Intersection Types Meet Session Types

Reconciling Two Different Views on Computational Resources

Jorge A. Pérez

University of Groningen, The Netherlands

UNIFYING CORRECTNESS FOR COMMUNICATING SOFTWARE

ITRS 2024 - Tallinn

Collaborators

Daniele Nantes
(Imperial College, UK and
University of Brasília, BR)

Joseph Paulus
(Oxford University, UK)

Bas van den Heuvel
(HKA Karlsruhe and
University of Freiburg, DE)

This Talk

A brief overview of formal connections between intersection types with session types for concurrency (FSCD'21, TYPES'21, APLAS'23, MFPS'24).

This Talk

A brief overview of formal connections between intersection types with session types for concurrency (FSCD'21, TYPES'21, APLAS'23, MFPS'24).

- Non-idempotent intersection types and session types with Curry-Howard foundations (aka 'propositions-as-sessions')

This Talk

A brief overview of formal connections between intersection types with session types for concurrency (FSCD'21, TYPES'21, APLAS'23, MFPS'24).

- Non-idempotent intersection types and session types with Curry-Howard foundations (aka 'propositions-as-sessions')
- Two different approaches on resource control:
  - In λ: Resources are terms, whose types can ensure quantitative properties
  - In π: Resources are channels, whose types enforce linear usage for correctness
- Key unifying aspects: non-determinism, confluence / commitment, failures

This Talk

A brief overview of formal connections between intersection types with session types for concurrency (FSCD'21, TYPES'21, APLAS'23, MFPS'24).

- Non-idempotent intersection types and session types with Curry-Howard foundations (aka 'propositions-as-sessions')
- Two different approaches on resource control:
  - In λ: Resources are terms, whose types can ensure quantitative properties
  - In π: Resources are channels, whose types enforce linear usage for correctness
- Key unifying aspects: non-determinism, confluence / commitment, failures
- Relative expressiveness: Typed λ-calculi translated into session-typed π-calculi

This Talk

A brief overview of formal connections between intersection types with session types for concurrency (FSCD'21, TYPES'21, APLAS'23, MFPS'24).

- Non-idempotent intersection types and session types with Curry-Howard foundations (aka 'propositions-as-sessions')
- Two different approaches on resource control:
  - In λ: Resources are terms, whose types can ensure quantitative properties
  - In π: Resources are channels, whose types enforce linear usage for correctness
- Key unifying aspects: non-determinism, confluence / commitment, failures
- Relative expressiveness: Typed λ-calculi translated into session-typed π-calculi
- A concurrent interpretation of intersection types, with strong correctness properties: static (type preservation) and dynamic (operational correspondence)

Our Translation, In a Nutshell (FSCD'21 / LMCS'23)

Source: $\lambda^{\lightning}_{\oplus}$, a resource λ-calculus with non-determinism and failure

- Applications $M\ B$, where B is a bag of terms, to be non-determically fetched
- Failure occurs when a bag's size doesn't match the variable occurrences
- Intersection types measure the size of a bag and count variable occurrences

Our Translation, In a Nutshell (FSCD'21 / LMCS'23)

Source: $\lambda_{\oplus}^{\lightning}$, a resource λ-calculus with non-determinism and failure

- Applications $M\ B$, where B is a bag of terms, to be non-determically fetched
- Failure occurs when a bag's size doesn't match the variable occurrences
- Intersection types measure the size of a bag and count variable occurrences
- Expressions can be well-typed (fail-free) or well-formed (fail-prone)

Our Translation, In a Nutshell (FSCD'21 / LMCS'23)

Source: $\lambda^{\lightning}_{\oplus}$, a resource λ-calculus with non-determinism and failure

- Applications $M\ B$, where B is a bag of terms, to be non-determically fetched
- Failure occurs when a bag's size doesn't match the variable occurrences
- Intersection types measure the size of a bag and count variable occurrences
- Expressions can be well-typed (fail-free) or well-formed (fail-prone)

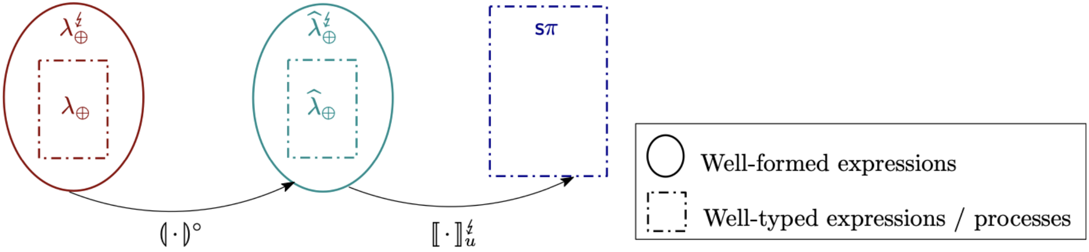
Target: sπ, a session-typed π-calculus with non-deterministically available behaviors

- **Session types**: The protocols that names of a process must respect
- Under 'propositions-as-sessions', session type = linear logic proposition
- Non-deterministic processes, whose protocols may proceed as intended or fail
- Well-typed processes respect their types, with confluence and deadlock-freedom

Our Translation, In a Nutshell (FSCD'21 / LMCS'23)

Structly speaking, two translations:

1. We first translate $\lambda^{\lightning}_{\oplus}$ into a resource λ-calculus with sharing constructs
   A direct translation, useful to make different variable occurrences explicit
2. The translation into processes leverages sharing to enforce linearity; failure for terms accounted as non-available protocols



A resource λ-calculus with failure ($\lambda^{\lightning}_{\oplus}$)

Syntax:

$$
\begin{array}{llcl}
\text{(Terms)} & M, N, L & ::= & x \mid \lambda x.M \mid (M\ B) \mid M\langle\!\langle B/x \rangle\!\rangle \mid \mathtt{fail}^{\widetilde{x}} \\
\text{(Bags)} & A, B & ::= & 1 \mid \wr\, M \,\int \mid A \cdot B \\
\text{(Expressions)} & \mathbb{M}, \mathbb{N}, \mathbb{L} & ::= & M \mid \mathbb{M} + \mathbb{N}
\end{array}
$$

A resource λ-calculus with failure ($\lambda_{\oplus}^{\lightning}$)

Syntax:

$$
\begin{array}{llll}
\text{(Terms)} & M, N, L & ::= & x \mid \lambda x.M \mid (M\ B) \mid M\langle\!\langle B/x \rangle\!\rangle \mid \mathtt{fail}^{\widetilde{x}} \\
\text{(Bags)} & A, B & ::= & 1 \mid \wr M \wr \mid A \cdot B \\
\text{(Expressions)} & \mathbb{M}, \mathbb{N}, \mathbb{L} & ::= & M \mid \mathbb{M} + \mathbb{N}
\end{array}
$$

Reduction (Excerpt):

$$
[\mathtt{R : Beta}]\ \frac{}{(\lambda x.M)B \longrightarrow M\ \langle\!\langle B/x \rangle\!\rangle}
$$

$$
[\mathtt{R : Fetch}]\ \frac{\mathsf{head}(M) = x \qquad B = \wr N_1, \ldots, N_k \wr\ ,\ k \geq 1 \qquad \#(x, M) = k}{M\ \langle\!\langle B/x \rangle\!\rangle \longrightarrow M\{\!| N_1/x |\!\}\langle\!\langle (B \setminus\!\setminus N_1)/x \rangle\!\rangle + \cdots + M\{\!| N_k/x |\!\}\langle\!\langle (B \setminus\!\setminus N_k)/x \rangle\!\rangle}
$$

$$
[\mathtt{R : Fail}]\ \frac{\#(x, M) \neq \mathsf{size}(B) \qquad \widetilde{y} = (\mathsf{mfv}(M) \setminus x) \uplus \mathsf{mfv}(B)}{M\ \langle\!\langle B/x \rangle\!\rangle \longrightarrow \sum_{\mathsf{PER}(B)} \mathtt{fail}^{\widetilde{y}}}
$$

Intersection Types

We define strict and multiset types by the grammar:

$$\begin{array}{lrcl} \text{(Strict)} & \sigma, \tau, \delta & ::= & \mathbf{unit} \mid \pi \to \sigma \\ \text{(Multiset)} & \pi & ::= & \sigma^k \mid \omega \end{array}$$

where σ^k stands for the intersection $\sigma \wedge \cdots \wedge \sigma$ (k times, for some $k > 0$)

Given **type contexts** $\Gamma, \Delta, \ldots$ (sets of type assignments $x : \pi$), **type judgements** are of the form

$$\Gamma \vdash \mathbb{M} : \sigma$$

Well-Typed $\neq$ Well-Formed

- **Well-typed** terms use resources properly

$$[\mathtt{T} : \mathtt{app}] \; \frac{\Gamma \vdash M : \pi \to \tau \qquad \Delta \vdash B : \pi}{\Gamma \wedge \Delta \vdash M\ B : \tau} \qquad [\mathtt{T} : \mathtt{ex\text{-}sub}] \; \frac{\Gamma, x : \sigma^k \vdash M : \tau \qquad \Delta \vdash B : \sigma^k}{\Gamma \wedge \Delta \vdash M \langle\!\langle B/x \rangle\!\rangle : \tau}$$

Well-Typed $\neq$ Well-Formed

- **Well-typed** terms use resources properly

$$[\mathsf{T:app}]\ \frac{\Gamma \vdash M : \pi \to \tau \qquad \Delta \vdash B : \pi}{\Gamma \wedge \Delta \vdash M\ B : \tau} \qquad [\mathsf{T:ex\text{-}sub}]\ \frac{\Gamma, x : \sigma^k \vdash M : \tau \qquad \Delta \vdash B : \sigma^k}{\Gamma \wedge \Delta \vdash M\langle\!\langle B/x \rangle\!\rangle : \tau}$$

- **Well-formed** terms allow resource mismatches, which lead to failure

$$[\mathsf{F:ex\text{-}sub}]\ \frac{\Gamma, x : \sigma^k \models M : \tau \qquad \Delta \models B : \sigma^j \quad k, j \geq 0}{\Gamma \wedge \Delta \models M\langle\!\langle B/x \rangle\!\rangle : \tau}$$

$$[\mathsf{F:app}]\ \frac{\Gamma \models M : \sigma^j \to \tau \qquad \Delta \models B : \sigma^k \qquad k, j \geq 0}{\Gamma \wedge \Delta \models M\ B : \tau}$$

Well-typed processes are also well-formed (but not the other way around)

Sharing is Caring

- In the sharing calculus, denoted $\widehat{\lambda}^{\lightning}_{\oplus}$, a variable can only appear once in a term
- Multiple occurrences of the same variable are “atomized”
- Sharing of variables $\widetilde{x}$ occurring in M using x:

$$M[\widetilde{x} \leftarrow x]$$

Sharing is Caring

- In the sharing calculus, denoted $\widehat{\lambda}^{\lightning}_{\oplus}$, a variable can only appear once in a term
- Multiple occurrences of the same variable are "atomized"
- Sharing of variables $\widetilde{x}$ occurring in M using x:

$$M[\widetilde{x} \leftarrow x]$$

- Examples:

$$\lambda x.x_1[x_1 \leftarrow x] \qquad \lambda x.x_1 \wr x_2 \int [x_1, x_2 \leftarrow x]$$

- The shared variables $\widetilde{x}$ can be empty:
  $M[\leftarrow x]$ says that x does not share any variables in M.

A π-calculus with non-deterministic sessions (sπ)

Fragment from Caires and Pérez (ESOP'21):

$$
\begin{array}{rlll}
P, Q ::= & \overline{x}[y].(P \mid Q) & \mid & x(y).P \\
\mid & x.\overline{\texttt{some}}; P & \mid & x.\texttt{some}_{w_1,\ldots,w_n}; P \\
\mid & x.\overline{\texttt{none}} & & \\
\mid & \overline{x}[\,] & \mid & x(\,).P \\
\mid & (\boldsymbol{\nu} x)(P \mid Q) & & \\
\mid & P \oplus Q & \mid & P \mid Q \\
\mid & 0 & \mid & [x \leftrightarrow y]
\end{array}
$$

Confluent Non-determinism and Failures in sπ

$$y_1(z).y_2(w).0$$

Confluent Non-determinism and Failures in sπ

$$x.\mathtt{some}_{(y_1, y_2)}; y_1(z).y_2(w).0$$

Confluent Non-determinism and Failures in sπ

$$R = \; (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid (x.\overline{\mathtt{some}}; P \oplus x.\overline{\mathtt{none}}))$$

Confluent Non-determinism and Failures in sπ

$$
\begin{aligned}
R = \;& (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid (x.\overline{\mathtt{some}}; P \oplus x.\overline{\mathtt{none}})) \\
\equiv \;& (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid x.\overline{\mathtt{some}}; P) \\
& \qquad\qquad\qquad\oplus \\
& (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid x.\overline{\mathtt{none}})
\end{aligned}
$$

Confluent Non-determinism and Failures in sπ

$$
\begin{aligned}
R = {} & (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid (x.\overline{\mathtt{some}}; P \oplus x.\overline{\mathtt{none}})) \\
\equiv {} & (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid x.\overline{\mathtt{some}}; P) \\
& \qquad\qquad\qquad\qquad \oplus \\
& (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; y_1(z).y_2(w).0 \mid x.\overline{\mathtt{none}})
\end{aligned}
$$

Letting $Q = y_1(z).y_2(w).0$, we have:

$$
\begin{tikzcd}
 & (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; Q \mid x.\overline{\mathtt{some}}; P) \oplus (y_1.\overline{\mathtt{none}} \mid y_2.\overline{\mathtt{none}}) \arrow[dr] & \\
R = (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; Q \mid (x.\overline{\mathtt{some}}; P \oplus x.\overline{\mathtt{none}})) \arrow[ur] \arrow[dr] & & (\nu x)(Q \mid P) \oplus (y_1.\overline{\mathtt{none}} \mid y_2.\overline{\mathtt{none}}) \\
 & (\nu x)(Q \mid P) \oplus (\nu x)(x.\mathtt{some}_{(y_1,y_2)}; Q \mid x.\overline{\mathtt{none}}) \arrow[ur] &
\end{tikzcd}
$$

Session Types for sπ

Linear logic propositions as session types (cf. Caires & Pfenning, Wadler):

$$
\begin{array}{rll}
A, B ::= & \bot & \text{(closed session)} \\
 & \mid 1 & \text{(closed session)} \\
 & \mid A \otimes B & \text{(output } A\text{, continue as } B\text{)} \\
 & \mid A \parr B & \text{(input } A\text{, continue as } B\text{)} \\
 & \mid \& A & \text{(may produce } A\text{)} \\
 & \mid \oplus A & \text{(may consume } A\text{)}
\end{array}
$$

A duality relation on types/propositions connects complementary behaviors

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to \mathsf{s}\pi$

$$
\begin{aligned}
[\![x]\!]_u &= x.\overline{\mathtt{some}}; [x \leftrightarrow u] \\
[\![\lambda x.M[\tilde{x} \leftarrow x]]\!]_u &= u.\overline{\mathtt{some}}; u(x).[\![M[\tilde{x} \leftarrow x]]\!]_u \\
[\![M(\wr N_1 \int \cdot \wr N_2 \int)]\!]_u &= (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_1 \int \cdot \wr N_2 \int)]\!]_x)) \\
&\quad \oplus \\
&\quad (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_2 \int \cdot \wr N_1 \int)]\!]_x)) \\
[\![M[\tilde{x} \leftarrow x]\langle\!\langle B/x \rangle\!\rangle]\!]_u &= \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)([\![M[\tilde{x} \leftarrow x]]\!]_u \mid [\![B_i]\!]_x)
\end{aligned}
$$

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to s\pi$

$$
\begin{aligned}
[\![x]\!]_u &= x.\overline{\mathtt{some}}; [x \leftrightarrow u] \\
[\![\lambda x.M[\tilde{x} \leftarrow x]]\!]_u &= u.\overline{\mathtt{some}}; u(x).[\![M[\tilde{x} \leftarrow x]]\!]_u \\
[\![M(\wr N_1 \int \cdot \wr N_2 \int)]\!]_u &= (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_1 \int \cdot \wr N_2 \int)]\!]_x)) \\
&\quad \oplus \\
&\quad (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_2 \int \cdot \wr N_1 \int)]\!]_x)) \\
[\![M[\tilde{x} \leftarrow x]\langle\!\langle B/x \rangle\!\rangle]\!]_u &= \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)([\![M[\tilde{x} \leftarrow x]]\!]_u \mid [\![B_i]\!]_x)
\end{aligned}
$$

The Translation $\llbracket \cdot \rrbracket_u : \widehat{\lambda}^{\lightning}_{\oplus} \to s\pi$

$$\llbracket x \rrbracket_u = x.\overline{\mathtt{some}}; [x \leftrightarrow u]$$

$$\llbracket \lambda x.M[\widetilde{x} \leftarrow x] \rrbracket_u = u.\overline{\mathtt{some}}; u(x).\llbracket M[\widetilde{x} \leftarrow x] \rrbracket_u$$

$$\begin{aligned}\llbracket M(\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_u = {} & (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\ & \qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_x)) \\ & \oplus \\ & (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\ & \qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_2 \int \cdot \wr N_1 \int) \rrbracket_x))\end{aligned}$$

$$\llbracket M[\widetilde{x} \leftarrow x]\langle\!\langle B/x \rangle\!\rangle \rrbracket_u = \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)(\llbracket M[\widetilde{x} \leftarrow x] \rrbracket_u \mid \llbracket B_i \rrbracket_x)$$

The Translation $\llbracket \cdot \rrbracket_u : \widehat{\lambda}^{\lightning}_{\oplus} \to \mathsf{s}\pi$

$$
\begin{aligned}
\llbracket x \rrbracket_u &= x.\overline{\mathtt{some}}; [x \leftrightarrow u] \\
\llbracket \lambda x.M[\tilde{x} \leftarrow x] \rrbracket_u &= u.\overline{\mathtt{some}}; u(x).\llbracket M[\tilde{x} \leftarrow x] \rrbracket_u \\
\llbracket M (\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_u &= (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_x)) \\
&\quad \oplus \\
&\quad (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad\qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_2 \int \cdot \wr N_1 \int) \rrbracket_x)) \\
\llbracket M[\tilde{x} \leftarrow x] \langle\!\langle B/x \rangle\!\rangle \rrbracket_u &= \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)(\llbracket M[\tilde{x} \leftarrow x] \rrbracket_u \mid \llbracket B_i \rrbracket_x)
\end{aligned}
$$

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to s\pi$

$$[\![x]\!]_u = x.\overline{\mathtt{some}}; [x \leftrightarrow u]$$

$$[\![\lambda x.M[\widetilde{x} \leftarrow x]]\!]_u = u.\overline{\mathtt{some}}; u(x).[\![M[\widetilde{x} \leftarrow x]]\!]_u$$

$$\begin{aligned}[\![M(\wr N_1 \int \cdot \wr N_2 \int)]\!]_u = {} & (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\ & \qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_1 \int \cdot \wr N_2 \int)]\!]_x)) \\ & \oplus \\ & (\nu v)([\![M]\!]_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\ & \qquad \overline{v}[x].([v \leftrightarrow u] \mid [\![(\wr N_2 \int \cdot \wr N_1 \int)]\!]_x))\end{aligned}$$

$$[\![M[\widetilde{x} \leftarrow x]\langle\!\langle B/x \rangle\!\rangle]\!]_u = \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)([\![M[\widetilde{x} \leftarrow x]]\!]_u \mid [\![B_i]\!]_x)$$

The Translation $\llbracket \cdot \rrbracket_u : \widehat{\lambda}^{\lightning}_{\oplus} \to s\pi$

$$
\begin{aligned}
\llbracket x \rrbracket_u &= x.\overline{\mathtt{some}}; [x \leftrightarrow u] \\
\llbracket \lambda x.M[\tilde{x} \leftarrow x] \rrbracket_u &= u.\overline{\mathtt{some}}; u(x).\llbracket M[\tilde{x} \leftarrow x] \rrbracket_u \\
\llbracket M (\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_u &= (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_1 \int \cdot \wr N_2 \int) \rrbracket_x)) \\
&\quad \oplus \\
&\quad (\nu v)(\llbracket M \rrbracket_v \mid v.\mathtt{some}_{u,\mathsf{fv}(B)}; \\
&\qquad \overline{v}[x].([v \leftrightarrow u] \mid \llbracket (\wr N_2 \int \cdot \wr N_1 \int) \rrbracket_x)) \\
\llbracket M[\tilde{x} \leftarrow x]\langle\!\langle {}^{B}/x \rangle\!\rangle \rrbracket_u &= \bigoplus_{B_i \in \mathsf{PER}(B)} (\nu x)(\llbracket M[\tilde{x} \leftarrow x] \rrbracket_u \mid \llbracket B_i \rrbracket_x)
\end{aligned}
$$

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to s\pi$

$$
\begin{aligned}
[\![M[x_1, x_2 \leftarrow x]]\!]_u = x.\overline{\mathtt{some}}.\overline{x}[y_1].(y_1.\mathtt{some}_{\emptyset}; y_1(\,); 0 \\
\mid x.\overline{\mathtt{some}}; x.\mathtt{some}_{u,(\mathsf{fv}(M)\setminus x_1,\cdots,x_n)}; x(x_1). \\
x.\overline{\mathtt{some}}.\overline{x}[y_2].(y_2.\mathtt{some}_{\emptyset}; y_2(\,); 0 \\
\mid x.\overline{\mathtt{some}}; x.\mathtt{some}_{u,(\mathsf{fv}(M)\setminus x_2)}; x(x_2) \\
.x.\overline{\mathtt{some}}; \overline{x}[y_3].(y_3.\mathtt{some}_{u,\mathsf{fv}(M)}; y_3(\,); [\![M]\!]_u \mid x.\overline{\mathtt{none}})\;)) \\
[\![M[\leftarrow x]]\!]_u = x.\overline{\mathtt{some}}.\overline{x}[y_i].(y_i.\mathtt{some}_{u,\mathsf{fv}(M)}; y_i(\,); [\![M]\!]_u \mid x.\overline{\mathtt{none}})
\end{aligned}
$$

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to \mathsf{s}\pi$

$$
\begin{aligned}
[\![M[x_1, x_2 \leftarrow x]]\!]_u = {} & x.\overline{\mathtt{some}}.\overline{x}[y_1].(y_1.\mathtt{some}_{\emptyset}; y_1(\,); 0 \\
& \quad \mid x.\overline{\mathtt{some}}; x.\mathtt{some}_{u,(\mathsf{fv}(M)\setminus x_1,\cdots,x_n)}; x(x_1). \\
& \quad\quad x.\overline{\mathtt{some}}.\overline{x}[y_2].(y_2.\mathtt{some}_{\emptyset}; y_2(\,); 0 \\
& \quad\quad\quad \mid x.\overline{\mathtt{some}}; x.\mathtt{some}_{u,(\mathsf{fv}(M)\setminus x_2)}; x(x_2) \\
& \quad\quad\quad\quad .x.\overline{\mathtt{some}}; \overline{x}[y_3].(y_3.\mathtt{some}_{u,\mathsf{fv}(M)}; y_3(\,); [\![M]\!]_u \mid x.\overline{\mathtt{none}})\ )) \\
[\![M[\leftarrow x]]\!]_u = {} & x.\overline{\mathtt{some}}.\overline{x}[y_i].(y_i.\mathtt{some}_{u,\mathsf{fv}(M)}; y_i(\,); [\![M]\!]_u \mid x.\overline{\mathtt{none}})
\end{aligned}
$$

The Translation $[\![\cdot]\!]_u : \widehat{\lambda}^{\lightning}_{\oplus} \to \mathsf{s}\pi$

$$
\begin{aligned}
[\![\wr M \int \cdot B]\!]_x = {} & x.\mathtt{some}_{\mathsf{fv}(\wr M \int \cdot B)}; x(y_i).x.\mathtt{some}_{y_i,\mathsf{fv}(\wr M \int \cdot B)}; x.\overline{\mathtt{some}}; \overline{x}[x_i] \\
& .(x_i.\mathtt{some}_{\mathsf{fv}(M)}; [\![M]\!]_{x_i} \mid [\![B]\!]_x \mid y_i.\overline{\mathtt{none}}) \\
[\![1]\!]_x = {} & x.\mathtt{some}_{\emptyset}; x(y_n).(y_n.\overline{\mathtt{some}}; \overline{y_n}[\,] \mid x.\mathtt{some}_{\emptyset}; x.\overline{\mathtt{none}}) \\
[\![\mathbb{M} + \mathbb{N}]\!]_u = {} & [\![\mathbb{M}]\!]_u \oplus [\![\mathbb{N}]\!]_u \\
[\![\mathtt{fail}^{x_1,x_2}]\!]_u = {} & u.\overline{\mathtt{none}} \mid x_1.\overline{\mathtt{none}} \mid x_2.\overline{\mathtt{none}}
\end{aligned}
$$

The Translation at Work

$$
\begin{tikzcd}
(\lambda a.a) \wr M \int \arrow[d] \\
a \langle\!\langle {}^{\wr M \int}/a \rangle\!\rangle \arrow[d] \\
M \langle\!\langle {}^{1}/a \rangle\!\rangle \arrow[d, "\leq"] \\
M
\end{tikzcd}
$$

The Translation at Work

$$
\begin{tikzcd}
(\lambda a. a_1[a_1 \leftarrow a]) \lbag M \rbag \arrow[d] \\
a_1[a_1 \leftarrow a]\langle\!\langle {}^{\lbag M \rbag}/_{a} \rangle\!\rangle \arrow[d] \\
M[\leftarrow a]\langle\!\langle {}^{1}/_{a} \rangle\!\rangle \arrow[d, "\leq"] \\
M
\end{tikzcd}
$$

The Translation at Work

$$
\begin{tikzcd}
(\lambda a.a_1[a_1 \leftarrow a]) \wr M \int \arrow[r, "{\llbracket \cdot \rrbracket_u}"] \arrow[d] & \begin{array}{l}(\nu v)(v.\overline{\mathtt{some}}; v(x).\llbracket a_1[a_1 \leftarrow a] \rrbracket_v \\ \qquad \mid v.\mathtt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid \llbracket \wr M \int \rrbracket_x)\end{array} \\
a_1[a_1 \leftarrow a]\langle\!\langle {}^{\wr M \int}/a \rangle\!\rangle \arrow[d] & \\
M[\leftarrow a]\langle\!\langle {}^{1}/a \rangle\!\rangle \arrow[d, "\leq"] & \\
M &
\end{tikzcd}
$$

The Translation at Work

$$
\begin{tikzcd}[column sep=large]
(\lambda a.a_1[a_1 \leftarrow a]) \wr M \int \arrow[r, "\llbracket \cdot \rrbracket_u"] \arrow[d] & \begin{array}{l}(\nu v)(v.\overline{\texttt{some}}; v(x).\llbracket a_1[a_1 \leftarrow a] \rrbracket_v \\ \quad \mid v.\texttt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid \llbracket \wr M \int \rrbracket_x)\end{array} \arrow[d] \\
a_1[a_1 \leftarrow a]\langle\langle {}^{\wr M \int}/a \rangle\rangle \arrow[r, "\llbracket \cdot \rrbracket_u"] \arrow[d] & (\nu a)(\llbracket a_1[a_1 \leftarrow a] \rrbracket_u \mid \llbracket \wr M \int \rrbracket_a) \\
M[\leftarrow a]\langle\langle {}^{1}/a \rangle\rangle \arrow[d, "\leq"] & \\
M &
\end{tikzcd}
$$

The Translation at Work

$$
\begin{tikzcd}
(\lambda a.a_1[a_1 \leftarrow a]) \wr M \wr \arrow[r, "{[\![\cdot]\!]_u}"] \arrow[d] & \begin{array}{l}(\nu v)(v.\overline{\mathtt{some}}; v(x).[\![a_1[a_1 \leftarrow a]]\!]_v \\ \quad \mid v.\mathtt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid [\![\wr M \wr]\!]_x)\end{array} \arrow[d] \\
a_1[a_1 \leftarrow a]\langle\!\langle \wr M \wr / a \rangle\!\rangle \arrow[r, "{[\![\cdot]\!]_u}"] \arrow[d] & (\nu a)([\![a_1[a_1 \leftarrow a]]\!]_u \mid [\![\wr M \wr]\!]_a) \arrow[dd] \\
M[\leftarrow a]\langle\!\langle 1/a \rangle\!\rangle \arrow[d, "\leq"] & \\
M \arrow[r, "{[\![\cdot]\!]_u}"] & [\![M]\!]_u
\end{tikzcd}
$$

The Translation at Work: Failure

$$
\begin{array}{c}
(\lambda a.a)\mathtt{1} \\
\big\downarrow \\
a\langle\!\langle \mathtt{1}/a \rangle\!\rangle \\
\big\downarrow \\
\mathtt{fail}^{\emptyset}
\end{array}
$$

The Translation at Work: Failure

$$(\lambda a.a_1[a_1 \leftarrow a])\mathtt{1}$$

$$\downarrow$$

$$a_1[a_1 \leftarrow a]\langle\langle \mathtt{1}/a \rangle\rangle$$

$$\downarrow$$

$$\mathtt{fail}^{\emptyset}$$

The Translation at Work: Failure

$$
\begin{tikzcd}
(\lambda a. a_1[a_1 \leftarrow a])1 \arrow[r, "\llbracket \cdot \rrbracket_u"] \arrow[d] & (\nu v)(v.\overline{\texttt{some}}; v(x).\llbracket a_1[a_1 \leftarrow a] \rrbracket_v \mid v.\texttt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid \llbracket 1 \rrbracket_x)) \\
a_1[a_1 \leftarrow a]\langle\!\langle 1/a \rangle\!\rangle \arrow[d] & \\
\texttt{fail}^{\emptyset} &
\end{tikzcd}
$$

The Translation at Work: Failure

$$
\begin{tikzcd}
(\lambda a.a_1[a_1 \leftarrow a])1 \arrow[r, "\llbracket \cdot \rrbracket_u"] \arrow[d] & (\nu v)(v.\overline{\mathtt{some}}; v(x).\llbracket a_1[a_1 \leftarrow a] \rrbracket_v \mid v.\mathtt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid \llbracket 1 \rrbracket_x)) \arrow[d] \\
a_1[a_1 \leftarrow a]\langle\!\langle 1/a \rangle\!\rangle \arrow[r, "\llbracket \cdot \rrbracket_u"] \arrow[d] & (\nu a)(\llbracket a_1[a_1 \leftarrow a] \rrbracket_u \mid \llbracket 1 \rrbracket_a) \\
\mathtt{fail}^{\emptyset} &
\end{tikzcd}
$$

The Translation at Work: Failure

$$
\begin{tikzcd}
(\lambda a.a_1[a_1 \leftarrow a])1 \arrow[r, "{[\![\cdot]\!]_u}"] \arrow[d] & (\nu v)(v.\overline{\mathtt{some}}; v(x).[\![a_1[a_1 \leftarrow a]]\!]_v \mid v.\mathtt{some}_{u,b}; \overline{v}[x].([v \leftrightarrow u] \mid [\![1]\!]_x)) \arrow[d] \\
a_1[a_1 \leftarrow a]\langle\!\langle 1/a \rangle\!\rangle \arrow[r, "{[\![\cdot]\!]_u}"] \arrow[d] & (\nu a)([\![a_1[a_1 \leftarrow a]]\!]_u \mid [\![1]\!]_a) \arrow[d] \\
\mathtt{fail}^{\emptyset} \arrow[r, "{[\![\cdot]\!]_u}"] & u.\overline{\mathtt{none}}
\end{tikzcd}
$$

FSCD'21-LMCS'23

- $\lambda^{\lightning}_{\oplus}$ and $\widehat{\lambda}^{\lightning}_{\oplus}$: subject reduction / subject expansion (for well-typed terms)
- Translation of $\lambda^{\lightning}_{\oplus}$ into $\widehat{\lambda}^{\lightning}_{\oplus}$ (variable atomization)
- Translation of intersection types (for $\widehat{\lambda}^{\lightning}_{\oplus}$) into session types (for $s\pi$)

FSCD'21-LMCS'23

- $\lambda_{\oplus}^{\lightning}$ and $\widehat{\lambda}_{\oplus}^{\lightning}$: subject reduction / subject expansion (for well-typed terms)
- Translation of $\lambda_{\oplus}^{\lightning}$ into $\widehat{\lambda}_{\oplus}^{\lightning}$ (variable atomization)
- Translation of intersection types (for $\widehat{\lambda}_{\oplus}^{\lightning}$) into session types (for sπ)

TYPES'21

- Extension of $\lambda_{\oplus}^{\lightning}$ and $\widehat{\lambda}_{\oplus}^{\lightning}$ with unrestricted resources
- Relies on client and servers in sπ (typable via $?A$ and $!A$, copying semantics)

FSCD'21-LMCS'23

- $\lambda^{\lightning}_{\oplus}$ and $\widehat{\lambda}^{\lightning}_{\oplus}$: subject reduction / subject expansion (for well-typed terms)
- Translation of $\lambda^{\lightning}_{\oplus}$ into $\widehat{\lambda}^{\lightning}_{\oplus}$ (variable atomization)
- Translation of intersection types (for $\widehat{\lambda}^{\lightning}_{\oplus}$) into session types (for sπ)

TYPES'21

- Extension of $\lambda^{\lightning}_{\oplus}$ and $\widehat{\lambda}^{\lightning}_{\oplus}$ with unrestricted resources
- Relies on client and servers in sπ (typable via $?A$ and $!A$, copying semantics)

APLAS'23

- Variants of $\lambda^{\lightning}_{\oplus}$ and sπ with non-confluent non-determinism (linear resources)
- A lazy semantics for commitment; translation of terms/types unchanged

FSCD'21-LMCS'23

- $\lambda^{\lightning}_{\oplus}$ and $\widehat{\lambda}^{\lightning}_{\oplus}$: subject reduction / subject expansion (for well-typed terms)
- Translation of $\lambda^{\lightning}_{\oplus}$ into $\widehat{\lambda}^{\lightning}_{\oplus}$ (variable atomization)
- Translation of intersection types (for $\widehat{\lambda}^{\lightning}_{\oplus}$) into session types (for sπ)

TYPES'21

- Extension of $\lambda^{\lightning}_{\oplus}$ and $\widehat{\lambda}^{\lightning}_{\oplus}$ with unrestricted resources
- Relies on client and servers in sπ (typable via $?A$ and $!A$, copying semantics)

APLAS'23

- Variants of $\lambda^{\lightning}_{\oplus}$ and sπ with non-confluent non-determinism (linear resources)
- A lazy semantics for commitment; translation of terms/types unchanged

MFPS'24

- $\lambda^{\lightning}_{\oplus}$ and sπ with non-confluent non-determinism (with unrestricted resources)
- An eager semantics for commitment; lazy and eager translations compared

Concluding Remarks

- A glimpse at a series of concurrent interpretations (governed by session types) of resource λ-calculi with non-idempotent intersection types
- Reconciling non-determinism, failures, and confluence across functional and concurrent paradigms
- Focus on a basic setting with linear resources and confluent non-determinism. Extensions:
  - Unrestricted resources
  - Non-confluent non-determinism
  - Lazy/eager semantics for typed processes

Concluding Remarks

- A glimpse at a series of concurrent interpretations (governed by session types) of resource λ-calculi with non-idempotent intersection types
- Reconciling non-determinism, failures, and confluence across functional and concurrent paradigms
- Focus on a basic setting with linear resources and confluent non-determinism. Extensions:
  - Unrestricted resources
  - Non-confluent non-determinism
  - Lazy/eager semantics for typed processes

Current/future work:

- Recursive types
- Quantitative properties for typed process behaviors

Intersection Types Meet Session Types

Reconciling Two Different Views on Computational Resources

Jorge A. Pérez

University of Groningen, The Netherlands

UNIFYING C•RRECTNESS FOR C•MMUNICATING S•FTWARE

ITRS 2024 - Tallinn