A Logical Framework for Reasoning with Names

Alwen Tiu
Penn State and École Polytechnique

Joint work with Dale Miller

November 17, 2003

Motivations

- Operational semantics of computation systems can often be encoded as logical theories. Computation can then be modeled as deduction inside logic. Reasoning about the computation can benefit from structural properties of deductions, e.g., cut-elimination, uniform proofs, generalizations.

- Operational semantics of most modern calculi often involves the uses of *names*, e.g., as reference to locations, place-holders for values (in imperative languages), nonces in security protocol, etc. Underlying these uses of names are the various forms of abstractions. One main challenge is to encode and reason about these variuos forms of abstractions in logic.

Encoding abstractions

- 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.

- 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 ∇

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

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

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

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

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

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

Intuitionistic logic with ∇

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

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

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

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

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

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

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 $\forall$-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

$$\frac{\Gamma \longrightarrow B\theta}{\Gamma \longrightarrow A}\ \mathit{defR}$$

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

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

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 $\bar{y} \triangleright A$

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

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

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

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. Global-signature weakening. *If the sequent* $\Sigma : \Gamma \longrightarrow \mathcal{C}$ *is provable then the sequent* $\Sigma, x : \Gamma \longrightarrow \mathcal{C}$*, where* $x \notin \Sigma$*, is provable.*

Theorem 3. Scope weakening. *If the sequent*

$$\Sigma, x : \sigma_1 \triangleright B_1, \ldots, \sigma_n \triangleright B_n \longrightarrow \sigma \triangleright C$$

is provable then the sequent

$$\Sigma : x\sigma_1 \triangleright B_1, \ldots, x\sigma_n \triangleright B_n \longrightarrow x\sigma \triangleright C$$

is provable.

Corollary 4. Local-signature weakening. *If the sequent*

$$\Sigma : \sigma_1 \rhd B_1, \ldots, \sigma_n \rhd B_n \longrightarrow \sigma \rhd C$$

is provable then the sequent

$$\Sigma : x\sigma_1 \rhd B_1, \ldots, x\sigma_n \rhd B_n \longrightarrow x\sigma \rhd C$$

is provable.

Example: encoding π calculus

- We consider encoding finite late π-calculus [Milner92].

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

- 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.

- Encoding in HOAS

π-calculus syntax	HOAS	
Names	*nm*	
Actions	*act*	
Processes	*proc*	
$\bar{x}y$	$\uparrow xy$	$\uparrow$: *nm* $\rightarrow$ *nm* $\rightarrow$ *act*
τ	τ	τ : *act*
$x(y)$	$\downarrow x$	$\downarrow$: *nm* $\rightarrow$ *nm* $\rightarrow$ *act*
$\bar{x}(y)$	$\uparrow x$	
0	0	$0 : proc$
τ.P	$\tau\, P$	τ : *proc* $\rightarrow$ *proc*
$\bar{x}y$.P	*out* $x\ y\ P$	out : *nm* $\rightarrow$ *nm* $\rightarrow$ *proc* $\rightarrow$ *proc*
$x(y)$.P	*in* $x\ \lambda y.P\, y$	in : *nm* $\rightarrow$ (*nm* $\rightarrow$ *proc*) $\rightarrow$ *proc*
P + Q	$P + Q$	$+$: *proc* $\rightarrow$ *proc*
P|Q	$P \mid Q$	$\mid$: *proc* $\rightarrow$ *proc*
$[x = y]$P	$[x = y]\, P$	$[. = .].$: *proc* $\rightarrow$ *proc* $\rightarrow$ *proc* $\rightarrow$ *proc*
(x)P	$\nu\lambda x.P\, x$	ν : (*nm* $\rightarrow$ *proc*) $\rightarrow$ *proc*

Table 1: Signatures for π-calculus

π-calculus: one step transitions

$$\frac{}{\tau.\mathtt{P} \xrightarrow{\tau} \mathtt{P}}\ \text{TAU}-\text{ACT}$$

$$\frac{}{\bar{x}y.\mathtt{P} \xrightarrow{\bar{x}y} \mathtt{P}}\ \text{OUTPUT}-\text{ACT}$$

$$\frac{}{x(z).\mathtt{P} \xrightarrow{x(w)} \mathtt{P}'}\ \text{INPUT}-\text{ACT}, w \notin \text{fn}((z)\mathtt{P})$$

$$\frac{\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'}{[x = x]\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'}\ \text{MATCH}$$

$$\frac{\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'}{\mathtt{P} + \mathtt{Q} \xrightarrow{\alpha} \mathtt{P}'}\ \text{SUM}$$

$$\frac{\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'}{\mathtt{P} \mid \mathtt{Q} \xrightarrow{\alpha} \mathtt{P}' \mid \mathtt{Q}}\ \text{PAR}, \text{bn}(\alpha) \cap \text{fn}(\mathtt{Q}) = \emptyset$$

$$\frac{\mathtt{P} \xrightarrow{\bar{x}y} \mathtt{P}' \quad \mathtt{Q} \xrightarrow{x(z)} \mathtt{Q}'}{\mathtt{P} \mid \mathtt{Q} \xrightarrow{\tau} \mathtt{P}' \mid \mathtt{Q}'[y/z]}\ \text{COM}$$

$$\frac{\mathtt{P} \xrightarrow{\bar{x}(w)} \mathtt{P}' \quad \mathtt{Q} \xrightarrow{x(w)} \mathtt{Q}'}{\mathtt{P} \mid \mathtt{Q} \xrightarrow{\tau} (w)(\mathtt{P}' \mid \mathtt{Q}')}\ \text{CLOSE}$$

$$\frac{\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'}{(y)\mathtt{P} \xrightarrow{\alpha} (y)\mathtt{P}'}\ \text{RES}, y \notin \text{n}(\mathtt{P}')$$

$$\frac{\mathtt{P} \xrightarrow{\bar{x}y} \mathtt{P}'}{(y)\mathtt{P} \xrightarrow{\bar{x}(w)} \mathtt{P}'[w/y]}\ \text{OPEN}, y \neq x, w \notin \text{fn}((y)\mathtt{P}')$$

$$\frac{}{\tau.P \xrightarrow{\tau} P}\tau \qquad \frac{P \xrightarrow{A} Q}{[x=x]P \xrightarrow{A} Q}\text{match} \qquad \frac{P \xrightarrow{A} Q}{[x=x]P \xrightarrow{A} Q}\text{match}$$

$$\frac{P \xrightarrow{A} R}{P+Q \xrightarrow{A} R}\text{sum} \qquad \frac{Q \xrightarrow{A} R}{P+Q \xrightarrow{A} R}\text{sum} \qquad \frac{P \xrightarrow{A} R}{P+Q \xrightarrow{A} R}\text{sum} \qquad \frac{Q \xrightarrow{A} R}{P+Q \xrightarrow{A} R}\text{sum}$$

$$\frac{P \xrightarrow{A} P'}{P \mid Q \xrightarrow{A} P' \mid Q}\text{par} \qquad \frac{Q \xrightarrow{A} Q'}{P \mid Q \xrightarrow{A} P \mid Q'}\text{par}$$

$$\frac{P \xrightarrow{A} M}{P \mid Q \xrightarrow{A} \lambda n(Mn \mid Q)}\text{par} \qquad \frac{Q \xrightarrow{A} N}{P \mid Q \xrightarrow{A} \lambda n(P \mid Nn)}\text{par}$$

$$\frac{\nabla n(Pn \xrightarrow{A} P'n)}{\nu n.Pn \xrightarrow{A} \nu n.P'n}\text{res} \qquad \frac{\nabla n(Pn \xrightarrow{A} P'n)}{\nu n.Pn \xrightarrow{A} \lambda m\ \nu n.(P'nm)}\text{res} \qquad \frac{\nabla y(My \xrightarrow{\uparrow xy} M'y)}{\nu y.My \xrightarrow{\uparrow x} M'}\text{ open}$$

$$\frac{}{\mathit{out}\ x\ y\ P \xrightarrow{\uparrow xy} P}\text{ out} \qquad \frac{P \xrightarrow{\downarrow x} M \qquad Q \xrightarrow{\uparrow x} N}{P \mid Q \xrightarrow{\tau} \nu n.(Mn \mid Nn)}\text{ close} \qquad \frac{P \xrightarrow{\uparrow x} M \qquad Q \xrightarrow{\downarrow x} N}{P \mid Q \xrightarrow{\tau} \nu n.(Mn \mid Nn)}\text{ close}$$

$$\frac{}{\mathit{in}\ x\ M \xrightarrow{\downarrow x} M}\text{ in} \qquad \frac{P \xrightarrow{\downarrow x} M \qquad Q \xrightarrow{\uparrow xy} Q'}{P \mid Q \xrightarrow{\tau} (My) \mid Q'}\text{ com} \qquad \frac{P \xrightarrow{\uparrow xy} P' \qquad Q \xrightarrow{\downarrow x} N}{P \mid Q \xrightarrow{\tau} P' \mid (Ny)}\text{ com}$$

Encoding one-step transitions

- We differentiate between bound transitions and free transitions:

$$\begin{array}{lcllcl}
\langle \mathtt{P} \xrightarrow{\bar{x}y} \mathtt{Q} \rangle & = & \langle \mathtt{P} \rangle \xrightarrow{\uparrow xy} \langle \mathtt{Q} \rangle & \langle \mathtt{P} \xrightarrow{\tau} \mathtt{Q} \rangle & = & \langle \mathtt{P} \rangle \xrightarrow{\tau} \langle \mathtt{Q} \rangle \\
\langle \mathtt{P} \xrightarrow{x(y)} \mathtt{Q} \rangle & = & \langle \mathtt{P} \rangle \xrightarrow{\downarrow x} \lambda y.\langle \mathtt{Q} \rangle & \langle \mathtt{P} \xrightarrow{\bar{x}(y)} \mathtt{Q} \rangle & = & \langle \mathtt{P} \rangle \xrightarrow{\uparrow x} \lambda y.\langle \mathtt{Q} \rangle
\end{array}$$

- Examples of definition clauses

$$\begin{array}{rcl}
\nu n.Pn \xrightarrow{A} \nu n.Qn & \triangleq & \nabla n(Pn \xrightarrow{A} Qn) \\
\nu y.Py \xrightarrow{\uparrow X} Q & \triangleq & \nabla y(Py \xrightarrow{\uparrow Xy} Qy) \\
\mathit{in}\ X\ M \xrightarrow{\downarrow X} M & \triangleq & \top \\
P \mid Q \xrightarrow{\tau} S \mid (T\ Y) & \triangleq & \exists X.P \xrightarrow{\uparrow XY} S \wedge Q \xrightarrow{\downarrow X} T
\end{array}$$

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

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

- Scoping and freshness are captured precisely by ∇:

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

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

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

Proposition 5. Adequacy. *Let* $\mathtt{P}$ *and* $\mathtt{Q}$ *be processes and* α *an action. Let* N *be the set of free names* $\mathrm{fn}(\mathtt{P}) \cup (\mathrm{fn}(Q) \cup \mathrm{fn}(\alpha) - \mathrm{bn}(A))$. *The transition* $\mathtt{P} \xrightarrow{\alpha} \mathtt{Q}$ *is derivable in* π*-calculus if and only if the sequent*

$$. : . \longrightarrow \bar{u} \triangleright \langle \mathtt{P} \xrightarrow{\alpha} \mathtt{Q} \rangle,$$

where $\bar{u}$ *is an enumeration of the set* N*, is provable in* $FO\lambda^{\Delta\nabla}$.

Strong bisimulation

Definition 6. [Milner et al.] A binary relation $\mathcal{S}$ on processes is a *strong simulation* if it satisfies the following requirements:

1. if $\mathtt{P} \xrightarrow{\alpha} \mathtt{P}'$ and α is a free action, then for some $\mathtt{Q}'$, $\mathtt{Q} \xrightarrow{\alpha} \mathtt{Q}'$ and $\mathtt{P}'\mathcal{S}\mathtt{Q}'$,
2. if $\mathtt{P} \xrightarrow{x(y)} \mathtt{P}'$ and $y \notin n(P, Q)$, then for some $\mathtt{Q}'$, $\mathtt{Q} \xrightarrow{x(y)} \mathtt{Q}'$ and for all w, $\mathtt{P}'[w/y]\mathcal{S}\mathtt{Q}'[w/y]$, and
3. if $\mathtt{P} \xrightarrow{\bar{x}(y)} \mathtt{P}'$ and $y \notin \mathrm{n}(\mathtt{P}, \mathtt{Q})$ then for some $\mathtt{Q}'$, $\mathtt{Q} \xrightarrow{\bar{x}(y)} \mathtt{Q}'$ and $\mathtt{P}'\mathcal{S}\mathtt{Q}'$.

The relation $\mathcal{S}$ is a *strong bisimulation* if both $\mathcal{S}$ and its inverse are simulations. The relation $\dot{\sim}$, *strong bisimilarity*, on processes is defined by $\mathtt{P} \dot{\sim} \mathtt{Q}$ if and only if there exists a bisimulation $\mathcal{S}$ such that $\mathtt{P}\mathcal{S}\mathtt{Q}$.

Strong bisimulation

The corresponding definition clause

$$
\begin{aligned}
\mathit{bisim}\ P\ Q \stackrel{\triangle}{=}\ & \forall A \forall P'\ [(P \xrightarrow{A} P') \supset \exists Q'.(Q \xrightarrow{A} Q') \wedge \mathit{bisim}\ P'\ Q'] \wedge \\
& \forall A \forall Q'\ [(Q \xrightarrow{A} Q') \supset \exists P'.(P \xrightarrow{A} P') \wedge \mathit{bisim}\ Q'\ P'] \wedge \\
& \forall X \forall P'\ [(P \xrightarrow{\downarrow X} P') \supset \exists Q'.(Q \xrightarrow{\downarrow X} Q') \wedge \forall w.\mathit{bisim}\ (P'w)\ (Q'w)] \wedge \\
& \forall X \forall Q'\ [(Q \xrightarrow{\downarrow X} Q') \supset \exists P'.(P \xrightarrow{\downarrow X} P') \wedge \forall w.\mathit{bisim}\ (Q'w)\ (P'w)] \wedge \\
& \forall X \forall P'\ [(P \xrightarrow{\uparrow X} P') \supset \exists Q'.(Q \xrightarrow{\uparrow X} Q') \wedge \nabla w.\mathit{bisim}\ (P'w)\ (Q'w)] \wedge \\
& \forall X \forall Q'\ [(Q \xrightarrow{\uparrow X} Q') \supset \exists P'.(P \xrightarrow{\uparrow X} P') \wedge \nabla w.\mathit{bisim}\ (Q'w)\ (P'w)]
\end{aligned}
$$

Theorem 7. Soundness. *Let* $\mathtt{P}$ *and* $\mathtt{Q}$ *be two processes. If the sequent*

$$. : . \longrightarrow \bar{u} \triangleright \mathit{bisim}\ \langle \mathtt{P} \rangle\ \langle \mathtt{Q} \rangle,$$

where $\bar{u}$ *are the free names in* $(\mathtt{P}, \mathtt{Q})$*, is provable in* $FO\lambda^{\Delta\nabla}$ *then* $\mathtt{P} \dot{\sim} \mathtt{Q}$*.*

Proof We show that the set

$$\mathcal{S} = \{(\mathtt{R}, \mathtt{T}) \mid . : . \longrightarrow \bar{w} \triangleright \mathit{bisim}\ \langle \mathtt{R} \rangle\ \langle \mathtt{T} \rangle\}$$

where $\bar{w}$ are free names in $(\mathtt{R}, \mathtt{T})$, is a strong bisimulation.

Conjecture 1. $\mathcal{S}$ *is the largest bisimulation.*

Distinction and Strong Equivalence

Definition 8. [Milner] $\mathtt{P}$ and $\mathtt{Q}$ are *strongly equivalent*, written $\mathtt{P} \sim \mathtt{Q}$, if $\mathtt{P}\theta \dot{\sim} \mathtt{Q}\theta$ for all (free name) substitution θ.

Theorem 9. *If $\forall \bar{u}.$bisim $\mathtt{P}$ $\mathtt{Q}$, where $\bar{u}$ are the free names in $\mathtt{P}$ and $\mathtt{Q}$, is provable then $\mathtt{P} \sim \mathtt{Q}$.*

Definition 10. *Distinction.* [Milner] A *distinction* is a symmetric irreflexive relation between names. Let D be a distinction. A substitution θ *respects* D if, for all $(x, y) \in D$, $x\theta \neq y\theta$.

Definition 11. $\mathtt{P}$ and $\mathtt{Q}$ are *strongly D-equivalent*, written $\mathtt{P} \sim_D \mathtt{Q}$, if $\mathtt{P}\theta \sim_D \mathtt{Q}\theta$ for all substitution θ respecting D.

Theorem 12. *Let D be a finite distinction, and let $\bar{u}$ be the list of distinct names in D. If $\nabla\bar{u}\forall\bar{w}$.bisim $\mathtt{P}$ $\mathtt{Q}$, where $\bar{w}$ are free names in $(\mathtt{P},\mathtt{Q})$ not already in $\bar{u}$, is provable, then $\mathtt{P} \sim_D \mathtt{Q}$.*

Conjecture 2. *Completeness of the above encodings.*

Examples: let $P = (x|\bar{y}|\bar{z})$ and let

$$Q = (x.\bar{y}.\bar{z} + x.\bar{z}.\bar{y} + \bar{y}.x.\bar{z} + \bar{y}.\bar{z}.x + \bar{z}.x.\bar{y} + \bar{z}.\bar{y}.x)$$

then

$$P \dot{\sim} Q, \qquad P \sim_{\{x,y\}} (Q + \tau.\bar{y}), \qquad P \sim (Q + \tau.\bar{y} + \tau.\bar{z}).$$

The corresponding judgments are provable.

$$\nabla x \nabla y \nabla z.\mathit{bisim}\ P\ Q \qquad \nabla x \nabla y \forall z.\mathit{bisim}\ P\ (Q + \tau.\bar{y})$$

$$\forall x \forall y \forall z.\mathit{bisim}\ P\ (Q + \tau.\bar{y} + \tau.\bar{z})$$

Summary

- The notion of abstractions in computation systems can be given a logical interpreation via appropriate use of proof-level binders. This gives a declarative readings of the specifications of computation systems and its properties.

- We are modeling computations with deductions. This allows use to use cut (modus ponen) and cut-elimination to aid the reasoning process. In certain cases, it is possible to exploit uniformity in proof search for automatic verification.

- The use of eigenvariables and $def\mathcal{L}$ rules gives us a lazy way to perform computation. This can potentially be applied to other areas such

as symbolic bisimulations, symbolic traces analysis (e.g., [Hennessy, Boreale]), etc.

- Future work: extend $FO\lambda^{\Delta\nabla}$ with induction and co-induction.