A Proof Theory for Generic Judgments

Dale Miller
INRIA/Futurs/Saclay and École Polytechnique

Alwen Tiu
École Polytechnique and Penn State University

LICS 2003, Ottawa, Canada, 23 June 2003

Outline

1. Motivations
2. Problems with universal quantification
3. A new quantifier
4. Intuitionistic logic with ∇
5. A proof theoretic notion of definitions
6. Meta-theories
7. Example: encoding π calculus
8. Related work
9. Conclusions and future work

Motivations

- To use proof-theory as a framework for studying *computational systems*. One main challenge is to encode and reason about *abstractions* in various computational systems, e.g., π-calculus, spi-calculus, imperative programming languages, etc.
- The static structures of abstractions are encoded as λ-terms, following the tradition of higher-order abstract syntax.
- The dynamic aspects of abstractions in computation is often modelled using universally quantified judgments and eigenvariables. This interpretation can be problematic.

Two approaches to prove a universal

The universal quantifier $\forall_\tau x.B$ can be proved:

- extensionally, i.e., by proving $B[t/x]$ for all terms t of type τ. Obviously, if τ is defined inductively, this approach can use induction.

- intensionally, i.e., by proving $B[c/x]$ for a new generic constant c (an eigenvariable). Such eigenvariables generally remain unchanged during proof search.

The collapse of eigenvariables

A cut-free proof of $\forall x \forall y. P\, x\, y$ first introduces two new eigenvariables c and d and then attempts to prove $P\, c\, d$.

Eigenvariables have been used to encode names in π-calculus [Miller93], nonces in security protocols [Cervesato, et. al. 99], reference locations in imperative programming [Chirimar95], etc.

Since

$$\forall x \forall y. P\ x\ y \supset \forall z. P\ z\ z$$

is provable, it follows that the provability of $\forall x \forall y. P\, x\, y$ implies the provability of $\forall z. P\, z\, z$. That is, there is also a proof where the eigenvariables c and d are identified.

Thus, eigenvariables are unlikely to capture the proper logic behind things like nonces, references, names, etc.

A new quantifier

- $\forall$ does not handle the intensional meaning well, hence we will introduce a new quantifier, $\nabla x.B\,x$ which focuses on an intensional reading.

- To accomodate this new quantifier, we add a new context to sequents.

$$\Sigma : B_1, \ldots, B_n \longrightarrow B_0$$
$$\Downarrow$$
$$\Sigma : \sigma_1 \triangleright B_1, \ldots, \sigma_n \triangleright B_n \longrightarrow \sigma_0 \triangleright B_0$$

  Σ is a set of eigenvariables, scoped over the sequent and σ_i is a list of generic variables, locally scoped over the formula B_i.

- The expression $\sigma_i \triangleright B_i$ is called a generic judgment. Equality between judgments is defined up to renaming of local variables.

Intuitionistic logic with ∇

$$\frac{}{\Sigma : \sigma \triangleright A, \Gamma \longrightarrow \sigma \triangleright A}\ \textit{init} \qquad \frac{\Sigma : \Delta \longrightarrow \mathcal{B} \qquad \Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}{\Sigma : \Delta, \Gamma \longrightarrow \mathcal{C}}\ \textit{cut}$$

$$\frac{}{\Sigma : \sigma \triangleright \bot, \Gamma \longrightarrow \mathcal{B}}\ \bot\mathcal{L} \qquad \frac{}{\Sigma : \Gamma \longrightarrow \sigma \triangleright \top}\ \top\mathcal{R}$$

$$\frac{\Sigma : \mathcal{B}, \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}{\Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}\ c\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \mathcal{C}}{\Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}\ w\mathcal{L}$$

$$\frac{\Sigma : \sigma \triangleright B_i, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B_1 \wedge B_2, \Gamma \longrightarrow \mathcal{D}}\ \wedge\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \qquad \Sigma : \Gamma \longrightarrow \sigma \triangleright B_2}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \wedge B_2}\ \wedge\mathcal{R}$$

$$\frac{\Sigma : \sigma \triangleright B_1, \Gamma \longrightarrow \mathcal{D} \qquad \Sigma : \sigma \triangleright B_2, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B_1 \vee B_2, \Gamma \longrightarrow \mathcal{D}}\ \vee\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_i}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \vee B_2}\ \vee\mathcal{R}$$

$$\frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B \qquad \Sigma : \sigma \triangleright C, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B \supset C, \Gamma \longrightarrow \mathcal{D}}\ \supset\mathcal{L} \qquad \frac{\Sigma : \sigma \triangleright B, \Gamma \longrightarrow \sigma \triangleright C}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B \supset C}\ \supset\mathcal{R}$$

Intuitionistic logic with ∇

$$\frac{}{\Sigma : \sigma \triangleright A, \Gamma \longrightarrow \sigma \triangleright A}\ \mathit{init} \qquad \frac{\Sigma : \Delta \longrightarrow \mathcal{B} \qquad \Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}{\Sigma : \Delta, \Gamma \longrightarrow \mathcal{C}}\ \mathit{cut}$$

$$\frac{}{\Sigma : \sigma \triangleright \bot, \Gamma \longrightarrow \mathcal{B}}\ \bot\mathcal{L} \qquad \frac{}{\Sigma : \Gamma \longrightarrow \sigma \triangleright \top}\ \top\mathcal{R}$$

$$\frac{\Sigma : \mathcal{B}, \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}{\Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}\ c\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \mathcal{C}}{\Sigma : \mathcal{B}, \Gamma \longrightarrow \mathcal{C}}\ w\mathcal{L}$$

$$\frac{\Sigma : \sigma \triangleright B_i, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B_1 \wedge B_2, \Gamma \longrightarrow \mathcal{D}}\ \wedge\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \qquad \Sigma : \Gamma \longrightarrow \sigma \triangleright B_2}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \wedge B_2}\ \wedge\mathcal{R}$$

$$\frac{\Sigma : \sigma \triangleright B_1, \Gamma \longrightarrow \mathcal{D} \qquad \Sigma : \sigma \triangleright B_2, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B_1 \vee B_2, \Gamma \longrightarrow \mathcal{D}}\ \vee\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_i}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B_1 \vee B_2}\ \vee\mathcal{R}$$

$$\frac{\Sigma : \Gamma \longrightarrow \sigma \triangleright B \qquad \Sigma : \sigma \triangleright C, \Gamma \longrightarrow \mathcal{D}}{\Sigma : \sigma \triangleright B \supset C, \Gamma \longrightarrow \mathcal{D}}\ \supset\mathcal{L} \qquad \frac{\Sigma : \sigma \triangleright B, \Gamma \longrightarrow \sigma \triangleright C}{\Sigma : \Gamma \longrightarrow \sigma \triangleright B \supset C}\ \supset\mathcal{R}$$

Intuitionistic logic with ∇

$$\frac{\Sigma : (\sigma, y:\tau) \triangleright B[y/x], \Gamma \longrightarrow \mathcal{C}}{\Sigma : \sigma \triangleright \nabla_\tau x.B, \Gamma \longrightarrow \mathcal{C}}\ \nabla\mathcal{L} \qquad \frac{\Sigma : \Gamma \longrightarrow (\sigma, y:\tau) \triangleright C[y/x]}{\Sigma : \Gamma \longrightarrow \sigma \triangleright \nabla_\tau x.C}\ \nabla\mathcal{R}$$

$$\frac{\Sigma, \sigma \vdash t : \gamma \quad \Sigma : \sigma \triangleright B[t/x], \Gamma \longrightarrow \mathcal{C}}{\Sigma : \sigma \triangleright \forall_\gamma x.B, \Gamma \longrightarrow \mathcal{C}}\ \forall\mathcal{L} \qquad \frac{\Sigma, h : \Gamma \longrightarrow \sigma \triangleright B[(h\ \sigma)/x]}{\Sigma : \Gamma \longrightarrow \sigma \triangleright \forall x.B}\ \forall\mathcal{R}$$

$$\frac{\Sigma, h : \sigma \triangleright B[(h\ \sigma)/x], \Gamma \longrightarrow \mathcal{C}}{\Sigma : \sigma \triangleright \exists x.B, \Gamma \longrightarrow \mathcal{C}}\ \exists\mathcal{L} \qquad \frac{\Sigma, \sigma \vdash t : \gamma \quad \Sigma : \Gamma \longrightarrow \sigma \triangleright B[t/x]}{\Sigma : \Gamma \longrightarrow \sigma \triangleright \exists_\gamma x.B}\ \exists\mathcal{R}$$

The typing of terms follows Church's Simple Theory of Types. Formulas are given type o, and quantified variables can be of higher types, as long as the type does not contain the type o.

Dependency between eigenvariables and local variables is encoded using the technique of ∀-lifting [Paulson] or *raising* [Miller92] of the types of the eigenvariables. Example:

$$\dfrac{\dfrac{\{x_\alpha, h_{\tau\to\gamma\to\beta}\} : \Gamma \longrightarrow (a_\tau, b_\gamma) \triangleright B\ (h\, a\, b)\ b}{\{x_\alpha\} : \Gamma \longrightarrow (a_\tau, b_\gamma) \triangleright \forall_\beta y. B\ y\ b}\ \forall\mathcal{L}}{\{x_\alpha\} : \Gamma \longrightarrow (a_\tau) \triangleright \nabla_\gamma z. \forall_\beta y. B\ y\ z}\ \nabla\mathcal{R}$$

Properties of ∇

Some theorems:

$$\nabla x \neg Bx \equiv \neg \nabla x Bx \qquad \nabla x(Bx \wedge Cx) \equiv \nabla x Bx \wedge \nabla x Cx$$
$$\nabla x(Bx \vee Cx) \equiv \nabla x Bx \vee \nabla x Cx \qquad \nabla x(Bx \supset Cx) \equiv \nabla x Bx \supset \nabla x Cx$$
$$\nabla x \forall y Bxy \equiv \forall h \nabla x Bx(hx) \qquad \nabla x \exists y Bxy \equiv \exists h \nabla x Bx(hx)$$
$$\nabla x \forall y Bxy \supset \forall y \nabla x Bxy \qquad \nabla x.\top \equiv \top, \quad \nabla x.\bot \equiv \bot$$

Consequence: ∇ can always be given atomic scope within formulas.

Non-theorems:

$$\nabla x \nabla y Bxy \supset \nabla z Bzz \qquad \nabla x Bx \supset \exists x Bx$$
$$\nabla z Bzz \supset \nabla x \nabla y Bxy \qquad \forall x Bx \supset \nabla x Bx$$
$$\forall y \nabla x Bxy \supset \nabla x \forall y Bxy \qquad \exists x Bx \supset \nabla x Bx$$
$$\nabla x Bx \supset \forall x Bx \qquad \nabla x B \equiv B$$
$$\nabla x \nabla y.B\,x\,y \equiv \nabla y \nabla x.B\,x\,y$$

A proof theoretic notion of definitions

We extend the logic further by allowing a non-logical constants (predicate) to be introduced. To each predicate, we associate some *definition clauses*. We write

$$\forall \bar{x}. p\, \bar{t} \stackrel{\triangle}{=} B$$

to denote a definition clause for predicate p. Free variables in B are in the set of free variables in $\bar{t}$, which are all in $\bar{x}$. The notion of definition has been previously studied by Schroeder-Heister, Girard, Miller and McDowell. By imposing certain restriction on definitions, we can prove cut-elimination.

Introduction rules for definitions

In intuitionistic logic without ∇, the right introduction rule for a predicate A is

$$\frac{\Gamma \longrightarrow B\theta}{\Gamma \longrightarrow A}\ \mathit{def}\mathcal{R}$$

provided that there is a definition clause $\forall \bar{x}.[H \stackrel{\triangle}{=} B]$ such that $A =_{\beta\eta} H\theta$
The left introduction rule is

$$\frac{\{B\theta, \Gamma\theta \longrightarrow C\theta \mid \forall \bar{x}.[H \stackrel{\triangle}{=} B] \text{ is a definition clause and } A\theta =_{\beta\eta} H\theta\}}{A, \Gamma \longrightarrow C}\ \mathit{def}\mathcal{L}$$

Notice that: *eigenvariables can be instantiated*, and the set of premises can be empty, finite or infinite, depending on the set of solutions for the associated equational problems.

Applying definitions to judgments

To apply definition rules to a judgment given a set of definition clauses, we need to raise the definition clauses. Given a definition clause $\forall \bar{x}.H \stackrel{\triangle}{=} B$, and a list of variables $\bar{y}$, its raised form w.r.t. $\bar{y}$ is

$$\forall \bar{h}.\bar{y} \triangleright H[(\bar{h}\,\bar{y})/\bar{x}] \stackrel{\triangle}{=} \bar{y} \triangleright B[(\bar{h}\,\bar{y})/\bar{x}].$$

The right introduction rule for a judgment $\bar{y} \triangleright A$

$$\frac{\Sigma : \Gamma \longrightarrow (\bar{y} \triangleright B)\theta}{\Sigma : \Gamma \longrightarrow \bar{y} \triangleright A}\ \mathit{def}\mathcal{R}$$

where $\forall \bar{h}.\bar{y} \triangleright H \stackrel{\triangle}{=} \bar{y} \triangleright B$ is a raised definition clause and

$$\lambda \bar{y}.A =_{\beta\eta} (\lambda \bar{y}.H)\theta.$$

The left rule is given by

$$\frac{\{\Sigma\theta : (\bar{y} \triangleright B)\theta, \Gamma\theta \longrightarrow \mathcal{C}\theta\}_{B,\theta}}{\Sigma : \bar{y} \triangleright A, \Gamma \longrightarrow \mathcal{C}} \; def\mathcal{L}$$

where $\forall \bar{h}.\bar{y} \triangleright H \stackrel{\triangle}{=} \bar{y} \triangleright B$ is a raised definition clause and

$$(\lambda \bar{y}.A)\theta =_{\beta\eta} (\lambda \bar{y}.H)\theta.$$

The signature $\Sigma\theta$ is obtained from Σ by removing variables in the domain of θ, and adding free variables in the range of θ.

Notice that *the local variables $\bar{y}$ are not instantiated*.

Meta theories

Theorem 1. Cut-elimination. *Given a fixed stratified definition, a sequent has a proof if and only if it has a cut-free proof.*

Theorem 2. *Given a* noetherian *definition, the following formula is provable.*

$$\nabla x \nabla y . B\, x\, y \equiv \nabla y \nabla x . B\, x\, y.$$

Theorem 3. *If we restrict to Horn definitions (no implication and negation in the body of the definitions) then*

1. $\forall$ *and* ∇ *are interchangable in definitions,*

2. $\vdash \nabla x . B\, x \supset \forall x . B\, x$ *for noetherian definitions.*

Example: encoding π calculus

- π-calculus is a formal model for concurrency. The main entity is process. The syntax is the following:

$$P := 0 \mid \tau.P \mid x(y).P \mid \bar{x}y.P \mid (P \mid P) \mid (P + P) \mid (x)P \mid [x = y]P$$

- Processes can make transitions (*actions*), which are guided by the syntax. Actions are of the following kind: input action $x(y)$, free output action $\bar{x}y$ and bound output action $\bar{x}(y)$ and the internal action τ. The variable y in bound output denotes a “fresh” names. The internal action is represented by a constant τ.

π-calculus: one step transitions

- Operational semantics:

$$\frac{}{\bar{x}y.\mathbf{P} \xrightarrow{\bar{x}y} \mathbf{P}}\ \mathrm{OUTPUT-ACT} \qquad \frac{\mathbf{P} \xrightarrow{\alpha} \mathbf{P'}}{[x = x]\mathbf{P} \xrightarrow{\alpha} \mathbf{P'}}\ \mathrm{MATCH} \qquad \frac{\mathbf{P} \xrightarrow{\alpha} \mathbf{P'}}{(y)\mathbf{P} \xrightarrow{\alpha} (y)\mathbf{P'}}\ \mathrm{RES}, y \notin \mathrm{n}(\mathbf{P'})$$

- Encoding restriction using $\forall$ is problematic.

$$\begin{array}{lrcl} \mathrm{RES:} & (x)P \xrightarrow{\alpha} (x)Q & \stackrel{\triangle}{=} & \forall x.(P \xrightarrow{\alpha} Q) \\ \mathrm{OUTPUT-ACT:} & \bar{x}y.P \xrightarrow{\bar{x}y} P & \stackrel{\triangle}{=} & \top \\ \mathrm{MATCH:} & [x = x]P \xrightarrow{\alpha} Q & \stackrel{\triangle}{=} & P \xrightarrow{\alpha} Q \end{array}$$

- Consider the process $(y)[x = y]\bar{x}z.0$. It cannot make any transition, since y has to be "fresh".

- The following statement should be provable

$$\forall x \forall Q \forall \alpha . [((y)[x = y](\bar{x}z.0) \xrightarrow{\alpha} Q) \supset \bot]$$

- Given the encoding of restriction using $\forall$, this reduces to proving the sequent

$$\{x, z, Q, \alpha\} : \forall y.([x = y](\bar{x}z.0) \xrightarrow{\alpha} Q) \longrightarrow \bot$$

- There are at least two instantiations of variables that identify x and y:

1. $y \mapsto w$, $x \mapsto w$, $\alpha \mapsto \bar{w}z$, $Q \mapsto 0$: (wrong *scoping*)

$$\{z\} : ([w = w](\bar{w}z.0) \xrightarrow{\bar{w}z} 0) \longrightarrow \bot$$

2. $y \mapsto x$, $\alpha \mapsto \bar{x}z$, $Q \mapsto 0$: (*freshness* assumption on y is violated)

$$\{z\} : ([x = x](\bar{x}z.0) \xrightarrow{\bar{x}z} 0) \longrightarrow \bot$$

- Scoping and freshness are captured precisely by ∇:

$$\mathrm{RES}: \quad (x)P \xrightarrow{\alpha} (x)Q \quad \stackrel{\triangle}{=} \quad \nabla x.(P \xrightarrow{\alpha} Q)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{x, z, Q, \alpha\} : w \triangleright ([x = w](\bar{x}z.0) \xrightarrow{\alpha} Q) \longrightarrow \bot}\ \mathit{def}\mathcal{L}}{\{x, z, Q, \alpha\} : . \triangleright \nabla y.([x = y](\bar{x}z.0) \xrightarrow{\alpha} Q) \longrightarrow \bot}\ \nabla\mathcal{L}}{\{x, z, Q, \alpha\} : . \triangleright ((y)[x = y](\bar{x}z.0) \xrightarrow{\alpha} Q) \longrightarrow \bot}\ \mathit{def}\mathcal{L}}{\{x, z, Q, \alpha\} : \longrightarrow . \triangleright ((y)[x = y](\bar{x}z.0) \xrightarrow{\alpha} Q) \supset \bot}\ \supset\mathcal{R}$$

- The success of *def*$\mathcal{L}$ depends on the failure of unification problem

$$\lambda w.x = \lambda w.w.$$

π-calculus: encoding (bi)simulation

One-step transition relation is encoded as three different predicates

$$P \xrightarrow{A} Q \quad \text{free actions, } A : act$$
$$P \xrightharpoonup{\downarrow x} M \quad \text{input action, } \downarrow x : nm \to act,\ M : nm \to proc$$
$$P \xrightharpoonup{\uparrow x} M \quad \text{output action, } \uparrow x : nm \to act,\ M : nm \to proc$$

$$\begin{aligned} \textit{sim}\ P\ Q \triangleq\ & \forall A \forall P'\ [(P \xrightarrow{A} P') \supset \exists Q'.(Q \xrightarrow{A} Q') \wedge \textit{sim}\ P'\ Q'] \wedge \\ & \forall X \forall P'\ [(P \xrightharpoonup{\downarrow X} P') \supset \exists Q'.(Q \xrightharpoonup{\downarrow X} Q') \wedge \forall w.\textit{sim}\ (P'w)\ (Q'w)] \wedge \\ & \forall X \forall P'\ [(P \xrightharpoonup{\uparrow X} P') \supset \exists Q'.(Q \xrightharpoonup{\uparrow X} Q') \wedge \nabla w.\textit{sim}\ (P'w)\ (Q'w)] \end{aligned}$$

Note that this definition clause is not Horn, and thus illustrates the differences between $\forall$ and ∇.

Related Work

- Pitts and Gabbay's new quantifier. Both ∇ and the "new" quantifier are self-dual but ∇ is not implied by $\forall$ nor does it imply $\exists$. Pitts and Gabbay's quantifier has set theory semantics and it assumes an infinite set of names, and hence it has some extensional flavor. ∇ on the other hand, does not require any assumption on the types of quantified variables, and is multisorted.

- O'Hearn and Pym's $\forall_{new}$ (The logic of bunched implications, BSL 99). Eigenvariables are treated as resource (linear). We haven't explored further possible relations to ∇.

Conclusions

- We have shown a simple extension of intuitionistic logic by focusing on the intensional character of eigenvariables. This gives rise to a new quantifier ∇, and a richer sequent with explicit local context.
- We proved cut-elimination, and hence consistency of the logic. The logic can be extended further with a proof-theoretic notion of definitions. Cut-elimination is also satisfied by this extended system.
- We have shown an example to illustrate the use of our logic to formalize generic reasoning, and show that ∇ captures the spirit of genericity better than $\forall$.

Future Work

- Implementation. It should be straightforward, since we are in a proof-search settings. The use of raising is not problematic with unification. We are working on a prototype, written in λProlog.
- We are considering adding induction and possibly coinduction to our current framework in order to capture reasoning about infinite behaviors.
- Other proof-theoretic properties are to be studied, e.g., permutation of rules, characterization of definitions in relation to properties of ∇.
- Another interesting direction would be to look for a type theory for the intuitionistic logic with ∇, e.g., typing system a la Martin-Löf dependent type.