A Proof Theory for Dual Nominal Quantifiers

Alwen Tiu

Joint work with Ross Horne, Bogdan Aman, Gabriel Ciobanu

Research School of Computer Science, The Australian National University

Second SYSMICS Workshop, Vienna, 2018

Motivations

- Relating logic and process calculus (in particular π calculus and extensions).
- Logical implication as pre-order on processes (protocols).
  - It should at least be sound w.r.t. trace inclusion.
- Applications:
  - Process refinement via implication
  - Leveraging theorem proving tools for protocol safety (secrecy) verification.

Approaches to relating process calculus and logic

- Proofs as processes: Abramsky (1994), Bellin and Scott (1994), Caires and Pfenning (2010), Wadler (2012).
- Processes as formulas: Miller (1992), Bruscoli (2002), Horne and Tiu (2016).
- Deep embedding: processes as terms, operational semantics as logical theories.
  - Pfenning (2002), Miller and Tiu (2003), Bengtson et. al. (2009).
  - Nominal quantifiers (Miller and Tiu's ∇, Gabbay and Pitts's $\mathcal{И}$) provide the interpretation of the restriction operator.

A proof-theoretic semantic for processes

- The meaning of a process is defined as the set of 'tests' it can successfully pass [Miller '92].

$$\mathcal{I}(P) = \{T \mid \quad \vdash [\![P]\!] \parr [\![T]\!] \quad \text{cut-free provable}\}$$

- Cut elimination relates tests: if $P \multimap Q$ then every test P can pass, Q can also pass.

$$\frac{\vdash P^{\perp}, Q \quad \vdash P, T}{\vdash Q, T}\ \mathit{cut}$$

  By cut elimination $\vdash Q \parr T$ has a cut-free proof.

Processes as formulas

CCS in BV (Bruscoli 2002)

Processes:

$$\begin{aligned} [\![0]\!] &= \mathrm{I} \\ [\![\alpha.P]\!] &= [\![\alpha]\!] \triangleleft [\![P]\!] \\ [\![P||Q]\!] &= [\![P]\!] \parr [\![Q]\!] \end{aligned}$$

Action prefix:

$$[\![a]\!] = a \qquad [\![\bar{a}]\!] = a^{\perp} \qquad [\![\tau]\!] = \mathrm{I}$$

- $\triangleleft$ is a self-dual non-commutative connective
- BV = MLL + $\triangleleft$ + mix + mix0
- I is the unit for $\otimes$, $\parr$ and $\triangleleft$

Embedding the π-calculus

Encoding the restriction operator:

$$[\![\nu x.P]\!] = Qx.\,[\![P]\!]$$

where Q is a quantifer.

But which quantifier?

Guiding principle

Implication should at least be sound with respect to a (reasonable) process pre-order.

$$[\![P]\!] \multimap [\![Q]\!] \quad \text{implies} \quad P \sqsubseteq Q$$

Completed trace preorder

- P has a completed trace σ if $P \stackrel{\sigma}{\Longrightarrow} P'$ and P' is a *successfully terminated process*.
  - Distinguish 0 (successful termination) and $\nu x.\bar{x}x$ (unsuccessful).
  - Essentially this means $P' \equiv 0$.
- Completed trace preorder: $P \sqsubseteq Q$ iff every completed trace of P is also a completed trace of Q.

Diagonalisation property

Interpreting ν as $\forall$ is problematic:

Let $P = \nu x \nu y.\bar{a}x.\bar{a}y$ and $Q = \nu z.\bar{a}z.\bar{a}z$.

We have:

$$[\![P]\!] = \forall x.\forall y.(ax^{\perp} \triangleleft a^{\perp}y) \multimap \forall z.(az^{\perp} \triangleleft \neg az) = [\![Q]\!]\,.$$

(Generally, $\forall x \forall y.F(x,y) \multimap \forall z.F(z,z)$.)

But P and Q are not related by the trace pre-order.

Distributivity over *par*

Nominal quantiifers (e.g., Gabbay-Pitts's $\mathrm{И}$ and Miller-Tiu's ∇) avoids the diagonalisation property, but distribute over *all propositional connectives*, e.g.:

$$\nabla x(P \parr Q) \multimap (\nabla x.P) \parr (\nabla x.Q)$$

Encoding ν using ∇ is problematic:

- Let $P = \nu x.(\bar{a}x \parallel \bar{a}x)$ and $Q = (\nu x.\bar{a}x) \parallel (\nu x.\bar{a}x)$
- P and Q are not related in the trace pre-order, but $[\![P]\!] \multimap [\![Q]\!]$.

Dual nominal quantifiers

- The self-dual nominal quantifier ∇ is split into a dual pair: $И$ ('new') and $Э$ ('wen').
  - Suggested by Alessio Guglielmi in 2004 in Proof Theory mailing list.
- Think of one as 'generating' fresh name ($И$), and the other as 'consuming a fresh name' ($Э$).
- This has a natural interpretation in Sangiorgi's internal π calculus.

System BV1: Syntax

Syntax of formulas:

$$
\begin{array}{rrlrlrl}
P ::= & \alpha & \text{(atom)} & \alpha^{\perp} & \text{(co-atom)} & \mathrm{I} & \text{(unit)} \\
 & P \parr P & \text{(par)} & P \otimes P & \text{(times)} & P \triangleleft P & \text{(seq)} \\
 & \forall x.P & \text{(all)} & \exists x.P & \text{(some)} & & \\
 & \text{И}x.P & \text{(new)} & \text{Э}x.P & \text{(wen)} & &
\end{array}
$$

Linear implication encoded as: $P \multimap Q \equiv P^{\perp} \parr Q$.

Congruence:

- $(P, \parr, \mathrm{I})$ and $(P, \otimes, \mathrm{I})$ are commutative monoids,
- $(P, \triangleleft, \mathrm{I})$ is a monoid.
- Equivariance:

$$\text{И}x.\text{И}y.P \equiv \text{И}y.\text{И}x.P \qquad \text{Э}x.\text{Э}y.P \equiv \text{Э}y.\text{Э}x.P$$

Equivariance is a design choice; does not affect cut-elimination.

System BV1: inference rules

Inference rules are presented as using the calculus of structures (essentially rewrite rules).

BV fragment:

$$\begin{array}{rcll} C\{\, \alpha^{\perp} \parr \alpha \,\} & \longrightarrow & C\{\, \mathrm{I} \,\} & \text{(atomic interaction)} \\ C\{\, P \parr (Q \otimes S) \,\} & \longrightarrow & C\{\, (P \parr Q) \otimes S \,\} & \text{(switch)} \\ C\{\, (P \triangleleft Q) \parr (R \triangleleft S) \,\} & \longrightarrow & C\{\, (P \parr R) \triangleleft (Q \parr S) \,\} & \text{(sequence)} \end{array}$$

First-order quantifiers:

$$\begin{array}{rcll} C\{\, \forall x.P \parr R \,\} & \longrightarrow & C\{\, \forall x.(P \parr R) \,\} \quad \text{only if } x \# R & \text{(extrude1)} \\ C\{\, \forall x.\mathrm{I} \,\} & \longrightarrow & C\{\, \mathrm{I} \,\} & \text{(tidy1)} \\ C\{\, \forall x.(P \triangleleft S) \,\} & \longrightarrow & C\{\, \forall x.P \triangleleft \forall x.S \,\} & \text{(medial1)} \\ C\{\, \exists x.P \,\} & \longrightarrow & C\{\, P\{^{v}/_{x}\} \,\} & \text{(select1)} \end{array}$$

System BV1: inference rules

Nominal quantifiers:

$$\mathcal{C}\{\, И x.P \parr Э x.Q \,\} \longrightarrow \mathcal{C}\{\, И x.(P \parr Q) \,\} \qquad \text{(close)}$$

$$\mathcal{C}\{\, И x.P \parr R \,\} \longrightarrow \mathcal{C}\{\, И x.(P \parr R) \,\} \quad \text{only if } x \# R \qquad \text{(extrude new)}$$

$$\mathcal{C}\{\, И x.(P \triangleleft S) \,\} \longrightarrow \mathcal{C}\{\, И x.P \triangleleft И x.S \,\} \qquad \text{(medial new)}$$

$$\mathcal{C}\{\, И x.\mathrm{I} \,\} \longrightarrow \mathcal{C}\{\, \mathrm{I} \,\} \qquad \text{(tidy name)}$$

$$\mathcal{C}\{\, Э x.P \,\} \longrightarrow \mathcal{C}\{\, И x.P \,\} \qquad \text{(fresh)}$$

$$\mathcal{C}\{\, И x.Э y.P \,\} \longrightarrow \mathcal{C}\{\, Э y.И x.P \,\} \qquad \text{(new wen)}$$

$$\mathcal{C}\{\, Э x.P \odot Э x.S \,\} \longrightarrow \mathcal{C}\{\, Э x.(P \odot S) \,\} \quad \text{where } \odot \in \{\parr, \triangleleft\} \qquad \text{(suspend name)}$$

$$\mathcal{C}\{\, \forall x.Ↄ y.P \,\} \longrightarrow \mathcal{C}\{\, Ↄ y.\forall x.P \,\} \quad \text{for } Ↄ \in \{И, Э\} \qquad \text{(all name)}$$

Some theorem and non-theorems

Provable:

- $\forall x.P \multimap И x.P$, $И x.P \multimap Э x.P$, $Э x.P \multimap \exists x.P$
- Scope contraction: $И x.(P \parr Q) \multimap И x.P \parr Q$, where $x \# Q$

Not provable:

- Diagonalization: $И x И y.Pxy \multimap И z.Pzz$.
- Distribution over par:

$$И x.P \parr И x.Q \multimap И x.(P \parr Q) \qquad И x.(P \parr Q) \multimap И x.P \parr И x.Q$$

- Scope expansion: $И x.P \parr Q \multimap И x.(P \parr Q)$, where $x \# Q$.

Cut elimination

- The cut rule: $\mathcal{C}\{\ \mathrm{I}\ \} \longrightarrow \mathcal{C}\{\ P \otimes P^{\perp}\ \}$
- Cut elimination proof uses the 'splitting' lemma:

  if $\vdash (P \otimes P^{\perp}) \parr R$ then R can be partitioned into $U \parr V$ s.t. $\vdash P \parr U$ and $\vdash P^{\perp} \parr V$.
- With nominal quantifiers, the splitting lemma becomes a bit more complicated:

  If $\vdash (P \otimes Q) \parr R$, then there exist V and W s.t. $\vdash P \parr V$ and $\vdash Q \parr W$, and killing context $\mathcal{T}\{\ \}$ such that $R \longrightarrow \mathcal{T}\{\ V \parr W\ \}$ and if x appears in $\mathcal{T}\{\ \}$ then $x \# (P \otimes Q)$.

- Killing contexts:

$$\mathcal{T}\{\ \} ::= \{\ \cdot\ \} \mid \forall x.\mathcal{T}\{\ \} \mid \text{И}x.\mathcal{T}\{\ \}$$

A finite π-calculus

Syntax of processes:

$$P ::= 1 \mid \nu x.P \mid x(y).P \mid \overline{x}y.P \mid P \parr P.$$

Actions: $A ::= \tau \mid \overline{x}[z] \mid \overline{x}z \mid x(z)$

$\overline{x}[z]$ denote fresh output.

$$\frac{}{x(y).P \xrightarrow{x(y)} P} \qquad \frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \qquad \frac{P \xrightarrow{\overline{x}z} Q}{\nu z.P \xrightarrow{\overline{x}[z]} Q}\ x \neq z$$

$$\frac{P \xrightarrow{A} Q}{\nu x.P \xrightarrow{A} \nu x.Q}\ x \notin \mathrm{n}(A) \qquad \frac{P \xrightarrow{A} Q}{P \parr R \xrightarrow{A} Q \parr R}\ \begin{array}{l}\text{if } A = \overline{x}[z] \\ \text{or } A = x(z) \\ \text{then } z \# R\end{array}$$

$$\frac{P \xrightarrow{\overline{x}[z]} P' \quad Q \xrightarrow{x(z)} Q'}{P \parr Q \xrightarrow{\tau} \nu z.(P' \parr Q')} \qquad \frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \parr Q \xrightarrow{\tau} P' \parr Q'\{^{y}/_{z}\}}$$

Processes as predicates

$$[\![1]\!]_\pi = \mathrm{I} \qquad [\![P \parr Q]\!]_\pi = [\![P]\!]_\pi \parr [\![Q]\!]_\pi \qquad [\![\nu x.P]\!]_\pi = И x.[\![P]\!]_\pi$$

$$[\![x(z).P]\!]_\pi = \exists z.(xz \triangleleft [\![P]\!]_\pi) \qquad [\![\overline{x}z.P]\!]_\pi = \overline{xz} \triangleleft [\![P]\!]_\pi$$

Internal π calculus

$$P ::= 1 \quad \text{(success)}$$
$$\mid \nu x.P \quad \text{(nu)}$$
$$\mid \overline{x}[z].P \quad \text{(private input)}$$
$$\mid x[z].P \quad \text{(private output)}$$
$$\mid P \parr P \quad \text{(par)}$$

$$A ::= \tau \mid \overline{x}[z] \mid x[z] \quad \text{(actions)}$$

$$\frac{}{\overline{x}[z].P \xrightarrow{\overline{x}[z]} P} \qquad \frac{}{x[z].P \xrightarrow{\overline{x}[z]} P}$$

$$\frac{P \xrightarrow{A} Q}{\nu x.P \xrightarrow{A} \nu x.Q} \; x \notin \mathrm{n}(A)$$

$$\frac{P \xrightarrow{A} Q}{P \parr R \xrightarrow{A} Q \parr R} \; \begin{array}{l}\text{if } A = \overline{x}[z] \\ \text{or } A = x[z] \\ \text{then } z \# R\end{array}$$

$$\frac{P \xrightarrow{\overline{x}[z]} P' \quad Q \xrightarrow{x[z]} Q'}{P \parr Q \xrightarrow{\tau} \nu z.(P' \parr Q')}$$

The process $x[z]\,P$ can only receive a fresh name, not arbitrary name.

Encoding the internal π calculus

$$[\![1]\!]_{\pi l} = \mathrm{I} \qquad [\![P \parr Q]\!]_{\pi l} = [\![P]\!]_{\pi l} \parr [\![Q]\!]_{\pi l} \qquad [\![\nu x.P]\!]_{\pi l} = И x.[\![P]\!]_{\pi l}$$

$$[\![x[z]\,.P]\!]_{\pi l} = Э z.(xz \triangleleft [\![P]\!]_{\pi l}) \qquad [\![\overline{x}[z]\,.P]\!]_{\pi l} = И z.(\overline{xz} \triangleleft [\![P]\!]_{\pi l})$$

Completed traces

- In (internal) π-calculus, a trace may have implicit name binding, e.g., in

$$ax.\bar{b}[y].cy.d[z].\bar{e}z.$$

  y is a binder whose scope is over cy, and z is a binder whose scope is $\bar{e}z$.
- Since traces will be used as a test, its encoding in logic is dualized.
- Fresh names are encoded using the $\Im$ quantifier, e.g., the above trace is encoded as:

$$\bar{a}x \triangleleft \Im y.(by \triangleleft \bar{c}y \triangleleft \mathrm{И}d.(\bar{d}z \triangleleft ez)).$$

Adequacy results

To prove soundness of implication w.r.t. completed trace pre-order, we need to show:

Soundness If $\vdash [\![P]\!] \parr [\![\sigma]\!]$ is cut-free provable then σ is a completed trace of P.

Completeness If σ is a completed trace of P then $\vdash [\![P]\!] \parr [\![\sigma]\!]$ is provable.

Prefixing vs sequential composition

- A mismatch between prefixing and its encoding as sequential compositon:

$$\bar{a}x.(\bar{b}y \parallel b(z)) \not\longrightarrow \bar{a}x.0$$

but

$$ax^{\perp} \triangleleft (by^{\perp} \parr \exists z.bz) \longrightarrow ax^{\perp} \triangleleft (by^{\perp} \parr by) \longrightarrow ax^{\perp} \triangleleft \mathrm{I}$$

- If an encoding of a process-trace pair $[\![P]\!] \parr T$ is provable, there can be many proofs that do not correspond to reduction in π-calculus.
  - Show that any proof of $[\![P]\!] \parr T$ can be transformed into a 'canonical proof' that corresponds to reduction.

Left contexts

- Given a BV1 proof of an encoding of a process, transform it so that interaction rules are applied only in *left contexts*.
- Left contexts:

$$\mathcal{L}\{\ \cdot\ \} ::= \{\ \cdot\ \} \mid \mathcal{L}\{\ \cdot\ \} \triangleleft P \mid \mathcal{L}\{\ \cdot\ \} \parr P \mid \mathcal{L}\{\ \cdot\ \} \otimes P \mid \Pi x.\mathcal{L}\{\ \cdot\ \} \mid \forall x.\mathcal{L}\{\ \cdot\ \}$$

- General strategy: permute interaction up, and restrict them to left context.

Extracting transitions from proofs

Lemma (progress)

For traces T, and π-calculus process R, if $\vdash T \parr [\![R]\!]$ then at least one of the following holds:

- $T = 1$ *and* R *successfully terminates.*
- $R = \mathcal{P}^0\{\, \mathcal{P}^1\{\, \overline{y}z.P \,\} \parr \mathcal{P}^2\{\, y(x).Q \,\}\,\}$, *where* y *and* z *are not bound by* $\mathcal{P}^1\{\ \}$ *or* $\mathcal{P}^2\{\ \}$, *and the following holds:* $\vdash T \parr [\![\mathcal{P}^0\{\, \mathcal{P}^1\{\, P \,\} \parr \mathcal{P}^2\{\, Q\{^z/_x\} \,\}\,\}]\!]$.
- $R = \mathcal{P}^0\{\, \mathcal{P}^1\{\, \mathit{И}z.\mathcal{P}^2\{\, \overline{x}z.P \,\}\,\} \parr \mathcal{P}^3\{\, x(z).Q \,\}\,\}$, *where* x *and* z *are not bound by* $\mathcal{P}^1\{\ \}$, $\mathcal{P}^2\{\ \}$, *or* $\mathcal{P}^3\{\ \}$, z *is fresh for all processes appearing in the contexts* $\mathcal{P}^1\{\ \}$ *and* $\mathcal{P}^3\{\ \}$, *and the following holds:* $\vdash T \parr [\![\mathcal{P}^0\{\, \mathit{И}z.(\mathcal{P}^1\{\, \mathcal{P}^2\{\, P \,\}\,\} \parr \mathcal{P}^3\{\, Q \,\})\,\}]\!]_\pi$.
- .....

Extracting πI-calculus Transitions from Proof

$$
\begin{array}{lll}
\left[\!\left[a[x].b[y] \parallel \overline{a}[x] \parallel \overline{b}[y] \right]\!\right]_{\pi I} = & \Im x.(ax \triangleleft \Im y.by) \parr И x.\overline{ax} \parr И y.\overline{by} & \\
\longrightarrow & \Im y.\Im x.(ax \triangleleft by) \parr И x.\overline{ax} \parr И y.\overline{by} & \text{(left wen + equivariance} \\
\longrightarrow & И y.\left(\Im x.(ax \triangleleft by) \parr \overline{by}\right) \parr И x.\overline{ax} & \text{(close)} \\
\longrightarrow & И y.\Im x.\left((ax \triangleleft by) \parr \overline{by}\right) \parr И x.\overline{ax} & \text{(extrude name)} \\
\longrightarrow & И y.\Im x.\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr И x.\overline{ax} & \text{(sequence)} \\
\longrightarrow & И y.\Im x.ax \parr И x.\overline{ax} & \text{(atomic interaction)} \\
\longrightarrow & \Im x.И y.ax \parr И x.\overline{ax} & \text{(new wen)} \\
\longrightarrow & И x.(И y.ax \parr \overline{ax}) & \text{(close)} \\
\longrightarrow & И x.И y.(ax \parr \overline{ax}) & \text{(extrude name)} \\
\longrightarrow & И x.И y.\mathrm{I} & \text{(atomic interaction)} \\
\longrightarrow & И x.\mathrm{I} & \text{(tidy name)} \\
\longrightarrow & \mathrm{I} & \text{(tidy name)}
\end{array}
$$

Interacting atoms marked red are not in left context.

Extracting πI-calculus Transitions from Proof

$$
\begin{array}{lll}
\left[\!\left[a[x].b[y] \parallel \overline{a}[x] \parallel \overline{b}[y] \right]\!\right]_{\pi I} = & \Im x.(ax \triangleleft \Im y.by) \parcel И x.\overline{ax} \parcel И y.\overline{by} & \\
\longrightarrow & \Im y.\Im x.(ax \triangleleft by) \parcel И x.\overline{ax} \parcel И y.\overline{by} & \text{(left wen + equivariance} \\
\longrightarrow & И y.\left(\Im x.(ax \triangleleft by) \parcel \overline{by}\right) \parcel И x.\overline{ax} & \text{(close)} \\
\longrightarrow & И y.\Im x.\left((ax \triangleleft by) \parcel \overline{by}\right) \parcel И x.\overline{ax} & \text{(extrude name)} \\
\longrightarrow & И y.\Im x.\left(ax \triangleleft \left(by \parcel \overline{by}\right)\right) \parcel И x.\overline{ax} & \text{(sequence)} \\
\longrightarrow & \Im x.И y.\left(ax \triangleleft \left(by \parcel \overline{by}\right)\right) \parcel И x.\overline{ax} & \text{(new wen)} \\
\longrightarrow & И x.\left(И y.\left(ax \triangleleft \left(by \parcel \overline{by}\right)\right) \parcel \overline{ax}\right) & \text{(close)} \\
\longrightarrow & И x.И y.\left(\left(ax \triangleleft \left(by \parcel \overline{by}\right)\right) \parcel \overline{ax}\right) & \text{(extrude name)} \\
\longrightarrow & И x.И y.\left((ax \parcel \overline{ax}) \triangleleft \left(by \parcel \overline{by}\right)\right) & \text{(sequence)} \\
\longrightarrow & И x.И y.\left(by \parcel \overline{by}\right) & \text{(atomic interaction)} \\
\longrightarrow & И x.И y.\mathrm{I} & \text{(atomic interaction)} \\
\longrightarrow & И x.\mathrm{I} & \text{(tidy name)} \\
\longrightarrow & \mathrm{I} & \text{(tidy name)}
\end{array}
$$

Red atoms are permutted up.

Blue atoms now interact first.

All atoms interact in a left context.

Extracting πI-calculus Transitions from Proof

$$
\begin{array}{lll}
\left[\!\left[a[x].b[y] \parallel \overline{a}[x] \parallel \overline{b}[y] \right]\!\right]_{\pi I} = & \exists x.(ax \triangleleft \exists y.by) \parr \forall x.\overline{ax} \parr \forall y.\overline{by} & \\
\longrightarrow & \exists y.\exists x.(ax \triangleleft by) \parr \forall x.\overline{ax} \parr \forall y.\overline{by} & \text{(left wen + equivariance)} \\
\longrightarrow & \forall y.\left(\exists x.(ax \triangleleft by) \parr \overline{by}\right) \parr \forall x.\overline{ax} & \text{(close)} \\
\longrightarrow & \forall y.\exists x.\left((ax \triangleleft by) \parr \overline{by}\right) \parr \forall x.\overline{ax} & \text{(extrude name)} \\
\longrightarrow & \forall y.\exists x.\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \forall x.\overline{ax} & \text{(sequence)} \\
\longrightarrow & \exists x.\forall y.\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \forall x.\overline{ax} & \text{(new wen)} \\
\longrightarrow & \forall x.\left(\forall y.\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \overline{ax}\right) & \text{(close)} \\
\longrightarrow & \forall x.\forall y.\left(\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \overline{ax}\right) & \text{(extrude name)} \\
\longrightarrow & \forall x.\forall y.\left((ax \parr \overline{ax}) \triangleleft \left(by \parr \overline{by}\right)\right) & \text{(sequence)} \\
\longrightarrow & \forall x.\forall y.\left(by \parr \overline{by}\right) & \text{(atomic interaction)} \\
\longrightarrow & \forall x.\forall y.\mathrm{I} & \text{(atomic interaction)} \\
\longrightarrow & \forall x.\mathrm{I} & \text{(tidy name)} \\
\longrightarrow & \mathrm{I} & \text{(tidy name)}
\end{array}
$$

Identify atoms of first interaction.

Identify guarding nominal quantifiers.

Extracting πI-calculus Transitions from Proof

$$
\begin{array}{lll}
 & \mathit{И}x.((ax \triangleleft \mathit{Э}y.by) \parr \overline{ax}) \parr \mathit{И}y.\overline{by} & \\
\longrightarrow & \mathit{И}x.(\mathit{Э}y.(ax \triangleleft by) \parr \overline{ax}) \parr \mathit{И}y.\overline{by} & \text{(left wen)} \\
\longrightarrow & \mathit{И}x.\left(\mathit{Э}y.(ax \triangleleft by) \parr \overline{ax} \parr \mathit{И}y.\overline{by}\right) & \text{(extrude name)} \\
\longrightarrow & \mathit{И}x.\left(\mathit{И}y.\left((ax \triangleleft by) \parr \overline{by}\right) \parr \overline{ax}\right) & \text{(close)} \\
\longrightarrow & \mathit{И}x.\left(\mathit{И}y.\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \overline{ax}\right) & \text{(sequence)} \\
\longrightarrow & \mathit{И}x.\mathit{И}y.\left(\left(ax \triangleleft \left(by \parr \overline{by}\right)\right) \parr \overline{ax}\right) & \text{(extrude name)} \\
\longrightarrow & \mathit{И}x.\mathit{И}y.\left((ax \parr \overline{ax}) \triangleleft \left(by \parr \overline{by}\right)\right) & \text{(sequence)} \\
\longrightarrow & \mathit{И}x.\mathit{И}y.\left(by \parr \overline{by}\right) & \text{(atomic interaction)} \\
\longrightarrow & \mathit{И}x.\mathit{И}y.\mathrm{I} & \text{(atomic interaction)} \\
\longrightarrow & \mathit{И}x.\mathrm{I} & \text{(tidy name)} \\
\longrightarrow & \mathrm{I} & \text{(tidy name)}
\end{array}
$$

Reorganise nominal quantifiers in conclusion and proof.

Extracting πI-calculus Transitions from Proof

$$
\begin{array}{lll}
\left[\!\left[\nu x.\,(b[y] \parallel 1) \parallel \overline{b}[y]\right]\!\right]_{\pi I} & \text{И}x.\text{Э}y.by \parr \text{И}y.\overline{by} & \\
\longrightarrow & \text{И}x.\text{Э}y.\Big(by \parr \text{И}y.\overline{by}\Big) & \text{(extrude name)} \\
\longrightarrow & \text{И}x.\text{Э}y.\text{I} & \text{(atomic interaction)} \\
\longrightarrow & \text{И}x.\text{I} & \text{(tidy name)} \\
\longrightarrow & \text{I} & \text{(tidy name)}
\end{array}
$$

Strip first interacting atoms.

The conclusion becomes a transition reachable from the first process by a τ transition.

Extracting πI-calculus Transitions from Proof

$$\begin{array}{rcll} [\![\nu y.(\nu x.(1 \parallel 1) \parallel 1)]\!]_{\pi I} = & \Im y.И x.\mathrm{I} & \\ \longrightarrow & \Im y.\mathrm{I} & \text{(tidy name)} \\ \longrightarrow & \mathrm{I} & \text{(tidy name)} \end{array}$$

Similar step result in another τ transition to the above proof.

Thereby we have extracted the following πI-calculus transitions.

$$a[x].b[y] \parallel \overline{a}[x] \parallel \overline{b}[y] \xrightarrow{\tau} \nu x.(b[y] \parallel 1) \parallel \overline{b}[y] \xrightarrow{\tau} \nu y.(\nu x.(b[y] \parallel 1) \parallel 1)$$

The same proof structure can also be used to extract the following π-calculus transitions.

$$\nu x.\overline{a}x \parr \nu y.\overline{b}y \xrightarrow{\overline{a}[x]} 1 \parallel \nu y.\overline{b}y \xrightarrow{\overline{b}[y]} 1 \parallel 1$$

Correspondence

Proposition

If T is a completed π-calculus trace and P a π-calculus process, then if $\vdash T \parr [\![P]\!]$ then P has completed trace T.

Theorem (completed trace inclusion)

For π-calculus processes P and Q, if $\vdash [\![P]\!] \multimap [\![Q]\!]$ then, for all completed traces T, if P has completed trace T then P also has completed trace T.

Conclusion and future work

- Linear implication is sound w.r.t. to trace preorder.
  But can we be more precise? What exactly is *the* preorder it captures?
- The π-calculus with sequential composition (not prefixing); how does one handle the scope extrusion?
- Extension with non-deterministic choice and replication (work in progress).