Communication Interference in Mobile Boxed Ambients

FST&TCS 2002

joint with: M. Bugliesi, S. Crafa, M. Merro

Vladimiro Sassone

University of Sussex,
Mobile Ambients

Both administrative domains and computational environments (Cardelli-Gordon)

- Subjective movements

\[
\begin{align*}
&n[ \text{in} \ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R] \\
&m[n[\text{out} \ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]
\end{align*}
\]

- Process interaction

\[
\begin{align*}
&n[\langle M \rangle.P \mid (x).Q] \rightarrow n[P \mid Q\{x := M\}],
\end{align*}
\]

- Boundary dissolver

\[
\text{open } n.P \mid n[Q] \rightarrow P \mid Q.
\]
Interferences in Mobile Ambients

The inherent nondeterminism of movement may go wild: Grave Interferences.

\[ k[n[in \ m.P \ | \ out \ k.R] \ | \ m[Q]] \]
Interferences in Mobile Ambients

The inherent nondeterminism of movement may go wild: Grave Interferences.

\[ k[n \text{ in } m.P \mid \text{ out } k.R] \mid m[Q] \]

Introducing Safe Ambients (Levi-Sangiorgi)

\[ n \text{ in } m.P \mid Q \mid m[\text{ in } m.R \mid S] \rightarrow m[n \text{ P } \mid Q \mid R \mid S] \]

Co-capabilities and single-threadedness rule out grave interferences
Interferences in Mobile Ambients

The inherent nondeterminism of movement may go wild: Grave Interferences.

\[ k[n[\text{in } m.P \mid \text{out } k.R] \mid m[Q]] \]

Introducing **Safe Ambients** (Levi-Sangiorgi)

\[ n[\text{in } m.P \mid Q] \mid m[\text{in } m.R \mid S] \rightarrow m[n[P \mid Q] \mid R \mid S] \]

**Co-capabilities** and **single-threadedness** rule out grave interferences.

**Safe Ambients** **with passwords** have a conveniently treatable semantics. (Merro-Hennessy)

\[ n[\text{in } (m,k).P \mid Q] \mid m[\text{in } (m,k).R \mid S] \rightarrow m[n[P \mid Q] \mid R \mid S] \]
Mobile Boxed Ambients

open’s nature of ambient dissolver is a potential source of problems.

Direct communication as alternative source of expressiveness: Mobile Boxed Ambients (Bugliesi et al.). Perform I/O on a subambient $n$’s local channel (viz. $(x)^n$) as well as from the parent’s local channel (viz. $(x)^\uparrow$)

$$
(x)^n.P \mid n[⟨M⟩.Q \mid R] \longrightarrow P\{x := M\} \mid n[Q \mid R]
$$

$$
⟨M⟩.P \mid n[(x)^\uparrow.Q \mid R] \longrightarrow P \mid n[Q\{x := M\} \mid R].
$$
Mobile Boxed Ambients

open's nature of ambient dissolver is a potential source of problems.

Direct communication as alternative source of expressiveness: Mobile Boxed Ambients (Bugliesi et al.). Perform I/O on a subambient \( n \)'s local channel (viz. \( (x^n) \)) as well as from the parent's local channel (viz. \( (x) \uparrow \))

\[
(x^n).P | n[\langle M \rangle.Q | R] \longrightarrow P\{x := M\} | n[Q | R]
\]

\[
\langle M \rangle.P | n[(x) \uparrow.Q | R] \longrightarrow P | n[Q\{x := M\} | R].
\]

But it is a great source of non-local nondeterminism and communication interference.

\[
m[(x^n).P | n[\langle M \rangle | (x).Q | k[(x) \uparrow.R]]]
\]
Mobile Boxed Ambients

open’s nature of ambient dissolver is a potential source of problems.

Direct communication as alternative source of expressiveness: Mobile Boxed Ambients (Bugliesi et al.). Perform I/O on a subambient \( n \)’s local channel (viz. \((x)^n\)) as well as from the parent’s local channel (viz. \((x)^\uparrow\))

\[
(x)^n.P \mid n[\langle M \rangle.Q \mid R] \longrightarrow P\{x := M\} \mid n[Q \mid R]
\]

\[
\langle M \rangle .P \mid n[(x)^\uparrow .Q \mid R] \longrightarrow P \mid n[Q\{x := M\} \mid R].
\]

But it is a great source of non-local nondeterminism and communication interference.

\[
m[ (x)^n.P \mid n[ \langle M \rangle \mid (x).Q \mid k[(x)^\uparrow.R]] ]]\]
Introducing NBA: Communication

NBA: a fresh foundation based on: each ambient comes equipped with two mutually non-interfering channels, for local and upward communications.

\[
(x)^n.P \mid n[\langle M \rangle^\wedge. Q \mid R] \longrightarrow P\{x := M\} \mid n[Q \mid R]
\]

\[
\langle M \rangle^n.P \mid n[(x)^\wedge. Q \mid R] \longrightarrow P \mid n[Q\{x := M\} \mid R]
\]
Introducing NBA: Communication

NBA: a fresh foundation based on: each ambient comes equipped with two mutually non-interfering channels, for local and upward communications.

\[(x)^n.P \mid n[\langle M \rangle]^{\hat{z}}.Q \mid R \rightarrow P\{x := M\} \mid n[Q \mid R]\]

\[\langle M \rangle^n.P \mid n[(x)^{\hat{z}}].Q \mid R \rightarrow P \mid n[Q\{x := M\} \mid R]\]
Introducing NBA: Communication

NBA: a fresh foundation based on: each ambient comes equipped with two mutually non-interfering channels, for local and upward communications.

\[(x)^n.P \mid n[\langle M \rangle^\hat{\cdot}.Q \mid R] \longrightarrow P\{x := M\} \mid n[Q \mid R]\]

\[\langle M \rangle^n.P \mid n[(x)^\hat{\cdot}.Q \mid R] \longrightarrow P \mid n[Q\{x := M\} \mid R]\]

- Good algebraic laws; simple type system;
- Expressiveness??
**Introducing NBA: Communication**

NBA: a fresh foundation based on: each ambient comes equipped with two mutually non-interfering channels, for **local** and **upward** communications.

\[
(x)^n.P \mid n[\langle M \rangle^\ast].Q \mid R \rightarrow P\{x := M\} \mid n[Q \mid R]
\]

\[
\langle M \rangle^n.P \mid n[(x)^\ast].Q \mid R \rightarrow P \mid n[Q\{x := M\} \mid R]
\]

- Good algebraic laws; simple type system;

- Expressiveness??

- Hmm, rather poor: \(n[P]\) cannot, for instance, communicate with children it doesn’t know statically. It can never learn about incoming ambients, and will never be able to talk to them.
Essentially, our idea is to introduce co-actions of the form \( \text{enter}(x) \) which have the effect of binding the variable \( x \).

Such a purely binding mechanism does not provide a way of control of access, but only to registers it. As a (realistic) access protocol where newly arrived agents must register themselves to be granted access to local resources.
Essentially, our idea is to introduce co-actions of the form $\text{enter}(x)$ which have the effect of binding the variable $x$.

Such a purely binding mechanism does not provide a way control of access, but only to registers it. As a (realistic) access protocol where newly arrived agents must register themselves to be granted access to local resources.

Need a finer mechanism of access control:

$$a[\text{enter}(b, k).P_1 | P_2] | b[\text{enter}(x, k).Q_1 | Q_2] \rightarrow b[a[P_1 | P_2] | Q_1\{x := a\} | Q_2]$$

This represent an access protocol where the credentials of incoming processes ($k$ in the rule above) are controlled, as a preliminary step to the registration protocol.
Introducing NBA: Mobility

Essentially, our idea is to introduce co-actions of the form $\text{enter}(x)$ which have the effect of binding the variable $x$.

Such a purely binding mechanism does not provide a way control of access, but only to registers it. As a (realistic) access protocol where newly arrived agents must register themselves to be granted access to local resources.

Need a finer mechanism of access control:

$$a[\text{enter}(b, k).P_1 | P_2] | b[\text{enter}(x, k).Q_1 | Q_2] \rightarrow b[a[P_1 | P_2] | Q_1\{x := a\} | Q_2]$$

This represent an access protocol where the credentials of incoming processes ($k$ in the rule above) are controlled, as a preliminary step to the registration protocol.
NBA: Syntax

Names: $a, b, \ldots n, x, y, \ldots \in \mathbb{N}$

Locations:

\[
\eta ::= a \quad \text{nested names} \quad \hat{\nu} \quad \text{enclosing ambient} \quad \star \quad \text{local}
\]

Messages:

\[
M, N ::= a \quad \text{name} \quad \text{enter}(M, N) \quad \text{may enter} \quad \text{exit}(M, N) \quad \text{may exit} \quad M \cdot N \quad \text{path}
\]

Processes:

\[
P ::= 0 \quad \text{nil process} \quad P_1 | P_2 \quad \text{composition} \quad (\nu n)P \quad \text{restriction} \quad !\pi.P \quad \text{replication} \quad M[P] \quad \text{ambient} \quad \pi.P \quad \text{prefixing}
\]

Prefixes:

\[
\pi ::= M \quad \text{messages} \quad (x_1, \ldots, x_k)^\eta \quad \text{input} \quad (M_1, \ldots, M_k)^\eta \quad \text{output} \quad \text{enter}(x, M) \quad \text{allow enter} \quad \text{exit}(x, M) \quad \text{allow exit}
\]
NBA: Reduction Semantics

mobility

\[ n[\text{enter}(m, k) . P_1 \mid P_2] \mid m[\text{enter}(x, k) . Q_1 \mid Q_2] \rightarrow m[n[P_1 \mid P_2] \mid Q_1 \{x := n\} \mid Q_2] \]

\[ n[m[\text{exit}(n, k) . P_1 \mid P_2] \mid Q] \mid \text{exit}(x, k) . R \rightarrow m[P_1 \mid P_2] \mid n[Q] \mid R\{x := m\} \]

communication

\[ (\tilde{x}) . P \mid \langle \tilde{M} \rangle . Q \rightarrow P\{\tilde{x} := \tilde{M}\} \mid Q \]

\[ (\tilde{x})^n . P \mid n[\langle \tilde{M} \rangle . Q \mid R] \rightarrow P\{\tilde{x} := \tilde{M}\} \mid n[Q \mid R] \]

\[ \langle \tilde{M} \rangle^n . P \mid n[(\tilde{x}) . Q \mid R] \rightarrow P \mid n[Q\{\tilde{x} := \tilde{M}\} \mid R] \]

structural congruence

\[ P \equiv Q \quad Q \rightarrow R \quad R \equiv S \text{ implies } P \rightarrow S \]
Barbs

\[
P \downarrow_n \text{ iff } P \equiv (\nu m)(n[\text{enter}(x, k).Q \mid R] \mid S), \quad \text{for } \{n, k\} \cap \{m\} = \emptyset.
\]

\[
P \Downarrow_n \text{ iff } P \Rightarrow P' \text{ and } P' \downarrow_n.
\]
NBA: Behavioural Equivalence

Barbs

\[ P \downarrow_n \text{ iff } P \equiv (\nu \tilde{m})(n[\text{enter}(x, k).Q | R] | S), \text{ for } \{n, k\} \cap \{\tilde{m}\} = \emptyset. \]

\[ P \downarrow_n \text{ iff } P \Rightarrow P' \text{ and } P' \downarrow_n. \]

A relation \( \mathcal{R} \) is reduction closed if

\[ P \mathcal{R} Q \text{ and } P \rightarrow P' \text{ implies } Q \Rightarrow Q' \text{ with } P' \mathcal{R} Q'; \]

it is barb preserving if \( P \mathcal{R} Q \) and \( P \downarrow_n \) implies \( Q \downarrow_n \).

Reduction barbed congruence, written \( \cong \), is the largest congruence relation over processes which is reduction closed and barb preserving.

Note: We could equivalently observe \( \langle \cdot \rangle^\ast \).
The rest of the talk

- Two small examples
- A few equational laws
- LTS characterization of reduction barbed bisimulation congruence.
- A type system
- An encoding of BA into NBA: $\simeq$ NBA + Guarded Choice
Let $w(k)$ be a bidirectional forwarder for any pair of incoming ambients.

\[ w(k) \triangleq w[ \overline{\text{enter}}(x, k).\overline{\text{enter}}(y, k).(!z^x.\langle z \rangle^y | !z^y.\langle z \rangle^x) ] \]

An agent can be defined as: $A(a, k, P, Q) \triangleq a[\text{enter}(w, k).P | \text{exit}(w, k).Q]$ and a communication server as:

\[ \sigma_2\sigma(k) = (\nu r) ( r[\langle \rangle^\hat{\langle} ] | !()^r. (w(k) | \overline{\text{exit}}(_, k).\overline{\text{exit}}(_, k).r[\langle \rangle^\hat{\langle} ] )) \]
A one-to-one communication server

Let \( w(k) \) be a bidirectional forwarder for any pair of incoming ambients.

\[
w(k) \triangleq w[ \overline{\text{enter}(x, k)} \cdot \overline{\text{enter}(y, k)} \cdot (z^x \cdot z^y \mid z^y \cdot z^x) ]
\]

An agent can be defined as: \( A(a, k, P, Q) \triangleq a[\text{enter}(w, k) \cdot P \mid \text{exit}(w, k) \cdot Q] \) and a communication server as:

\[
\omega_2 \omega(k) = (\nu r) ( r[\langle \rangle^\hat{\cdot} ] \mid (\nu r)^\cdot (w(k) \mid \overline{\text{exit}(\_, k)} \cdot \overline{\text{exit}(\_, k)} \cdot r[\langle \rangle^\hat{\cdot} ] ) )
\]

It can be proved that:

\[
(\nu k)(\omega_2 \omega(k) \mid A(k, a_1, \langle M \rangle^\hat{\cdot} \cdot P_1, Q_1) \mid A(k, a_2, (x)^\hat{\cdot} \cdot P_2 \{x\}, Q_2) \mid \Pi_{i \in I} A(K, a_i, R_i, S_i) )
\]

\[
\implies \cong (\nu k)(\omega_2 \omega(k) \mid a_1[P_1 \mid Q_1] \mid a_2[P_1 x := M \mid Q_2] \mid \Pi_{i \in I} A(K, a_i, R_i, S_i) )
\]

that is, once two agents engage in communication no other agent knowing the key \( k \) can interfere with their completing the exchange.
A print server

The following process assigns a progressive number to incoming jobs.

\[
\text{enqueue}_k \triangleq (\nu c) (\ c[\langle 1 \rangle^\wedge] \mid !(n)^c.\text{enter}(x, k).\langle n \rangle^x.c[\langle n + 1 \rangle^\wedge])
\]
A print server

The following process assigns a progressive number to incoming jobs.

\[ \text{enqueue}_k \triangleq (\nu c) ( c[ \langle 1 \rangle^\hat{\_} ] | !(n)^c.\text{enter}(x, k).\langle n \rangle^x.c[ \langle n + 1 \rangle^\hat{\_} ] ) \]

We can turn it into a print server (which consumes such numbers).

\[ \text{prtsrv}(k) \triangleq k[ \text{enqueue}_k | \text{print} ] \]

\[ \text{print} \triangleq (\nu c) ( c[ \langle 1 \rangle^\hat{\_} ] | !(n)^c.\text{exit}(x, n).\langle data \rangle^x.(P\{data\} | c[ \langle n + 1 \rangle^\hat{\_} ] ) ) \]

A client then acts as:

\[ \text{job}(M, k) \triangleq (\nu p)p[ \text{enter}(k, k).\langle n \rangle^\hat{\_}.(\nu q)q[\text{exit}(p, n).\langle M \rangle^\hat{\_} ] ] \]

It enters the server \( \text{prtsrv}(k) \) (using \text{enqueue}), it is assigned a number that it uses as a password to carry job \( M \) to \text{print} (which eventually will bind it to \text{data} in \( P \). (Dynamic name discovery and passwords are fundamental here.)
Some Equational Laws

Garbage Collection laws

\[ l[ (\tilde{x}_i)^n.P \mid (\tilde{x}).Q \mid \langle \tilde{M} \rangle^m.R ] \simeq 0 \]

\[ l[ (\tilde{x})^n.P \mid \langle \tilde{M} \rangle.P \mid \langle \tilde{M} \rangle^m.P ] \simeq 0 \]
Some Equational Laws

Garbage Collection laws

\[ l[ (\tilde{x}_i)^n.P \mid (\tilde{x}).Q \mid \langle \tilde{M} \rangle^m.R ] \cong 0 \]

\[ l[ (\tilde{x})^n.P \mid \langle \tilde{M} \rangle.P \mid \langle \tilde{M} \rangle^m.P ] \cong 0 \]

Communication laws

\[ l[ \langle \tilde{M}_0 \rangle^\hat{\cdot} \mid \langle \tilde{M}_1 \rangle^\hat{\cdot} ] \cong l[\langle \tilde{M}_0 \rangle^\hat{\cdot} ] \mid l[\langle \tilde{M}_1 \rangle^\hat{\cdot} ] \]

\[ l[(\tilde{x}).P \mid \langle \tilde{M} \rangle.Q] \cong l[P\{\tilde{x} := \tilde{M}\} \mid Q] \]

\[ (v_l)\left( (\tilde{x})^l.P \mid l[\langle \tilde{M} \rangle^\hat{\cdot}.Q] \right) \cong (v_l)\left( P\{\tilde{x} := \tilde{M}\} \mid l[Q] \right) \]

\[ m[(\tilde{x})^l.P \mid l[\langle \tilde{M} \rangle^\hat{\cdot}.Q]] \cong m[P\{\tilde{x} := \tilde{M}\} \mid l[Q]] \]
Some Equational Laws

Garbage Collection laws

- \( l[ (\tilde{x}_i)^n.P \mid (\tilde{x}).Q \mid \langle \tilde{M} \rangle^m.R ] \cong 0 \)
- \( l[ (\tilde{x})^n.P \mid \langle \tilde{M} \rangle.P \mid \langle \tilde{M} \rangle^m.P ] \cong 0 \)

Communication laws

- \( l[ \langle \tilde{M}_0 \rangle \hat{\cdot} \mid \langle \tilde{M}_1 \rangle \hat{\cdot} ] \cong l[\langle \tilde{M}_0 \rangle \hat{\cdot} ] \mid l[\langle \tilde{M}_1 \rangle \hat{\cdot} ] \)
- \( l[(\tilde{x}).P \mid \langle \tilde{M} \rangle.Q] \cong l[P\{\tilde{x} := \tilde{M}\} \mid Q] \)
- \( (\nu l)( (\tilde{x})^l.P \mid l[\langle \tilde{M} \rangle \hat{\cdot}.Q]) \cong (\nu l)( P\{\tilde{x} := \tilde{M}\} \mid l[Q] \)
- \( m[(\tilde{x})^l.P \mid l[\langle \tilde{M} \rangle \hat{\cdot}.Q]] \cong m[P\{\tilde{x} := \tilde{M}\} \mid l[Q] \]

Mobility laws

- \( (\nu p)( m[\text{enter}\langle n, p \rangle.P] \mid n[\text{enter}\langle x, p \rangle.Q]) \cong (\nu p)( n[Q\{x := m\} \mid m[P]]) \)
- \( l[m[\text{enter}\langle n, p \rangle.P] \mid n[\text{enter}\langle x, p \rangle.Q]] \cong l[n[Q\{x := m\} \mid m[P]]] \)
An LTS for NBA

Concretions: \((\nu \tilde{p})\langle P \rangle Q\) and \((\nu \tilde{p})\langle M \rangle Q\)

**Amb Co-enter**

\[
P \xrightarrow{\text{enter}(n,k)} P'\]

\[
m[P] \xrightarrow{\text{enter}(n,k)} (\nu)\langle P' \rangle 0\]

**Co-enter HO**

\[
P \xrightarrow{m \text{ enter}(n,k)} (\nu \tilde{p})\langle P_1 \rangle P_2 \quad \tilde{p} \cap \text{fn}(Q) = \emptyset\]

\[
P \xrightarrow{m \text{ enter}(n,k)Q} (\nu \tilde{p})(m[n[Q] \mid P_1 \mid P_2)\]

**Exit**

\[
P \xrightarrow{\text{exit}(n,k)} (\nu \tilde{p})\langle m[P_1] \rangle P_2\]

\[
n[P] \xrightarrow{\text{exit}(k)} (\nu \tilde{p})\langle m \rangle (m[P_1] \mid n[P_2])\]

**\(\tau\)-Exit**

\[
P \xrightarrow{\text{exit}(k)} (\nu \tilde{p})\langle m \rangle P' \quad Q \xrightarrow{\text{exit}(m,k)} Q'\]

\[
P \mid Q \xrightarrow{\tau} (\nu \tilde{p})(P' \mid Q')\]

**Exit HO**

\[
P \xrightarrow{\text{exit}(n,k)} (\nu \tilde{p})\langle m[P_1] \rangle P_2 \quad x \in \text{fn}(R) \quad \tilde{p} \cap \text{fn}(Q|R) = \emptyset\]

\[
P \xrightarrow{\text{exit}(n,k)QR} (\nu \tilde{p})(m[P_1] \mid n[P_2 \mid Q] \mid R\{x := m\})\]
A Characterisation of Reduction Bisimulation

**Thm.** If $P \xrightarrow{\tau} P'$ then $P \xrightarrow{\cdot} P'$. If $P \xrightarrow{\cdot} P'$ then $P \xrightarrow{\tau} \equiv P'$.

**Bisimilarity.** A symmetric relation $R$ is a bisimulation if

$$P R Q \quad \text{and} \quad P \xrightarrow{\alpha} P' \quad \text{implies} \quad \exists Q \xrightarrow{\hat{\alpha}} Q' \quad \text{with} \quad P' R Q'.$$

$$P \simeq Q \quad \text{if} \quad P R Q \quad \text{for some bisimulation} \ R.$$

The closure under substitutions of $\simeq$ is denoted by $\simeq_c$.

**Thm.** If $P \simeq_c Q$ then $P \simeq Q$ and viceversa.
A Type System for NBA

Types

Message Types

\[ W ::= N[E] \quad \text{ambient/password} \]
\[ \quad | \quad C[E] \quad \text{capability} \]

Exchange Types

\[ E, F ::= \text{shh} \quad \text{no exchange} \]
\[ \quad | \quad W_1 \ldots W_k \quad \text{tuples} (k \geq 0) \]

Process Types

\[ T ::= [E, F] \quad \text{composite exchange} \]

\(N[E]\) types both ambients and passwords; \(\text{shh}\) is the silent type; \(N[\text{shh}]\) is an ambient with no upward exchanges or a password that reveal the visitor’s name.
A Type System for NBA

Types

Message Types
\[ W ::= N[E] \text{ ambient/password} \]
\[ \mid C[E] \text{ capability} \]

Exchange Types
\[ E, F ::= \text{shh no exchange} \]
\[ \mid W_1 \ldots W_k \text{ tuples (} k \geq 0 \text{)} \]

Process Types
\[ T ::= [E, F] \text{ composite exchange} \]

\( N[E] \) types both ambients and passwords; \( \text{shh} \) is the silent type; \( N[\text{shh}] \) is an ambient with no upward exchanges or a password that reveal the visitor’s name.

Type Environments

\[ (\text{Env Empty}) \]
\[ \Gamma \vdash \diamond \quad a \notin \text{Dom} (\Gamma) \]

\[ \emptyset \vdash \diamond \]

\[ \Gamma, a : W \vdash \diamond \]
Typing Rules

Messages

**Projection**

\[
\Gamma, a : W, \Gamma' \vdash \diamond
\]

\[
\Gamma, a : W, \Gamma' \vdash a : W
\]

**Path**

\[
\Gamma \vdash M_1 : C[E_1] \quad \Gamma \vdash M_2 : C[E_2]
\]

\[
\Gamma \vdash M_1.M_2 : C[E_1 \sqcap E_2]
\]

**Enter**

\[
\Gamma \vdash M : N[E] \quad \Gamma \vdash N : N[F] \quad (F \leq G)
\]

\[
\Gamma \vdash \text{enter}(M, N) : C[G]
\]

**Exit**

\[
\Gamma \vdash M : N[E] \quad \Gamma \vdash N : N[F] \quad (F \leq G)
\]

\[
\Gamma \vdash \text{exit}(M, N) : C[G]
\]
Typing Rules

Messages

(Projection)
\[ \Gamma, a : W, \Gamma' \vdash \diamond \]
\[ \Gamma, a : W, \Gamma' \vdash a : W \]

(Path)
\[ \Gamma \vdash M_1 : C[E_1] \quad \Gamma \vdash M_2 : C[E_2] \]
\[ \Gamma \vdash M_1.M_2 : C[E_1 \sqcup E_2] \]

(Enter)
\[ \Gamma \vdash M : N[E] \quad \Gamma \vdash N : N[F] \quad (F \leq G) \]
\[ \Gamma \vdash \text{enter}(M, N) : C[G] \]

(Exit)
\[ \Gamma \vdash M : N[E] \quad \Gamma \vdash N : N[F] \quad (F \leq G) \]
\[ \Gamma \vdash \text{exit}(M, N) : C[G] \]

Processes

(PAR)
\[ \Gamma \vdash P : [E, F] \quad \Gamma \vdash Q : [E, F] \]
\[ \Gamma \vdash P \mid Q : [E, F] \]

(REPL)
\[ \Gamma \vdash P : [E, F] \]
\[ \Gamma \vdash !P : [E, F] \]

(Dead)
\[ \Gamma \vdash \diamond \]

(NEW)
\[ \Gamma, n : N[G] \vdash P : [E, F] \]
\[ \Gamma \vdash (\nu n : N[G])P : [E, F] \]
Typing Rules: II

Processes: mobility

(Amb)
\[ \Gamma \vdash M : N[E] \quad \Gamma \vdash P : [F, E] \]
\[ \Gamma \vdash M[P] : [G, H] \]

(Prefix)
\[ \Gamma \vdash M : C[F] \quad \Gamma \vdash P : [E, G] \quad (F \leq G) \]
\[ \Gamma \vdash M.P : [E, G] \]

(Co-enter)
\[ \Gamma \vdash M : N[\tilde{W}] \quad \Gamma, x : N[\tilde{W}] \vdash P : [E, F] \]
\[ \Gamma \vdash \text{enter}(x, M).P : [E, F] \]

(Co-exit)
\[ \Gamma \vdash M : N[\tilde{W}] \quad \Gamma, x : N[\tilde{W}] \vdash P : [E, F] \]
\[ \Gamma \vdash \text{exit}(x, M).P : [E, F] \]

(Co-enter-silent)
\[ \Gamma \vdash M : N[\text{shh}] \quad \Gamma \vdash P : [E, F] \quad (x \not\in \text{fv}(P)) \]
\[ \Gamma \vdash \text{enter}(x, M).P : [E, F] \]

(Co-exit-silent)
\[ \Gamma \vdash M : N[\text{shh}] \quad \Gamma \vdash P : [E, F] \quad (x \not\in \text{fv}(P)) \]
\[ \Gamma \vdash \text{exit}(x, M).P : [E, F] \]
Typing Rules: II

Processes: I/O

(INPUT)
\[ \Gamma, x: \tilde{W} \vdash P : [\tilde{W}, E] \]
\[ \Gamma \vdash (x: \tilde{W}).P : [\tilde{W}, E] \]

(INPUT ^)
\[ \Gamma, x: \tilde{W} \vdash P : [E, \tilde{W}] \]
\[ \Gamma \vdash (x: \tilde{W})^\cdot P : [E, \tilde{W}] \]

(INPUT M)
\[ \Gamma \vdash M : N[\tilde{W}] \quad \Gamma, x: \tilde{W} \vdash P : [G, H] \]
\[ \Gamma \vdash (x: \tilde{W})^M.P : [G, H] \]

(Output)
\[ \Gamma \vdash \tilde{M} : \tilde{W} \quad \Gamma \vdash P : [\tilde{W}, E] \]
\[ \Gamma \vdash \langle \tilde{M} \rangle.P : [\tilde{W}, E] \]

(Output ^)
\[ \Gamma \vdash \tilde{M} : \tilde{W} \quad \Gamma \vdash P : [E, \tilde{W}] \]
\[ \Gamma \vdash \langle \tilde{M} \rangle^\cdot.P : [E, \tilde{W}] \]

(Output N)
\[ \Gamma \vdash N : N[\tilde{W}] \quad \Gamma \vdash \tilde{M} : \tilde{W} \quad \Gamma \vdash P : [G, H] \]
\[ \Gamma \vdash \langle \tilde{M} \rangle^N.P : [G, H] \]

Subject Reduction. If \( \Gamma \vdash P : T \) and \( P \rightarrow Q \), then \( \Gamma \vdash Q : T \).
We can encode BA into NBA enriched with a focused form of nondeterminism.

\[
\begin{align*}
\{ P \}_n &= \text{cross} \mid \{ P \}_n \\
\langle m[P] \rangle_n &= m[\{ P \}_m] \\
\langle (x)^a P \rangle_n &= (x)^a \{ P \}_n \\
\langle (x)P \rangle_n &= (x) \{ P \}_n + (x)\hat{\langle P \rangle}_n + \overline{\text{exit}(y, pw)}(x)^y \{ P \}_n & y \notin \text{fn}(P) \\
\langle (x)\hat{\rangle P \rangle_n &= (\nu p)p[\text{exit}(n, pr). (x)\hat{\langle P \rangle}_n + \text{enter}(n, p). (x)^\hat{\langle P \rangle}_n] \mid \overline{\text{enter}(y, p)}(x)^y \{ P \}_n & p, y \notin \text{fn}(P) \\
\langle \langle M \rangle^a P \rangle_n &= \langle M \rangle^a \{ P \}_n \\
\langle \langle M \rangle P \rangle_n &= \langle M \rangle \{ P \}_n + \langle M \rangle\hat{\langle P \rangle}_n + \overline{\text{exit}(y, pr)}\langle M \rangle^y \{ P \}_n & y \notin \text{fn}(P) \\
\langle \langle M \rangle\hat{\rangle P \rangle_n &= (\nu p)p[\text{exit}(n, pw). \langle M \rangle\hat{\langle P \rangle}_n + \text{enter}(n, p). (\cdot)^\hat{\langle P \rangle}_n] \mid \overline{\text{enter}(y, p)}(\cdot)^y \{ P \}_n & p, y \notin \text{fn}(P)
\end{align*}
\]

where \text{cross} = \overline{\text{enter}(x, mv)} \mid \overline{\text{exit}(x, mv)}, \text{in } n = \text{enter}(n, mv), \text{and out } n = \text{exit}(n, mv).
Encoding: BA in NBA

We can encode BA into NBA enriched with a focused form of nondeterminism.

\[
\begin{align*}
\{ P \}^n &= \text{cross} \mid \{ P \}^n \\
\langle m[P] \rangle^m &= m[\{ P \}^m] \\
\langle (x)^a P \rangle^m &= (x)^a \langle P \rangle^m \\
\langle (x) P \rangle^m &= (x) \langle P \rangle^m + (x)\hat{\langle P \rangle}^m + \text{exit}(y, pw)(x)^y \langle P \rangle^m & y \notin \text{fn}(P) \\
\langle (x)^\uparrow P \rangle^m &= (\nu p)p[\text{exit}(n, pr).\langle x \rangle^\hat{\langle P \rangle} \cdot \text{enter}(n, p).\langle x \rangle^\hat{\langle P \rangle} | \text{enter}(y, p)(x)^y \langle P \rangle^m & p, y \notin \text{fn}(P) \\
\langle \langle M \rangle^a P \rangle^m &= \langle M \rangle^a \langle P \rangle^m \\
\langle \langle M \rangle P \rangle^m &= \langle M \rangle \langle P \rangle^m + \langle M \rangle\hat{\langle P \rangle}^m + \text{exit}(y, pr)\langle M \rangle^y \langle P \rangle^m & y \notin \text{fn}(P) \\
\langle \langle M \rangle^\uparrow P \rangle^m &= (\nu p)p[\text{exit}(n, pw).\langle M \rangle^\hat{\langle P \rangle} \cdot \text{enter}(n, p).\langle \cdot \rangle^\hat{\langle P \rangle} | \text{enter}(y, p)(\cdot)^y \langle P \rangle^m & p, y \notin \text{fn}(P)
\end{align*}
\]

where \text{cross} = \text{!enter}(x, mv) \mid \text{!exit}(x, mv), in \ n = \text{enter}(n, mv), and out \ n = \text{exit}(n, mv).

\[\textbf{Thm.}\] If \( P \xrightarrow{\tau} P' \) then \( \{ P \} \xrightarrow{\tau} \succeq \{ P' \} \).

If \( \{ P \} \xrightarrow{\tau} Q \), then \( \exists P \xrightarrow{\tau} P' \) with \( Q \succeq \{ P' \} \).

If \( P \) and \( Q \) are single-threaded, then \( \{ P \}^m \equiv \{ Q \}^m \) implies \( P \equiv Q \).
Conclusion and Future Work

- Type inference.
- Information flow analysis.
- Comparison with Seal calculus.
- Implementation.
- Logics.