider the following derivations in Gentzen’s sequent calculus:

\[
\frac{\psi, \phi \vdash \phi}{\phi \vdash \psi \supset \phi} \quad (\text{Ax}) \\
\frac{\phi \vdash \psi \supset \phi}{\vdash \phi \supset \psi \supset \phi} \quad (\supset \text{R})
\]

\[
\frac{p(d), p(c) \vdash p(c)}{p(c) \vdash p(d) \supset p(c)} \quad (\text{Ax}) \\
\frac{p(c) \vdash p(d) \supset p(c)}{\vdash p(c) \supset p(d) \supset p(c)} \quad (\supset \text{R})
\]

And for \( b \notin fn(\phi) \):

\[
\frac{\forall a. \phi \vdash \forall b. \phi[a \mapsto b]}{\forall a. \phi \vdash \forall b. \phi[a \mapsto b]} \quad (\text{Ax}) \\
\frac{\vdash \forall a. \phi \supset \forall b. \phi[a \mapsto b]}{(\supset \text{R})}
\]

\[
\frac{\forall c. p(c) \vdash \forall d. p(d)}{\forall c. p(c) \vdash \forall d. p(d)} \quad (\text{Ax}) \\
\frac{\vdash \forall c. p(c) \supset \forall d. p(d)}{(\supset \text{R})}
\]
Motivation (2)

Consider the following derivations in Gentzen’s sequent calculus:

\[
\frac{\psi, \phi \vdash \phi}{\phi \vdash \psi \supset \phi} \quad (\text{Ax})
\]

\[
\frac{\phi \vdash \psi \supset \phi}{\neg \phi \supset \psi \supset \phi} \quad (\supset \text{R})
\]

\[
\frac{p(d), p(c) \vdash p(c)}{p(c) \vdash p(d) \supset p(c)} \quad (\text{Ax})
\]

\[
\frac{p(c) \vdash p(d) \supset p(c)}{p(c) \supset p(d) \supset p(c)} \quad (\supset \text{R})
\]

And for \( b \notin fn(\phi) \):

\[
\frac{\forall a. \phi \vdash \forall b. \phi[a \mapsto b]}{\neg \forall a. \phi \supset \forall b. \phi[a \mapsto b]} \quad (\text{Ax})
\]

\[
\frac{\forall c. p(c) \vdash \forall d. p(d)}{\forall c. p(c) \supset \forall d. p(d)} \quad (\supset \text{R})
\]

The left ones are not derivations, they are schemas of derivations. When \( p \) is a specific atomic predicate and \( c \) and \( d \) are specific variables, the right ones are derivations; they are instances of the schemas on the left.
Motivation (2)

Consider the following derivations in Gentzen’s sequent calculus:

\[
\frac{\psi, \phi \vdash \phi}{\phi \vdash \psi \supset \phi} \quad \text{(Ax)} \\
\frac{\phi \vdash \psi \supset \phi}{\vdash \phi \supset \psi \supset \phi} \quad \text{(\supset R)}
\]

\[
\frac{p(d), p(c) \vdash p(c)}{p(c) \vdash p(d) \supset p(c)} \quad \text{(Ax)} \\
\frac{p(c) \vdash p(d) \supset p(c)}{\vdash p(c) \supset p(d) \supset p(c)} \quad \text{\supset R)}
\]

And for \( b \notin fn(\phi) \):

\[
\frac{\forall a. \phi \vdash \forall b. \phi[a \mapsto b]}{\vdash \forall a. \phi \supset \forall b. \phi[a \mapsto b]} \quad \text{(Ax)} \\
\frac{\forall c. p(c) \vdash \forall d. p(d)}{\vdash \forall c. p(c) \supset \forall d. p(d)} \quad \text{(Ax)}
\]

The left ones are not derivations, they are schemas of derivations. When \( p \) is a specific atomic predicate and \( c \) and \( d \) are specific variables, the right ones are derivations; they are instances of the schemas on the left.

Is there a logic in which the derivation on the left is a derivation too?
Motivation (3)

First-order logic and its sequent calculus formalises reasoning.

But also a lot of reasoning is about first-order logic.

So why shouldn’t that be formalised?
Motivation (3)

First-order logic and its sequent calculus formalises reasoning.

But also a lot of reasoning is about first-order logic.

So why shouldn’t that be formalised?

One-and-a-halfth-order logic does this by means of:

- formalising meta-variables;
- making properties of syntax explicit.
Overview

- Introduction to one-and-a-halfth-order logic
- Syntax of one-and-a-halfth-order logic
- Sequent calculus for one-and-a-halfth-order logic
- Relation to first-order logic
- Axiomatisation of one-and-a-halfth-order logic
- Conclusions, related and future work
Introduction

In the syntax of one-and-a-halfth-order logic:

- **Unknowns** $P$, $Q$, and $T$ represent meta-level variables $\phi$, $\psi$, and $t$.
- **Atoms** $a$ and $b$ represent meta-level variables $a$ and $b$.
- **Freshness** $a \# P$ represents $a \notin fn(\phi)$.
- **Explicit substitution** $P[a \mapsto T]$ represents $\phi[a \mapsto t]$. 
Introduction (2)

The meta-level assertions in first-order logic

- $\phi \supset \psi \supset \phi$
- if $a \not\in fn(\phi)$ then $\phi \supset \forall a.\phi$
- if $a \not\in fn(\phi)$ then $\phi \supset \phi[a \mapsto t]$
- if $b \not\in fn(\phi)$ then $\forall a.\phi \supset \forall b.\phi[a \mapsto b]$

correspond to valid assertions in the syntax of one-and-a-halfth-order logic:

- $P \supset Q \supset P$
- $a\#P \rightarrow P \supset \forall[a]P$
- $a\#P \rightarrow P \supset P[a \mapsto T]$
- $b\#P \rightarrow \forall[a]P \supset \forall[b]P[a \mapsto b]$
Introduction (3)

In derivations of one-and-a-halfth-order logic:

- *Contexts of freshnesses* are added to the sequents.
- *Derivability of freshnesses* are added as side-conditions.
- *Substitutional equivalence on terms* is added as two derivation rules, taking care of $\alpha$-equivalence and substitution.
Introduction (4)

The (schematic) derivations in first-order logic

\[
\frac{\psi, \phi \vdash \phi}{\phi \vdash \psi \supset \phi} \quad \text{(Ax) } (\supset \text{R})
\]

\[
\frac{p(d), p(c) \vdash p(c)}{p(c) \vdash p(d) \supset p(c)} \quad \text{(Ax) } (\supset \text{R})
\]

correspond to valid derivations in one-and-a-halfth-order logic:

\[
\frac{Q, P \vdash P}{P \vdash Q \supset P} \quad \text{(Ax) } (\supset \text{R})
\]

\[
\frac{p(d), p(c) \vdash p(c)}{p(c) \vdash p(d) \supset p(c)} \quad \text{(Ax) } (\supset \text{R})
\]
Introduction (5)

The (schematic) derivations in first-order logic, where $b \not \in fn(\phi)$,

\[
\begin{align*}
\forall a. \phi & \vdash \forall b. \phi[a \mapsto b] \quad \text{(Ax)} \\
\vdash \forall a. \phi & \supset \forall b. \phi[a \mapsto b] \quad \text{(\supset R)} \\
\forall c. p(c) & \vdash \forall d. p(d) \quad \text{(Ax)} \\
\vdash \forall c. p(c) & \supset \forall d. p(d) \quad \text{(\supset R)}
\end{align*}
\]

correspond to valid derivations in one-and-a-halfth-order logic:

\[
\begin{align*}
\forall [a] P & \vdash_{\mathrm{b\#P}} \forall [a] P \quad \text{(Ax)} \\
\forall [a] P & \vdash_{\mathrm{b\#P}} \forall [b] P[a \mapsto b] \quad \text{(\supset R)} \\
\forall [c] p(c) & \vdash_{\emptyset} \forall [c] p(c) \quad \text{(Ax)} \\
\forall [c] p(c) & \vdash_{\emptyset} \forall [d] p(d) \quad \text{(\supset R)} \\
\vdash_{\emptyset} \forall [c] p(c) & \supset \forall [d] p(d) \quad \text{(\supset R)}
\end{align*}
\]

(b\#P \vdash_{\mathrm{SUB}} \forall [a] P = \forall [b] P[a \mapsto b])

(\emptyset \vdash_{\mathrm{SUB}} \forall [c] p(c) = \forall [d] p(d))
Syntax of one-and-a-halfth-order logic

We use Nominal Terms to specify the syntax.

Nominal terms have built-in support for:

- meta-variables
- freshness
- binding

Nominal terms allow for a direct and natural representation of systems with binding.

Nominal terms are first-order, not higher-order.
Sorts

**Base sorts** $\mathbb{P}$ for ‘predicates’ and $\mathbb{T}$ for ‘terms’.

**Atomic sort** $\mathbb{A}$ for the object-level variables.

**Sorts** $\tau$:

$$\tau ::= \mathbb{P} \mid \mathbb{T} \mid \mathbb{A} \mid [\mathbb{A}] \tau$$
Terms

Atoms $a, b, c, \ldots$ have sort $\mathbb{A}$; they represent object-level variable symbols.

Unknowns $X, Y, Z, \ldots$ have sort $\tau$; they represent meta-level variable symbols. Let $P, Q, R$ be unknowns of sort $\mathbb{P}$, and $T, U$ of sort $\mathbb{T}$.

We call $\pi \cdot X$ a **moderated unknown**.
This represents the permutation of atoms $\pi$ acting on an unknown term.

Term-formers $f_\rho$ have an associated arity $\rho = (\tau_1, \ldots, \tau_n) \tau$.
$f : \rho$ means ‘$f$ with arity $\rho$’.

Terms $t$, subscripts indicate sorting rules:

$$t ::= a_\mathbb{A} \mid (\pi \cdot X_\tau)_\tau \mid ([a_\mathbb{A}]t_\tau)_\mathbb{A}_\tau \mid (f_{(\tau_1, \ldots, \tau_n)}(t^1_{\tau_1}, \ldots, t^n_{\tau_n}))_\tau$$

Write $f$ for $f()$ if $n = 0$. 
Terms (2)

Term-formers for one-and-a-halfth-order logic:

- \( \bot : (\text{P})\text{P} \) represents \textit{falsity};
- \( \supset : (\text{P}, \text{P})\text{P} \) represents \textit{implication}, write \( \phi \supset \psi \) for \( \supset(\phi, \psi) \);
- \( \forall : ([\text{A}]\text{P})\text{P} \) represents \textit{universal quantification}, write \( \forall[a]\phi \) for \( \forall([a]\phi) \);
- \( \approx : (\text{T}, \text{T})\text{P} \) represents \textit{object-level equality}, write \( t \approx u \) for \( \approx(t, u) \);
- \( \text{var} : ([\text{A}]\text{P})\text{P} \) is \textit{variable casting}, forced upon us by the sort system, write \( a \) for \( \text{var}(a) \);
- \( \text{sub} : ([\text{A}]\tau, \text{T})\tau \), where \( \tau \in \{\text{T}, [\text{A}]\text{T}, \text{P}, [\text{A}]\text{P}\} \), is \textit{explicit substitution}, write \( v[a \mapsto t] \) for \( \text{sub}([a]v, t) \);
- \( p_1, \ldots, p_n : (\text{T}, \ldots, \text{T})\text{P} \) are \textit{object-level predicate term-formers};
- \( f_1, \ldots, f_m : (\text{T}, \ldots, \text{T})\text{T} \) are \textit{object-level term-formers}. 
Terms (3)

Sugar:

\[
\begin{align*}
\top & \text{ is } \perp \sqsubseteq \perp \\
\neg \phi & \text{ is } \phi \sqsubseteq \perp \\
\phi \land \psi & \text{ is } \neg (\phi \sqsubseteq \neg \psi) \\
\phi \lor \psi & \text{ is } \neg \phi \sqsupset \psi \\
\phi \iff \psi & \text{ is } (\phi \sqsubseteq \psi) \land (\psi \sqsubseteq \phi) \\
\exists[a] \phi & \text{ is } \neg \forall[a] \phi
\end{align*}
\]

Descending order of operator precedence:

\[
[a], \ lbracket \_ \ mapsto \_ rbracket, \ \cong, \ \{\\neg, \forall, \exists\}, \ \{\land, \lor\}, \ \sqsubset, \ \iff
\]

\land, \ \lor, \ \sqsubset \text{ and } \iff \text{ associate to the right.}
Terms (3)

Sugar:

\[ \top \text{ is } \bot \sqcup \bot, \quad \neg \phi \text{ is } \phi \sqcup \bot, \quad \phi \land \psi \text{ is } \neg (\phi \sqcup \neg \psi) \]
\[ \phi \lor \psi \text{ is } \neg \phi \sqcup \psi, \quad \phi \iff \psi \text{ is } (\phi \sqcup \psi) \land (\psi \sqcup \phi) \quad \exists[a] \phi \text{ is } \neg \forall[a] \phi \]

Descending order of operator precedence:

\[ [a], \quad \_ [\_ \mapsto \_], \quad \approx, \quad \{ \neg, \forall, \exists \}, \quad \{ \land, \lor \}, \quad \sqsubset, \quad \iff \]

\( \land, \lor, \sqsubset \) and \( \iff \) associate to the right.

Example terms of sort \( \mathbb{P} \):

\[ P \sqsubset Q \sqsubset P \quad P \sqsubset \forall[a]P \quad P \sqsubset P[a \mapsto T] \quad \forall[a]P \sqsubset \forall[b]P[a \mapsto b] \]
Freshness

**Freshness (assertions)** $a \# t$, which means ‘$a$ is fresh for $t$. If $t$ is an unknown $X$, the freshness is called **primitive**.

Write $\Delta$ for a set of **primitive** freshesses and call it a **freshness context**. We may leave out set brackets, writing $a \# X, b \# Y$ instead of $\{a \# X, b \# Y\}$. We may also write $a \# X, Y$ for $a \# X, a \# Y$.

We call $\Delta \rightarrow t$ a **term-in-context**. We may write $t$ if $\Delta = \emptyset$. 
Freshness

**Freshness (assertions)** $a \# t$, which means ‘$a$ is fresh for $t$.

If $t$ is an unknown $X$, the freshness is called **primitive**.

Write $\Delta$ for a set of *primitive* freshesses and call it a **freshness context**.

We may leave out set brackets, writing $a \# X, b \# Y$ instead of $\{a \# X, b \# Y\}$.

We may also write $a \# X, Y$ for $a \# X, a \# Y$.

We call $\Delta \rightarrow t$ a **term-in-context**.

We may write $t$ if $\Delta = \emptyset$.

Example terms-in-context of sort $\mathbb{P}$:

\[
\begin{align*}
P \supset Q \supset P & \quad a \# P \rightarrow P \supset \forall[a] P \\
a \# P \rightarrow P \supset P[a \mapsto T] & \quad b \# P \rightarrow \forall[a] P \supset \forall[b] P[a \mapsto b]
\end{align*}
\]
Derivability of freshness

\[
\begin{align*}
\frac{a \# b}{\# \text{ab}} & \quad \frac{\pi^{-1}(a) \# X}{a \# \pi \cdot X} (\# X) \\
\frac{a \# [a]t}{\# \text{a}} & \quad \frac{a \# t}{\# \text{b}} t (\# \text{b}) \quad \frac{a \# t_1 \cdots a \# t_n}{\# f(t_1, \ldots, t_n)} (\# f)
\end{align*}
\]

\(a\) and \(b\) range over distinct atoms.

Write \(\Delta \vdash a \# t\) when there exists a derivation of \(a \# t\) using the elements of \(\Delta\) as assumptions. Say that \(a \# t\) is derivable from \(\Delta\).
Derivability of freshness

\[
\frac{a \# b}{\text{a} \# \text{b}} \quad \frac{\pi^{-1}(a) \# X}{\text{a} \# \pi \cdot X} \quad (\#X)
\]

\[
\frac{a \# [a]t}{\text{a} \# \text{[a]}t} \quad \frac{a \# [b]t}{\text{a} \# \text{[b]}t} \quad \frac{a \# \pi \cdot X}{\text{a} \# \pi \cdot X} \quad \frac{a \# t}{\text{a} \# \text{t}} \quad \frac{a \# t_1 \cdots a \# t_n}{\text{a} \# \text{f}(t_1, \ldots, t_n)} \quad (\#f)
\]

\(a \) and \(b\) range over distinct atoms.

Write \( \Delta \vdash a \# t \) when there exists a derivation of \( a \# t \) using the elements of \( \Delta \) as assumptions. Say that \( a \# t \) is derivable from \( \Delta \).

Examples:

\[ \vdash a \# \forall[a]P \quad a \# P \vdash a \# \forall[b]P \quad a \# T, U \vdash a \# T \approx U \]
Derivability of equality

Equality (assertions) $t = u$, where $t$ and $u$ are of the same sort.

Derivability:

\[
\frac{t = t}{t = u} \text{ (refl)} \quad \frac{t = u}{u = t} \text{ (symm)} \quad \frac{t = u}{t = v} \text{ (tran)}
\]

\[
\frac{t = u}{C[t] = C[u]} \text{ (cong)} \quad \frac{a \# t \quad b \# t}{(a \ b) \cdot t = t} \text{ (perm)}
\]

\[
\frac{\Delta \pi \sigma}{t^\pi \sigma = u^\pi \sigma} \text{ (ax} \ A) \text{ A is } \Delta \rightarrow t = u
\]

\[
[a \# X_1, \ldots, a \# X_n] \quad \Delta
\]

\[
\vdots
\]

\[
\frac{t = u}{t = u} \text{ (fr)} \quad (a \notin t, u, \Delta)
\]

Write $\Delta \vdash^\tau t = u$ when $t = u$ is derivable from $\Delta$ using axioms $A$ from $\mathsf{T}$ only.
Derivability of equality (2)

Nominal Algebra is the logic of equality between nominal terms.

Nominal algebraic theory SUB of explicit substitution:

- **(var)** \(a \mapsto T\):
  \[a[a \mapsto T] = T\]

- **(#)\:** \(a\#X \mapsto X[a \mapsto T] = X\)

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

- **(abs)** \(b\#T \mapsto ([b]X)[a \mapsto T] = [b](X[a \mapsto T])\)

- **(ren)** \(b\#X \mapsto X[a \mapsto b] = (b\ a) \cdot X\)
Derivability of equality (2)

**Nominal Algebra** is the logic of equality between nominal terms.

Nominal algebraic theory $\text{SUB}$ of explicit substitution:

\[
\begin{align*}
\text{(var $\leftrightarrow$)} & \quad a[a \mapsto T] = T \\
\text{(♯ $\leftrightarrow$)} & \quad a\#X \rightarrow X[a \mapsto T] = X \\
\text{(f $\leftrightarrow$)} & \quad f(X_1, \ldots, X_n)[a \mapsto T] = f(X_1[a \mapsto T], \ldots, X_n[a \mapsto T]) \\
\text{(abs $\leftrightarrow$)} & \quad b\#T \rightarrow ([b]X)[a \mapsto T] = [b](X[a \mapsto T]) \\
\text{(ren $\leftrightarrow$)} & \quad b\#X \rightarrow X[a \mapsto b] = (b \, a) \cdot X
\end{align*}
\]

Examples:

\[
\begin{align*}
b\#P \vdash_{\text{SUB}} \forall[a]P &= \forall[b]P[a \mapsto b] \\
\vdash_{\text{SUB}} X[a \mapsto a] &= X \\
a\#Y \vdash_{\text{SUB}} Z[a \mapsto X][b \mapsto Y] &= Z[b \mapsto Y][a \mapsto X[b \mapsto Y]]
\end{align*}
\]
Sequent calculus for one-and-a-halfth-order logic

We may call terms of sort $\mathbb{P}$ predicates, and denote them by $\phi$ and $\psi$.

Let (predicate) contexts $\Phi, \Psi$ be finite sets of predicates. We may write $\phi$ for $\{\phi\}$, $\phi, \Phi$ for $\{\phi\} \cup \Phi$, and $\Phi, \Phi'$ for $\Phi \cup \Phi'$.

A sequent is a triple $\Phi \vdash_\Delta \Psi$. We may omit empty predicate contexts, e.g. writing $\vdash_\Delta$ for $\emptyset \vdash_\Delta \emptyset$.

Define derivability on sequents...
Sequent calculus (2)

Rules resembling Gentzen’s sequent calculus for first-order logic:

\[ \phi, \Phi \vdash_{\Delta} \Psi, \phi \quad (\text{Ax}) \quad \bot, \Phi \vdash_{\Delta} \Psi \quad (\bot \text{L}) \]

\[ \Phi \vdash_{\Delta} \Psi, \phi, \psi, \Phi \vdash_{\Delta} \Psi \quad (\supset \text{L}) \quad \phi, \Phi \vdash_{\Delta} \Psi, \psi \quad (\supset \text{R}) \]

\[ \phi[a \mapsto t], \Phi \vdash_{\Delta} \Psi \quad (\forall \text{L}) \quad \Phi \vdash_{\Delta} \Psi, \psi \quad (\forall \text{R}) \quad (\Delta \vdash a \neq \Phi, \Psi) \]

\[ \phi[a \mapsto t'], \Phi \vdash_{\Delta} \Psi \quad (\approx \text{L}) \quad \phi[a \mapsto t], \Phi \vdash_{\Delta} \Psi \quad (\approx \text{R}) \]

\[ t' \approx t, \quad \phi[a \mapsto t], \Phi \vdash_{\Delta} \Psi \]

\[ \Phi \vdash_{\Delta} \Psi, t \approx t \]
Sequent calculus (3)

Other rules:

- **StructL**
  \[ \phi', \Phi \vdash \Delta \Psi \quad (\Delta \vdash_{\text{SUB}} \phi' = \phi) \]

- **StructR**
  \[ \Phi \vdash \Delta \Psi, \psi' \quad (\Delta \vdash_{\text{SUB}} \psi' = \psi) \]

- **Fresh**
  \[ \Phi \vdash \Delta \{\forall a \# X_1, \ldots, a \# X_n\} \Psi \quad (\text{Fresh}) \quad (a \notin \Phi, \Psi, \Delta) \]

- **Cut**
  \[ \Phi \vdash \Delta \Psi, \phi', \Phi \vdash \Delta \Psi \quad (\Delta \vdash_{\text{SUB}} \phi = \phi') \]
Example derivations

Derivation of $a \# P \rightarrow P \supset \forall[a]P$:

\[
\begin{align*}
&P \vdash a \# P \quad (Ax) \\
&P \vdash P \quad (Ax) \\
&P \vdash A[a]P \quad (\forall R) \\
&\vdash P \supset A[a]P \quad (\supset R)
\end{align*}
\]

Derivation of $a \# P \rightarrow P \supset P[a \mapsto T]$:

\[
\begin{align*}
&P \vdash a \# P \quad (Ax) \\
&P \vdash P \quad (Ax) \\
&P \vdash P[a \mapsto T] \quad (\text{StructR}) \\
&\vdash a \# P \quad (\supset R)
\end{align*}
\]
Properties of the sequent calculus

We may instantiate unknowns and permute atoms in derivations.

**Theorem 1** If $\Pi$ is a valid derivation of $\Phi \vdash_{\Delta} \Psi$ and $\Delta' \vdash_{\Delta^\pi\sigma} \Psi$, then $\Pi^{\pi}(\sigma, \Delta')$ is a valid derivation of $\Phi^{\pi\sigma} \vdash_{\Delta'} \Psi^{\pi\sigma}$.

$\Pi^{\pi}(\sigma, \Delta')$ is $\Pi$ in which:

- each atom $a$ is replaced by $\pi(a)$;
- each moderated unknown $\pi' \cdot X$ is replaced by $\pi' \cdot \sigma(X)$;
- each freshness context $\Delta$ is replaced by $\Delta'$.
Properties of the sequent calculus (2)

For example, $\Pi$ is the derivation of $a \# P \rightarrow P \supset P[a \mapsto T]$:

\[
\begin{array}{c}
P \vdash a \# P \\
\hline
P \vdash a \# P \\
\hline
\vdash a \# P
\end{array}
\]

\[\text{(Ax)}\]

\[
\begin{array}{c}
P \vdash a \# P \\
\hline
P[\vdash a \mapsto T]
\end{array}
\]

\[\text{(StructR)}\]

\[a \# P \vdash_{\text{sub}} P = P[a \mapsto T]\]

\[\supset R\]

Take $\pi = (a \ b), \sigma = [p(a)/P, a/T]$ and $\Delta' = \emptyset$, then:

- $\Delta' \vdash \Delta^{\pi}\sigma$, i.e. $\emptyset \vdash b \# p(a)$;

- $\Pi^{\pi}(\sigma, \Delta')$ is the following valid derivation of $p(a) \supset p(a)[b \mapsto a]$:

\[
\begin{array}{c}
p(a) \vdash p(a) \\
\hline
\vdash p(a)
\end{array}
\]

\[\text{(Ax)}\]

\[
\begin{array}{c}
p(a) \vdash p(a)[b \mapsto a] \\
\vdash p(a)[b \mapsto a]
\end{array}
\]

\[\text{(StructR)}\]

\[\emptyset \vdash_{\text{sub}} p(a) = p(a)[b \mapsto a]\]

\[\supset R\]
Properties of the sequent calculus (3)

**Theorem 2** [Cut elimination]
The \((\text{Cut})\) rule is admissible in the system without it.
Properties of the sequent calculus (3)

**Theorem 2** [Cut elimination]
The (Cut) rule is admissible in the system without it.

**Corollary 3** The sequent calculus is **consistent**, i.e. $\vdash \Delta$ can never be derived.
Relation to First-order Logic

Call a term or a predicate context **ground** if it does not contain unknowns or explicit substitutions.

Call $\Phi \vdash \Psi$ a **first-order sequent**, when $\Phi$ and $\Psi$ are ground predicate contexts.

Genzten’s sequent calculus for first-order logic:

$\phi, \Phi \vdash \Psi, \phi$ (Ax)  $\perp, \Phi \vdash \Psi$ (⊥L)

$\Phi \vdash \Psi, \phi \vdash \Psi, \Phi \vdash \Psi$ (⊤L)  $\phi, \Phi \vdash \Psi, \psi, \phi \vdash \Psi$ (⊤R)

$\phi[a \mapsto t], \Phi \vdash \Psi$ (∀L)  $\Phi \vdash \Psi, \phi$ (∀R)  ($a \not\in fn(\Phi, \Psi)$)

$\phi[a \mapsto t'], \Phi \vdash \Psi$ (≈ L)  $t' \approx t, \phi[a \mapsto t], \Phi \vdash \Psi$ (≈ R)  $\Phi \vdash \Psi, t \approx t$ (≈ R)
Relation to First-order Logic (2)

Note that:

- we write $\forall a. \phi$ for $\forall[a] \phi$;
- $[a \mapsto t]$ is capture-avoiding substitution;
- $a \notin fn(\phi)$ is ‘$a$ does not occur in the free names of $\phi$’;
- We take predicates up to $\alpha$-equivalence.
Relation to First-order Logic (2)

Note that:

- we write $\forall a.\phi$ for $\forall[a]\phi$;
- $[a \mapsto t]$ is capture-avoiding substitution;
- $a \notin fn(\phi)$ is ‘$a$ does not occur in the free names of $\phi$’;
- We take predicates up to $\alpha$-equivalence.

**Theorem 4** $\Phi \vdash \Psi$ is derivable in the sequent calculus for first-order logic, iff $\Phi \vdash_\emptyset \Psi$ is derivable in the sequent calculus for one-and-a-halfth-order logic.

So on ground terms, one-and-a-halfth-order logic is first-order logic.
Axiomatisation of one-and-a-halfth-order logic

Theory FOL extends theory SUB with the following axioms:

\[ P \cup Q \cup P = \top \quad \neg \neg P \cup P = \top \]
\[ (P \cup Q) \cup (Q \cup R) \cup (P \cup R) = \top \quad \bot \cup P = \top \]

\[ \forall[a]P \cup P[a \mapsto T] = \top \]
\[ \forall[a](P \land Q) \iff \forall[a]P \land \forall[a]Q = \top \]
\[ a \# P \rightarrow \forall[a](P \cup Q) \iff P \cup \forall[a]Q = \top \]

\[ T \approx T = \top \quad U \approx T \land P[a \mapsto T] \cup P[a \mapsto U] = \top \]

Axioms are all of the form \( \phi = \top \), which intuitively means ‘\( \phi \) is true’.

Note that this is a finite number of axioms.
Axiomatisation of one-and-a-halfth-order logic (2)

For \( \Phi \equiv \{\phi_1, \ldots, \phi_n\} \), define its **conjunctive form** \( \Phi^\land \) to be \( \top \) when \( n = 0 \), and \( \phi_1 \land \cdots \land \phi_n \) when \( n > 0 \). Analogously, define the **disjunctive form** \( \Phi^\lor \) to be \( \bot \) when \( n = 0 \), and \( \phi_1 \lor \cdots \lor \phi_n \) when \( n > 0 \).
Axiomatisation of one-and-a-halfth-order logic (2)

For $\Phi \equiv \{\phi_1, \ldots, \phi_n\}$, define its **conjunctive form** $\Phi^\wedge$ to be $\top$ when $n = 0$, and $\phi_1 \land \cdots \land \phi_n$ when $n > 0$. Analogously, define the **disjunctive form** $\Phi^\lor$ to be $\bot$ when $n = 0$, and $\phi_1 \lor \cdots \lor \phi_n$ when $n > 0$.

**Theorem 5** For all predicate contexts $\Phi$, $\Psi$ and freshness contexts $\Delta$:

$$
\Phi \vdash_\Delta \Psi \text{ is derivable} \quad \text{iff} \quad \Delta \vdash_{\text{FOL}} \Phi^\wedge \supset \Psi^\lor = \top.
$$

So sequent and equational derivability are equivalent.
Axiomatisation of one-and-a-halfth-order logic (2)

For $\Phi \equiv \{\phi_1, \ldots, \phi_n\}$, define its **conjunctive form** $\Phi^\wedge$ to be $\top$ when $n = 0$, and $\phi_1 \land \cdots \land \phi_n$ when $n > 0$. Analogously, define the **disjunctive form** $\Phi^\lor$ to be $\bot$ when $n = 0$, and $\phi_1 \lor \cdots \lor \phi_n$ when $n > 0$.

**Theorem 5** For all predicate contexts $\Phi$, $\Psi$ and freshness contexts $\Delta$:

$$\Phi \vdash_\Delta \Psi \text{ is derivable} \iff \Delta \vdash_{\text{FOL}} \Phi^\wedge \supset \Psi^\lor = \top.$$  

So sequent and equational derivability are equivalent.

**Corollary 6** Theory FOL is consistent, i.e. $\Delta \vdash_{\text{FOL}} \top = \bot$ does not hold.
Conclusions

Using nominal terms, we can:

- **accurately** represent systems with binding:
  e.g. explicit substitution and first-order logic;
- specify *novel* systems with their own mathematical interest:
  e.g. one-and-a-halfth-order logic.

One-and-a-halfth-order logic:

- makes meta-level concepts of first-order logic *explicit*;
- has a sequent calculus with *syntax-directed* rules;
- has a *semantics* in first-order logic on ground terms;
- has a *finite* equational axiomatisation;
- is the *result* of axiomatising first-order logic in nominal algebra.
Related work

Second-order logic:

- In this logic we can quantify over predicates anywhere, which makes it more expressive than one-and-a-halfth-order logic.
- On the other hand, we can easily extend theory FOL with one axiom to express the principle of induction on natural numbers:

\[ P[a \mapsto 0] \land \forall[a](P \supset P[a \mapsto \text{succ}(\text{var}(a))]) \supset \forall[a]P = T. \]

Higher-order logic (HOL):

- is type raising, while one-and-a-halfth-order logic is not: \( P[a \mapsto t] \) corresponds to \( f(t) \) in HOL, where \( f : \mathbb{T} \to \mathbb{P} \); \( P[a \mapsto t][a' \mapsto t'] \) corresponds to \( f'(t)(t') \) where \( f' : \mathbb{T} \to \mathbb{T} \to \mathbb{P} \), and so on...
- One-and-a-halfth-order logic is not a subset of HOL because of freshnesses.
Future work

- Concrete semantics for one-and-a-halfth-order logic on non-ground terms.
- Let unknowns range over *sequent derivations*, and establish a Curry-Howard correspondence (term-in-contexts as types, derivations as terms).
- Two-and-a-halfth-order logic (where you can abstract X)?
- Implementation and automation?
Future work

- Concrete semantics for one-and-a-halfth-order logic on non-ground terms.
- Let unknowns range over *sequent derivations*, and establish a Curry-Howard correspondence (term-in-contexts as types, derivations as terms).
- Two-and-a-halfth-order logic (where you can abstract X)?
- Implementation and automation?

Current status

- M.J. Gabbay, A.H.J. Mathijssen, Nominal Algebra, submitted CSL’06.
- M.J. Gabbay, A.H.J. Mathijssen, Capture-avoiding Substitution as a Nominal Algebra, submitted ICTAC’06.
- M.J. Gabbay, A.H.J. Mathijssen, One-and-a-halfth-order Logic, PPDP’06.
Just to scare you

\[
P[b \leftrightarrow c][a \leftrightarrow c] \vdash_{c \#P} P[b \leftrightarrow c][a \leftrightarrow c] \quad (Ax)
\]
\[
\forall[a] P[b \leftrightarrow c] \vdash_{c \#P} P[b \leftrightarrow c][a \leftrightarrow c] \quad (\forall L)
\]
\[
(\forall[a] P)[b \leftrightarrow c] \vdash_{c \#P} P[b \leftrightarrow a][a \leftrightarrow c] \quad (\text{StructL}) \quad (1.)
\]
\[
\forall[b] \forall[a] P \vdash_{c \#P} P[b \leftrightarrow c][a \leftrightarrow c] \quad (\forall L)
\]
\[
(\forall[b] \forall[a] P \vdash_{c \#P} \forall[c] P[b \leftrightarrow c][a \leftrightarrow c] \quad (\text{StructR}) \quad (3.)
\]
\[
\forall[b] \forall[a] P \vdash_{c \#P} \forall[a] P[b \leftrightarrow a] \quad (\text{Fresh}) \quad (4.)
\]

Side-conditions:

1. \( c \# P \vdash_{\text{sub}} \forall[a] P[b \leftrightarrow c] = (\forall[a] P)[b \leftrightarrow c] \)

2. \( c \# P \vdash c \# \forall[b] \forall[a] P \)

3. \( c \# P \vdash_{\text{sub}} \forall[c] P[b \leftrightarrow c][a \leftrightarrow c] = \forall[a] P[b \leftrightarrow a] \)

4. \( c \notin \forall[b] \forall[a] P, \forall[a] P[b \leftrightarrow a] \)
