On the Expressiveness of Polyadic and Synchronous Communication in Higher-Order Process Calculi

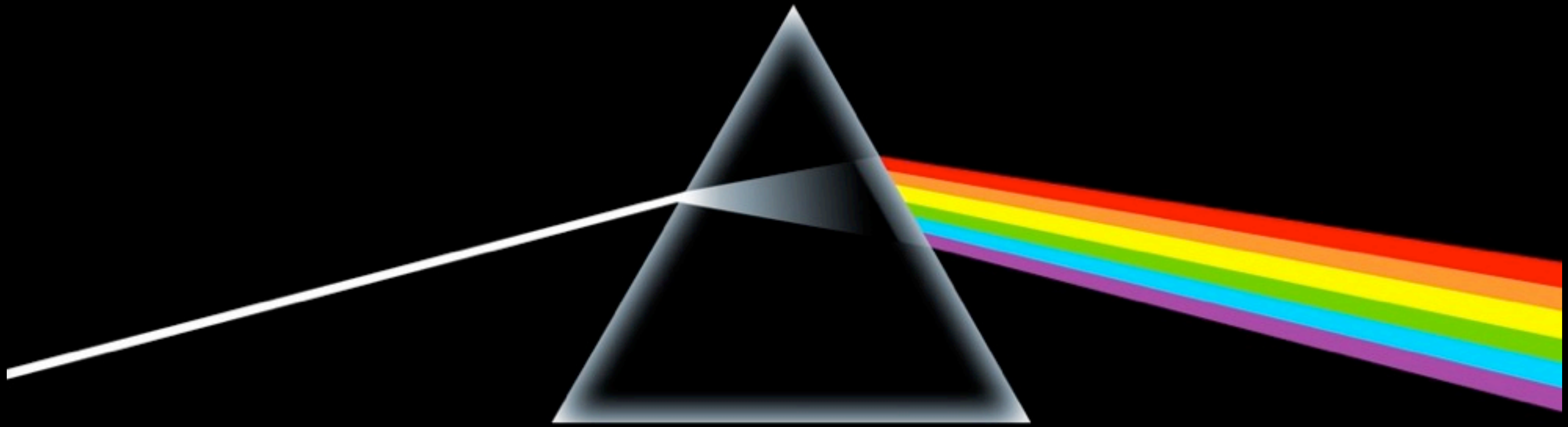
Ivan Lanese   Jorge A. Pérez   Davide Sangiorgi   Alan Schmitt

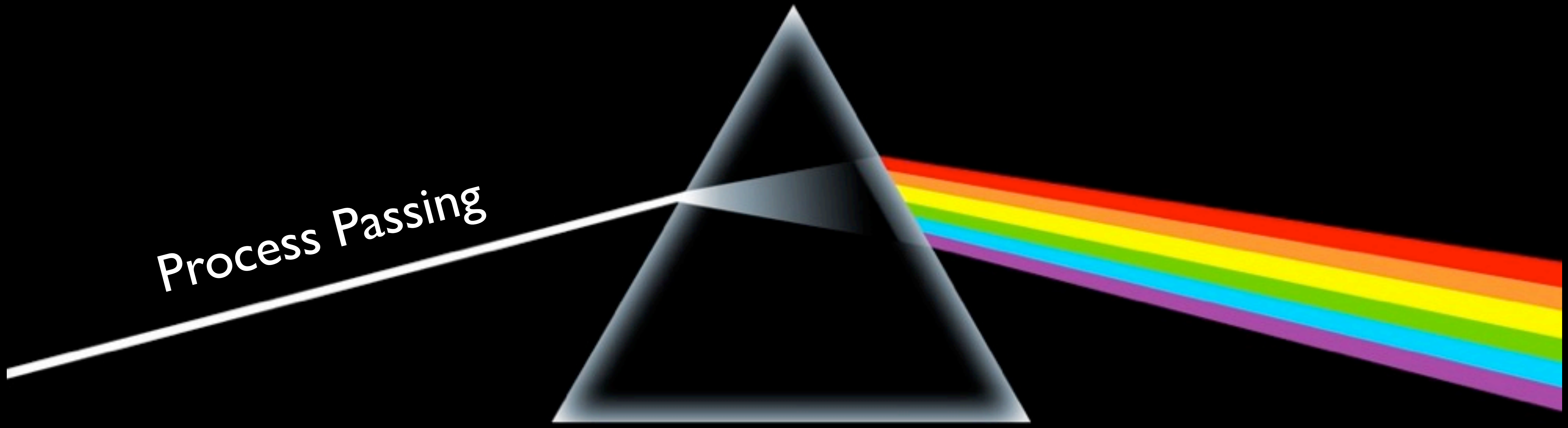
ICALP 2010, Bordeaux.

On the Expressiveness of Polyadic and Synchronous Communication in Higher-Order Process Calculi

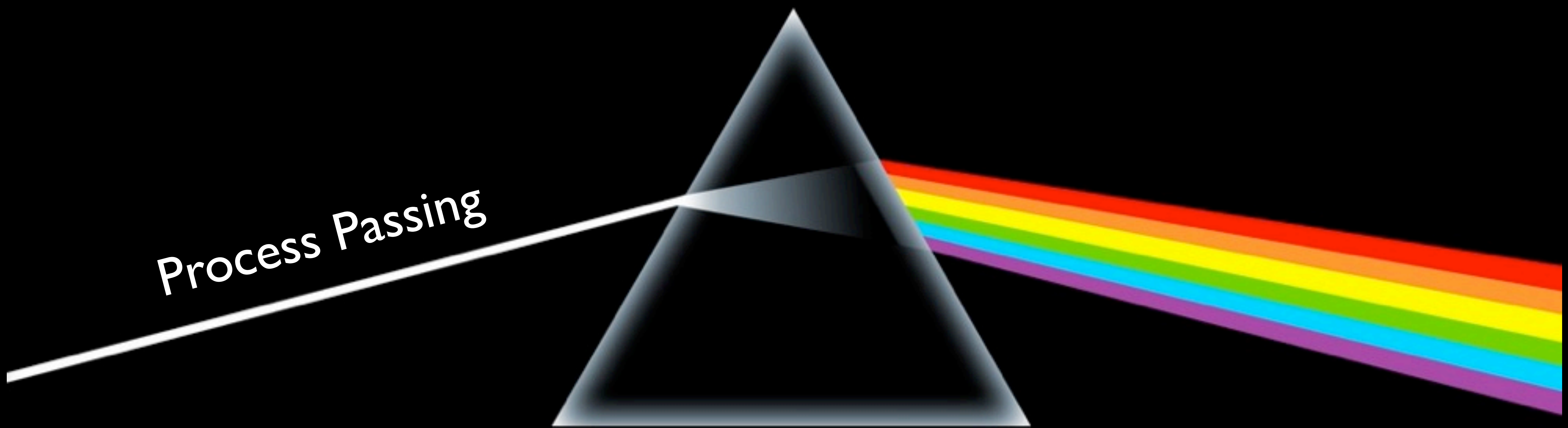
Ivan Lanese Jorge A. Pérez Davide Sangiorgi Alan Schmitt

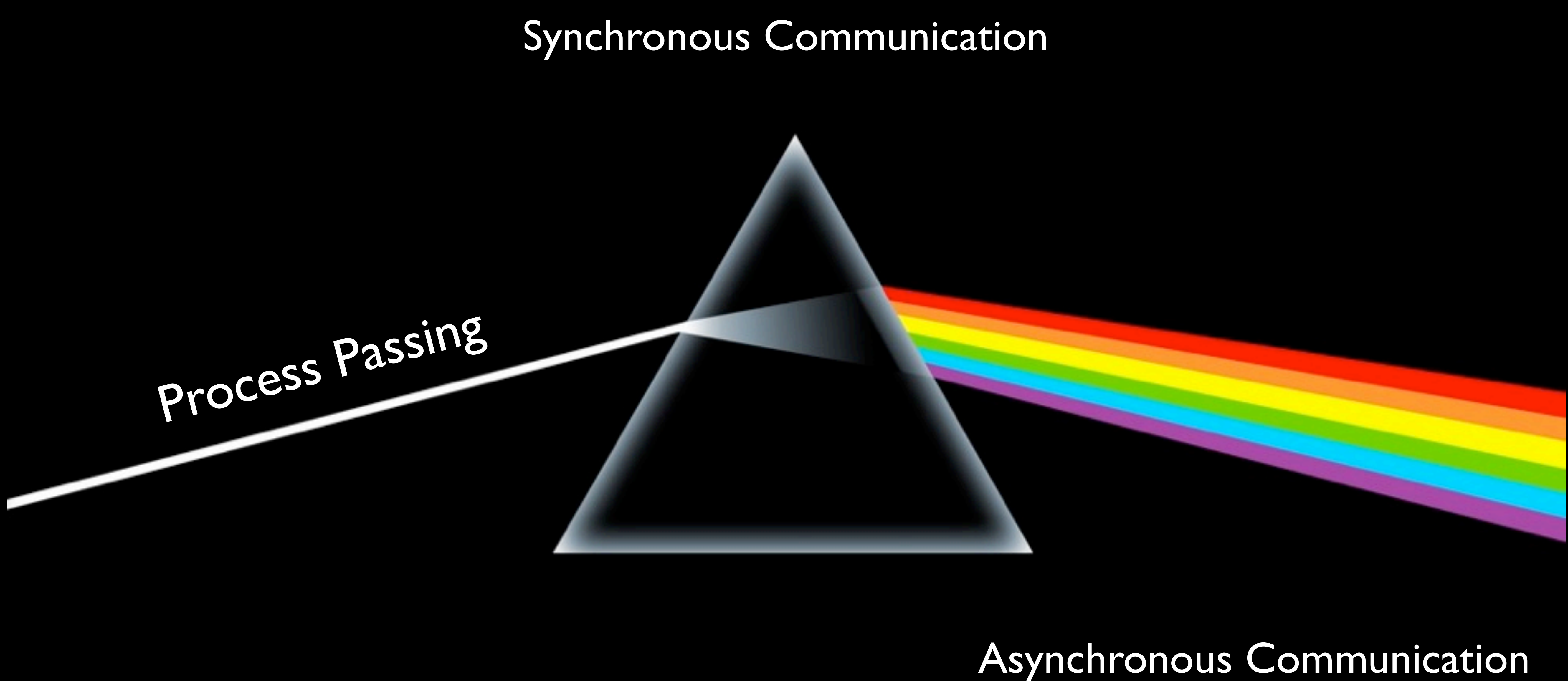
ICALP 2010, Bordeaux.



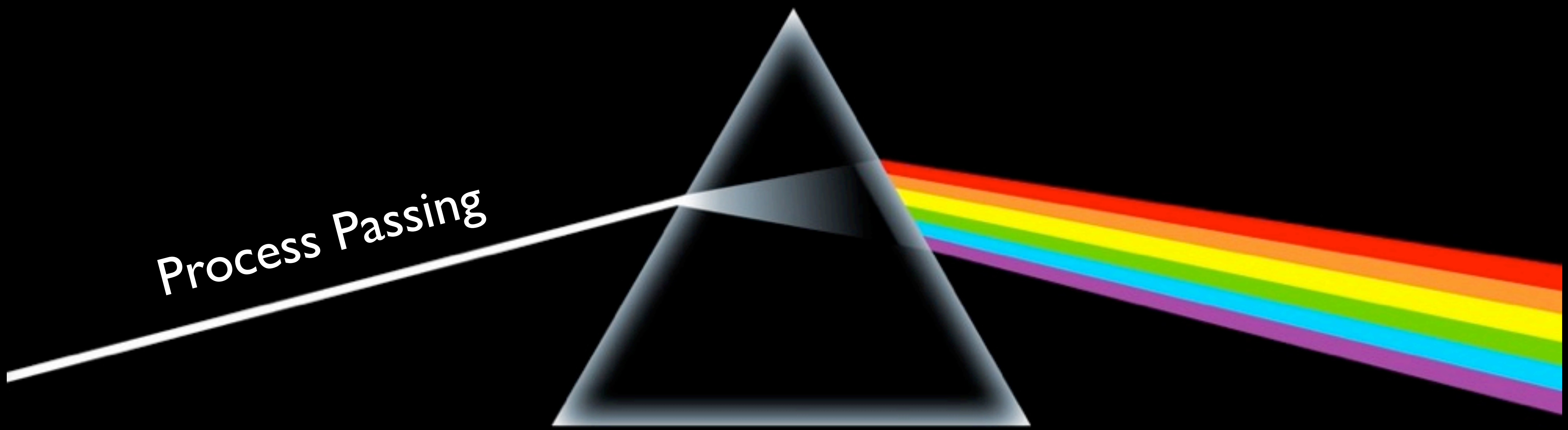

Process Passing

Synchronous Communication



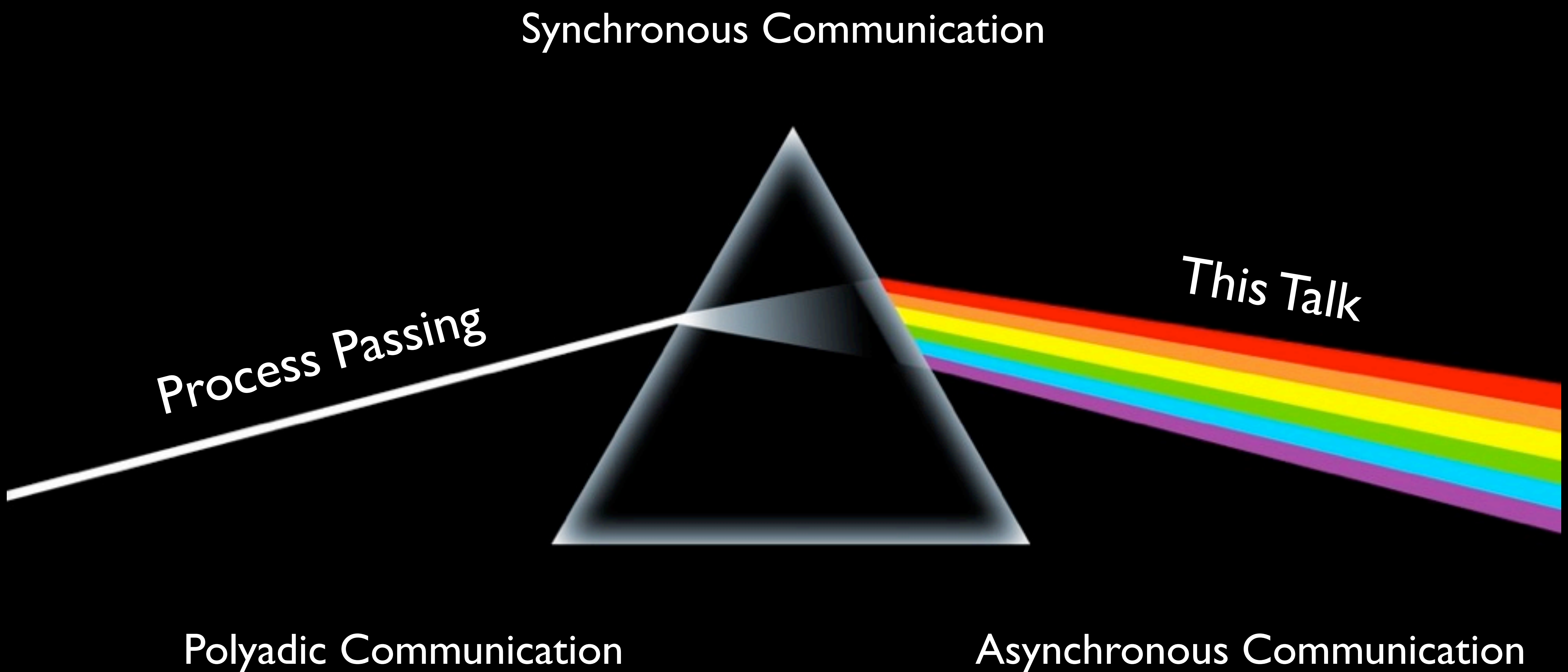

Synchronous Communication
Process Passing
Asynchronous Communication

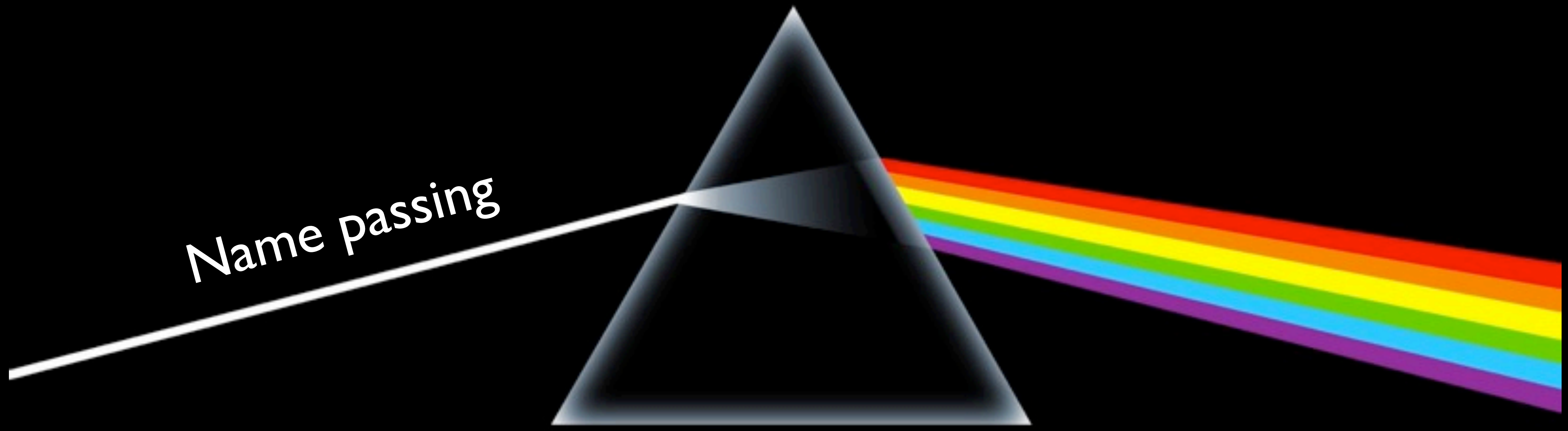
Synchronous Communication

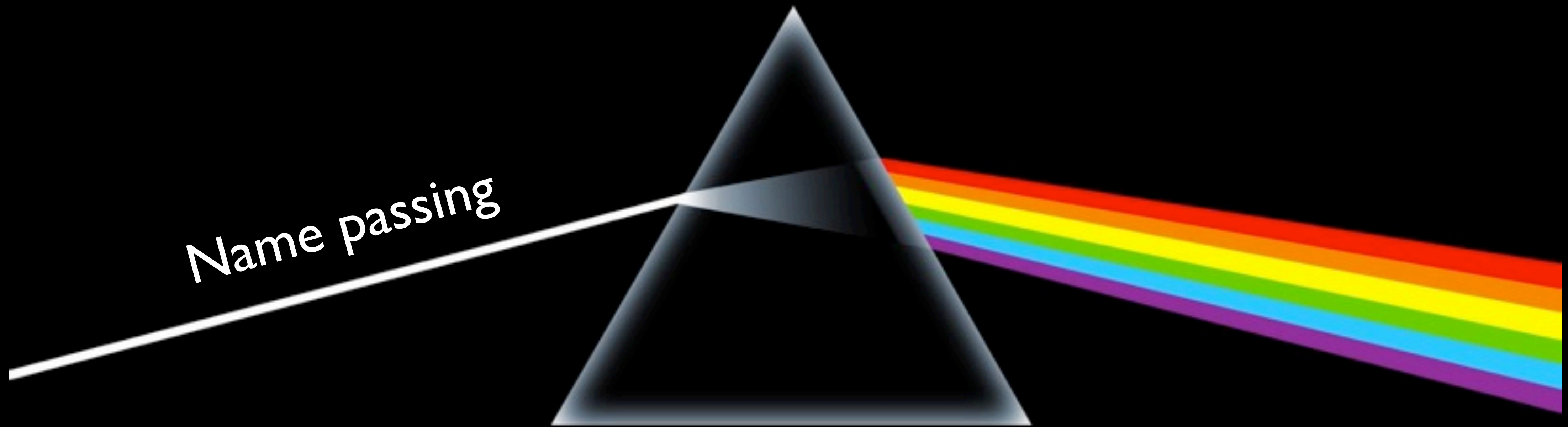


Polyadic Communication

Asynchronous Communication

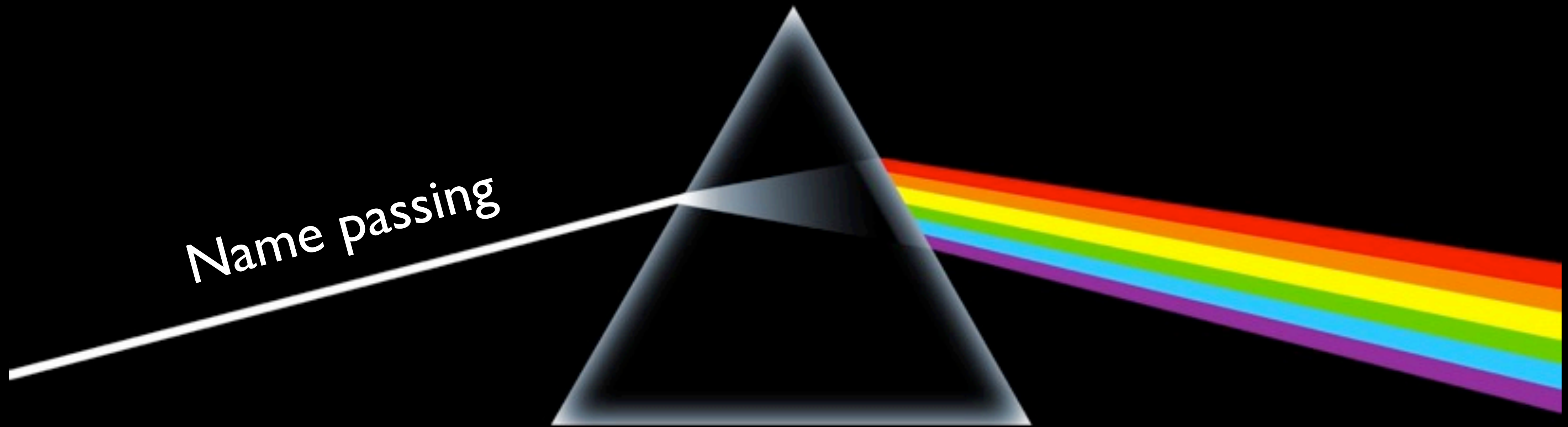

Synchronous Communication
Process Passing
This Talk
Polyadic Communication
Asynchronous Communication


Name passing

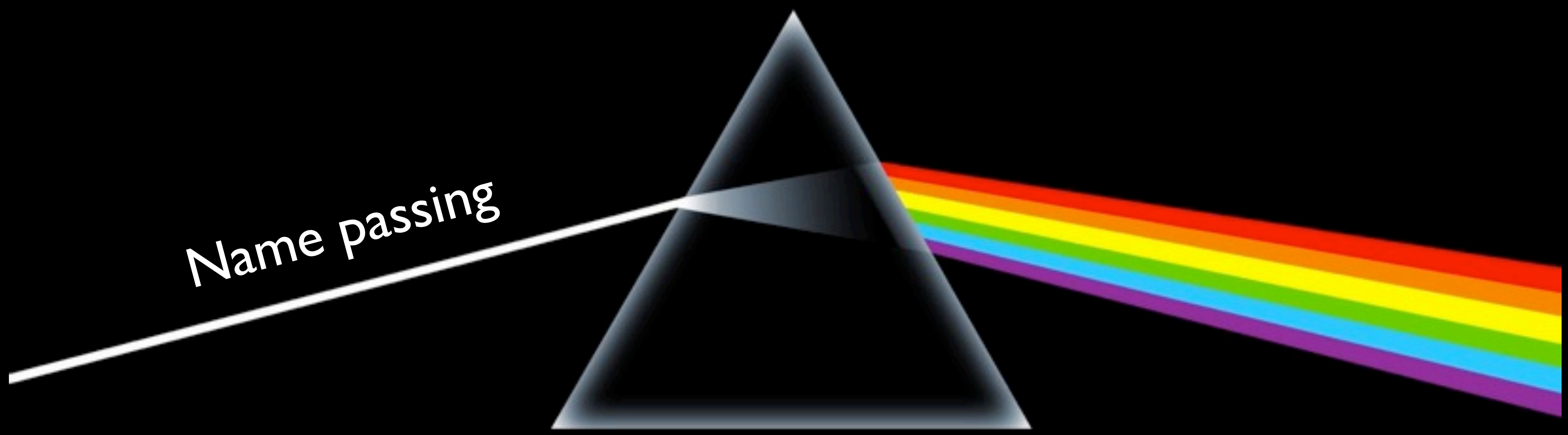


✦ Encodings of synchronous π-calculus *without choice* into asynchronous π [Honda-Tokoro91, Boudol92]

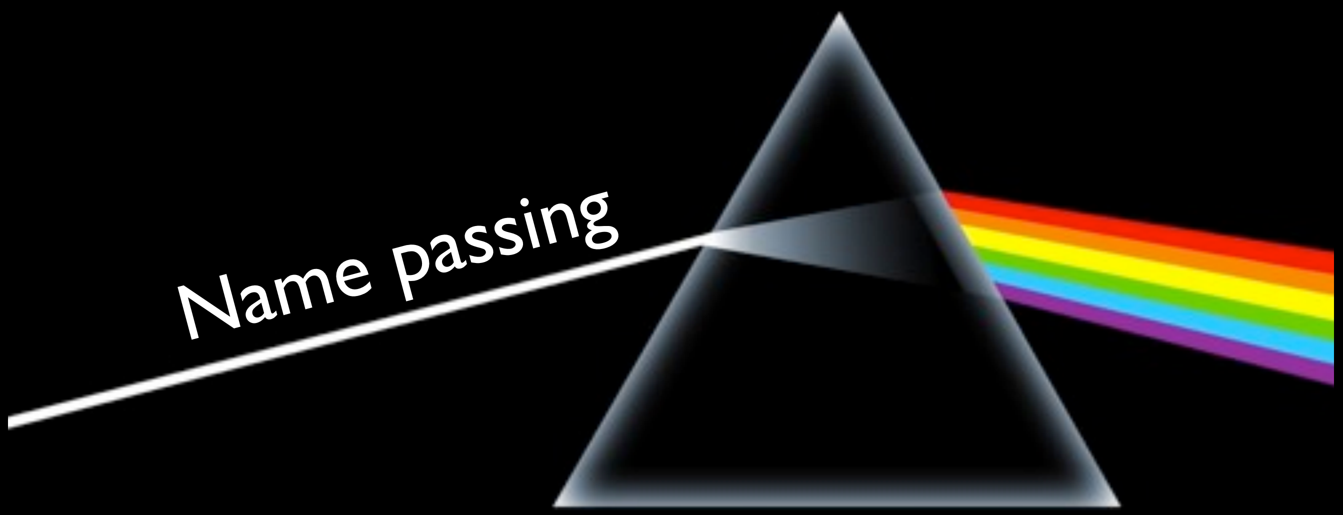
✦ Synchronous π-calculus with *mixed choice* is more expressive than the asynchronous π [Palamidessi96]



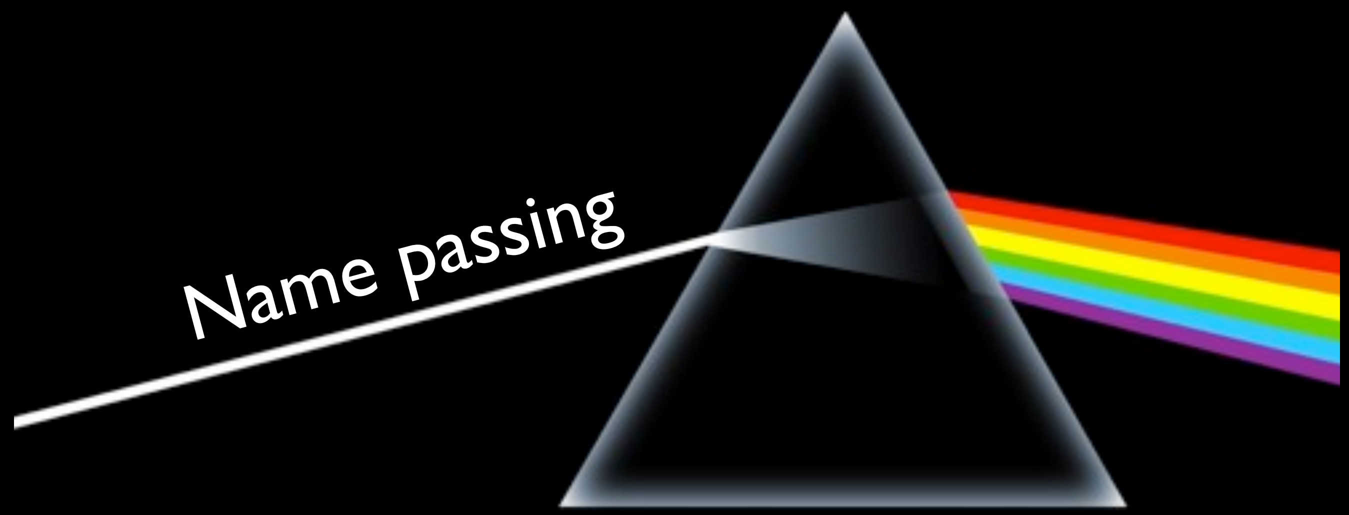
✦ Encodings of synchronous π-calculus *without choice* into asynchronous π [Honda-Tokoro91, Boudol92]
✦ Synchronous π-calculus with *mixed choice* is more expressive than the asynchronous π [Palamidessi96]
✦ The encoding of polyadic into monadic name passing [Milner93]



- Encodings of synchronous π-calculus *without choice* into asynchronous π [Honda-Tokoro91, Boudol92]
- Synchronous π-calculus with *mixed choice* is more expressive than the asynchronous π [Palamidessi96]
- The encoding of polyadic into monadic name passing [Milner93]
- The encoding of higher-order π-calculus (*name* AND *process* passing) into the π-calculus [Sangiorgi93]

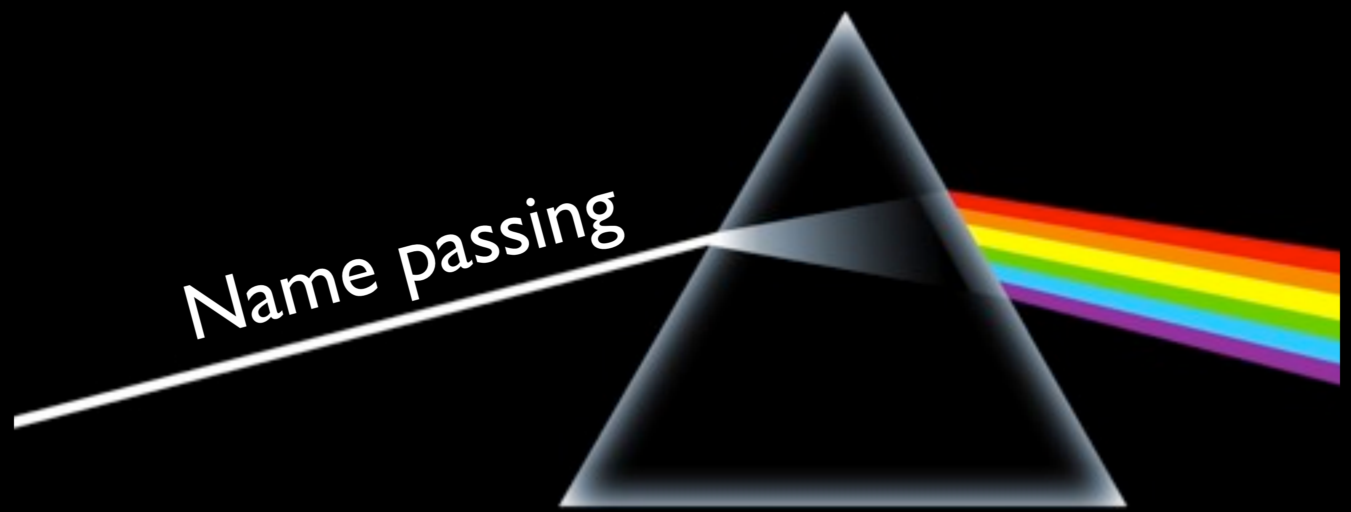


Relies on *private links*
by combining *restriction*
and *name passing*



Relies on *private links* by combining *restriction* and *name passing*

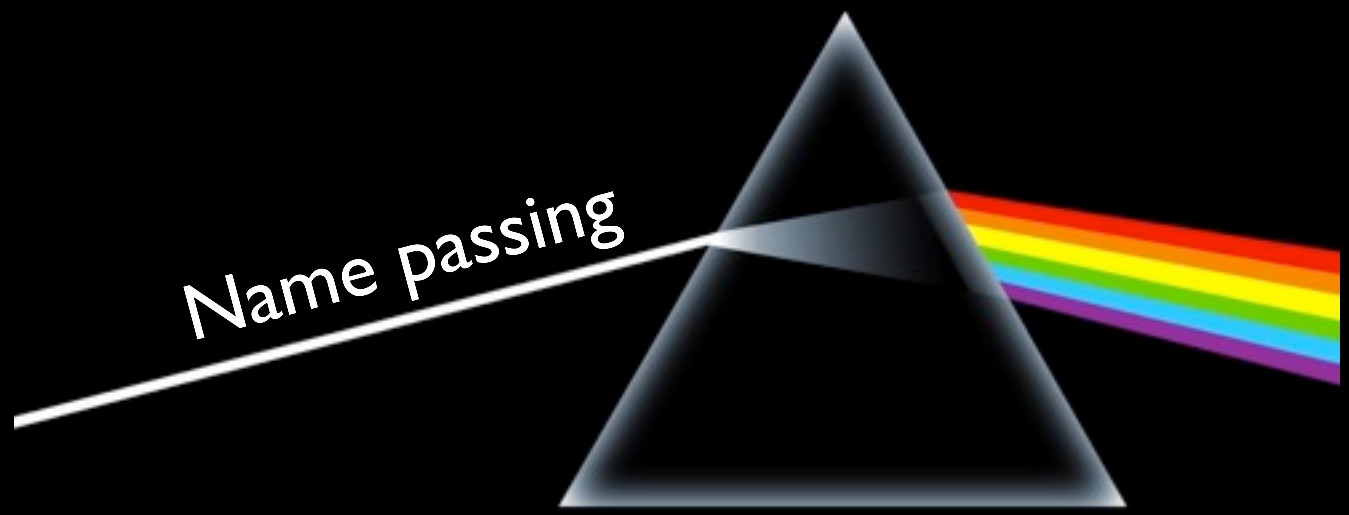
Example: *biadic into monadic* name passing



Relies on *private links* by combining *restriction* and *name passing*

Example: *biadic into monadic* name passing

$$
\begin{aligned}
[\![\overline{a}\langle m, n\rangle.P]\!] &= \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.[\![P]\!]) \\
[\![a(x, y).Q]\!] &= a(r).r(x).r(y).[\![Q]\!]
\end{aligned}
$$

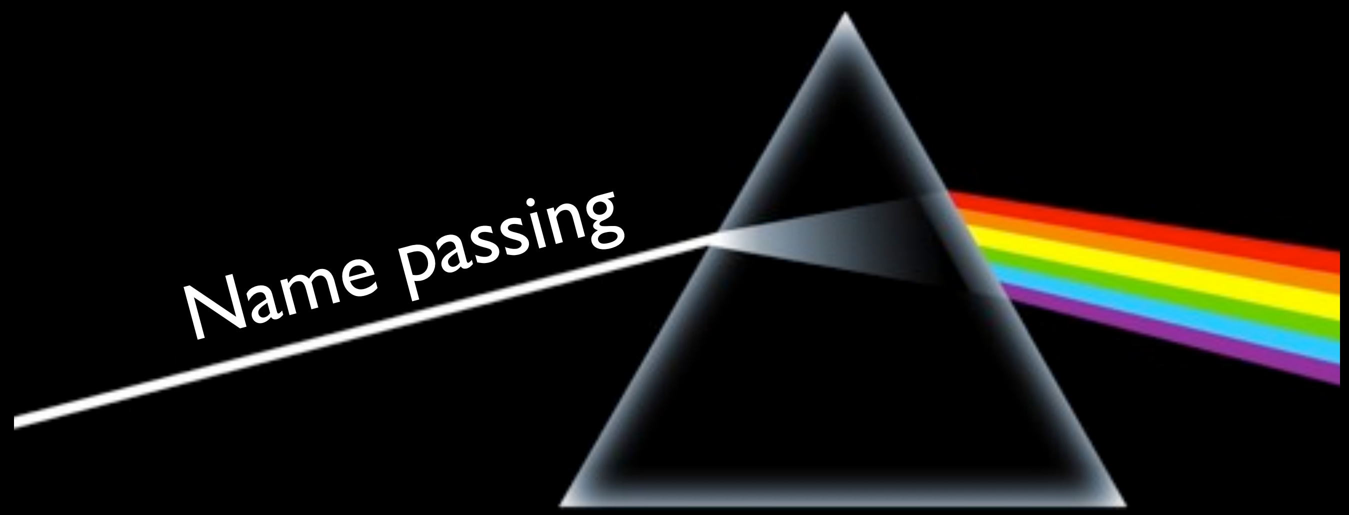


Relies on *private links* by combining *restriction* and *name passing*

Example: *biadic into monadic* name passing

$$
\begin{aligned}
[\![\overline{a}\langle m, n\rangle.P]\!] &= \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.[\![P]\!]) \\
[\![a(x, y).Q]\!] &= a(r).r(x).r(y).[\![Q]\!]
\end{aligned}
$$

Private links represent *agreements* on a restricted name

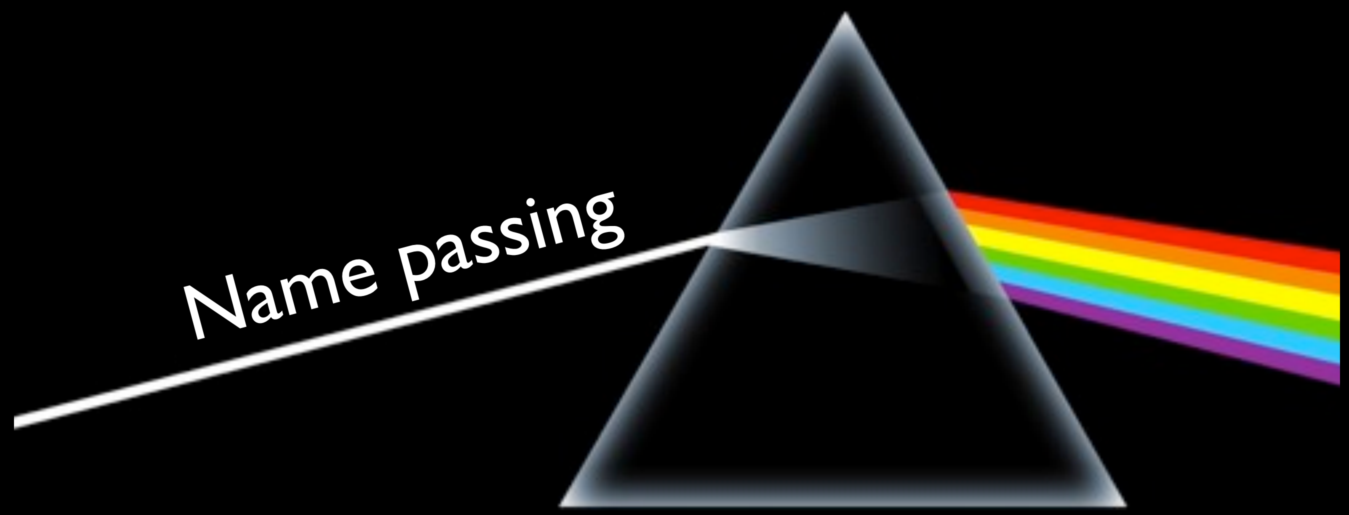


Relies on *private links*
by combining *restriction*
and *name passing*

Example: *biadic into monadic* name passing

$$\llbracket \overline{a}\langle m, n\rangle.P \rrbracket = \nu r\, (\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.\llbracket P \rrbracket)$$
$$\llbracket a(x, y).Q \rrbracket \; = a(r).r(x).r(y).\llbracket Q \rrbracket$$

Private links represent *agreements* on a restricted name



Relies on *private links* by combining *restriction* and *name passing*

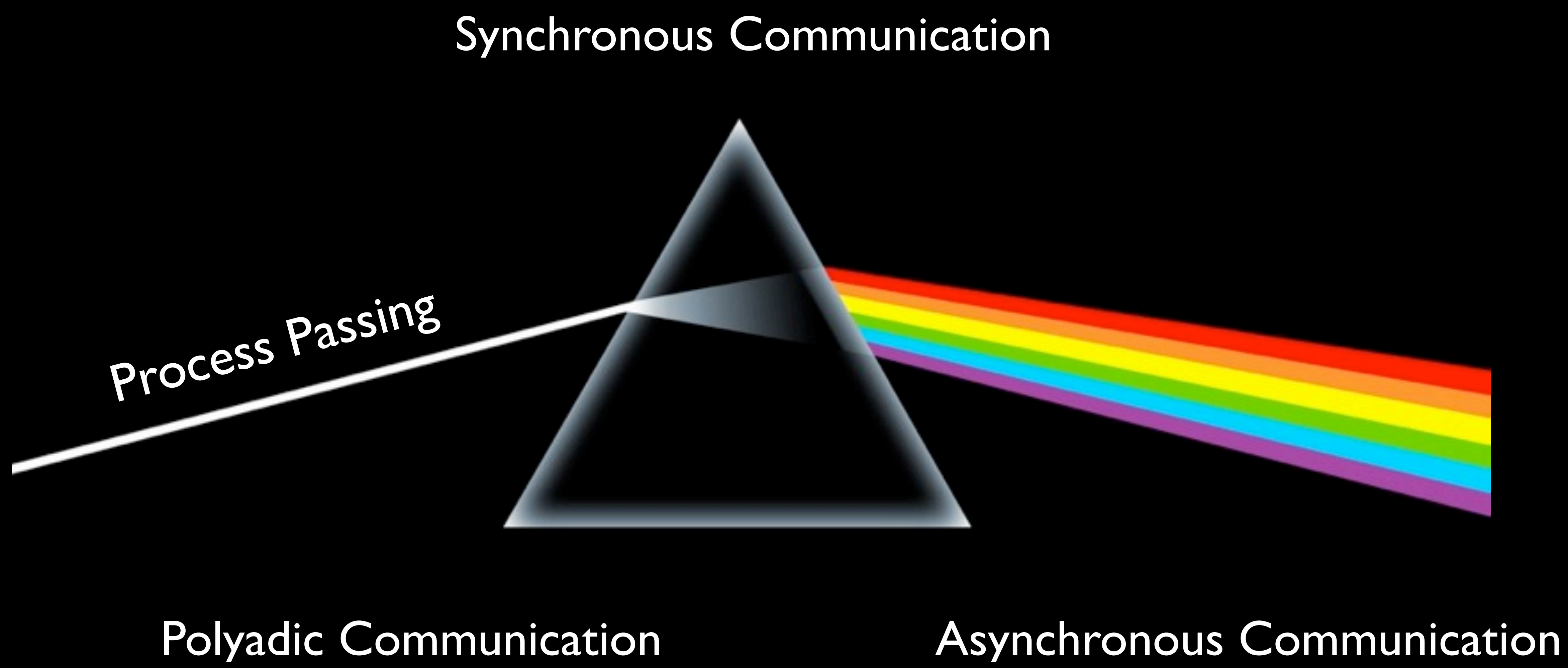
Example: *biadic into monadic* name passing

$$[\![\overline{a}\langle m,n\rangle.P]\!] = \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.[\![P]\!])$$
$$[\![a(x,y).Q]\!] \;= a(r).r(x).r(y).[\![Q]\!]$$

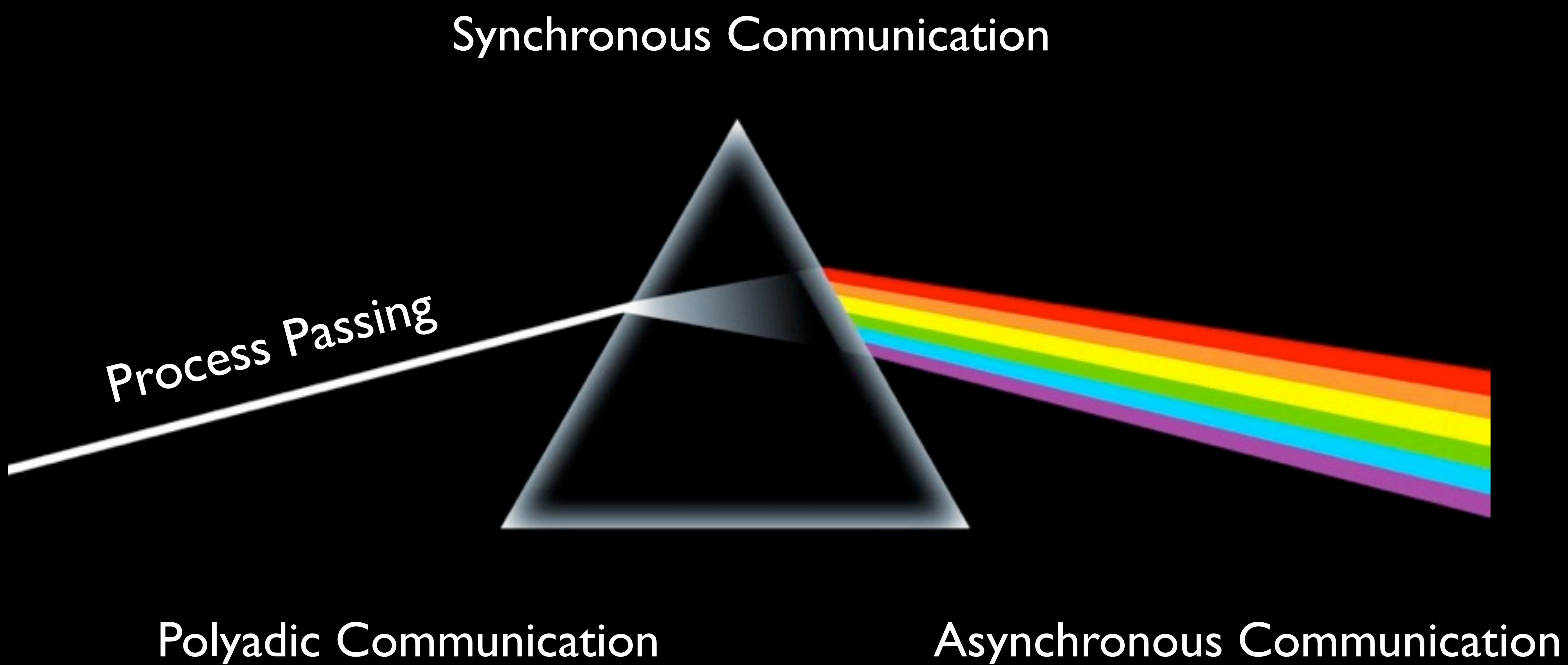
Private links represent *agreements* on a restricted name

Encodings are compact and robust wrt interferences

What about process passing?

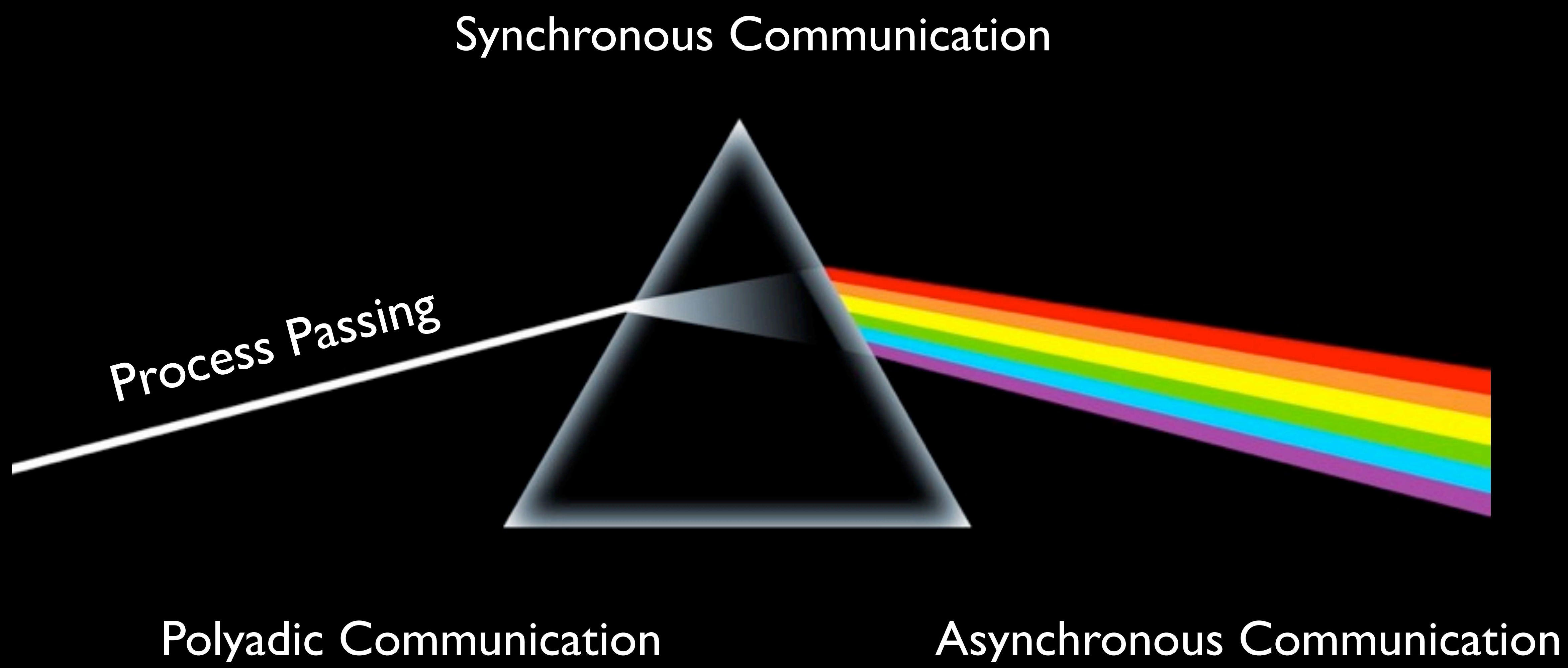


What about process passing?



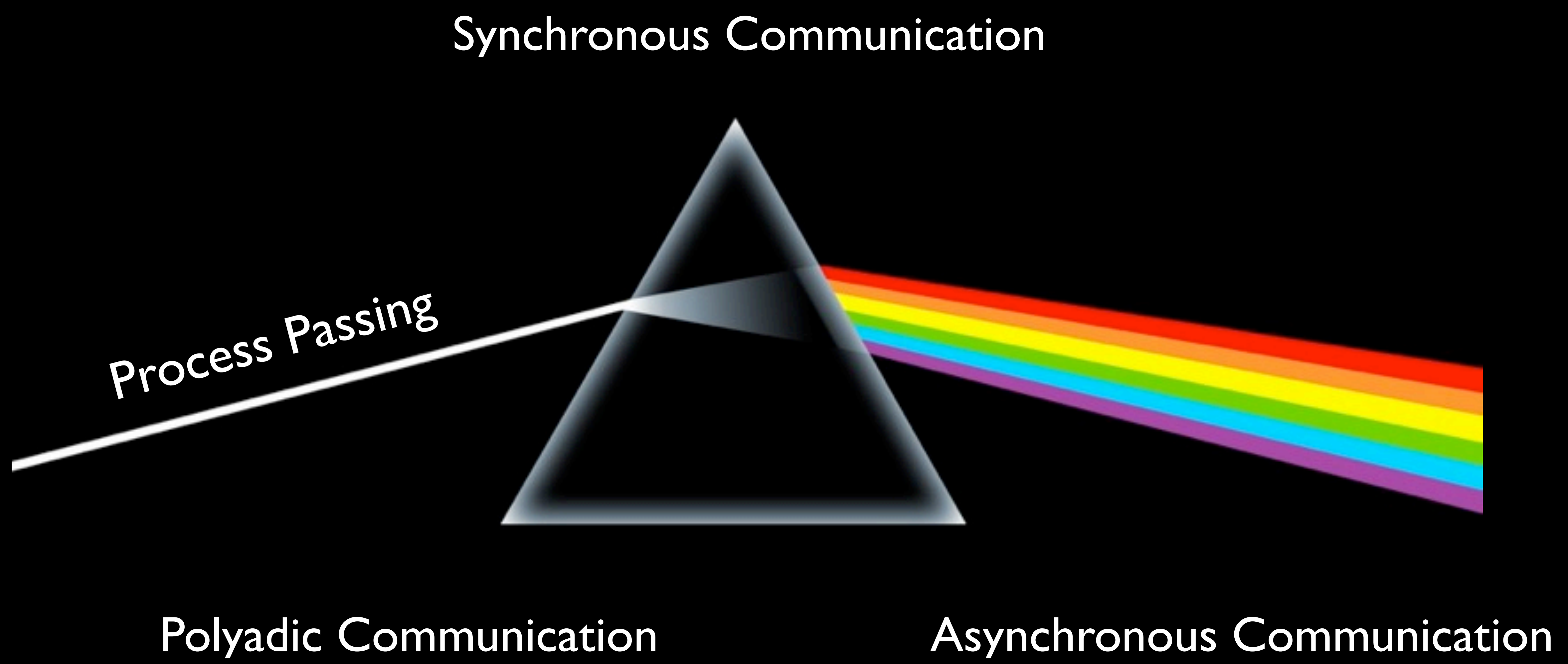
- No similar studies as in the name passing setting

What about process passing?



- No similar studies as in the name passing setting
- What if names are not considered?

What about process passing?



- No similar studies as in the name passing setting
- What if names are not considered?

Here: *pure* process passing

Processes as black boxes

Processes as black boxes

- They can only be *executed*, *forwarded*, or *discarded*

Processes as black boxes

- They can only be *executed*, *forwarded*, or *discarded*
- They *can contain* restricted names
  But a receiver cannot "dig into" the structure of a process. So it *cannot actually use* such names.

Processes as black boxes

- They can only be *executed*, *forwarded*, or *discarded*
- They *can contain* restricted names
  But a receiver cannot "dig into" the structure of a process. So it *cannot actually use* such names.
- "Hollow" scope extrusions: the scope expands but their effect is limited

Names *actually* used

Names *actually* used

Recall the encoding of biadic into monadic:

$$\llbracket \overline{a}\langle m, n\rangle.P \rrbracket = \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.\llbracket P \rrbracket)$$
$$\llbracket a(x, y).Q \rrbracket \;= a(r).r(x).r(y).\llbracket Q \rrbracket$$

Once received, *r* can be freely used

Names *actually* used

Recall the encoding of biadic into monadic:

$$\llbracket \overline{a}\langle m, n\rangle.P \rrbracket = \nu r\, (\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.\llbracket P \rrbracket)$$
$$\llbracket a(x, y).Q \rrbracket \ = a(r).r(x).r(y).\llbracket Q \rrbracket$$

Once received, *r* can be freely used

Two interacting process passing terms:

$$\nu n\, (\overline{a}\langle P\rangle.S') \parallel a(x).R'$$

Names *actually* used

Recall the encoding of biadic into monadic:

$$\llbracket \overline{a}\langle m, n\rangle.P \rrbracket = \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.\llbracket P \rrbracket)$$
$$\llbracket a(x,y).Q \rrbracket \ = a(r).r(x).r(y).\llbracket Q \rrbracket$$

Once received, *r* can be freely used

Two interacting process passing terms:

$$\nu n\,(\overline{a}\langle P\rangle.S') \parallel a(x).R' \longrightarrow \nu n\,(S' \parallel R'\{^{P}/x\})$$

Names *actually* used

Recall the encoding of biadic into monadic:

$$[\![\overline{a}\langle m, n\rangle.P]\!] = \nu r\,(\overline{a}\langle r\rangle.\overline{r}\langle m\rangle.\overline{r}\langle n\rangle.[\![P]\!])$$
$$[\![a(x, y).Q]\!] = a(r).r(x).r(y).[\![Q]\!]$$

Once received, *r* can be freely used

Two interacting process passing terms:

$$\nu n\,(\overline{a}\langle P\rangle.S') \parallel a(x).R' \longrightarrow \nu n\,(S' \parallel R'\{^{P}/x\})$$

In *R'*, name *n* <u>*can only be*</u> used as defined in *P and S'*

Our Results

1. Synchronous communication *can be encoded* into asynchronous communication
2. Polyadic communication of arity *n* *cannot be encoded* into communication of arity *n-1*
3. *Abstraction passing* *cannot be encoded* into polyadic communication

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\mid\; \overline{a}\langle\tilde{Q}\rangle.P \;\mid\; P_1 \parallel P_2 \;\mid\; \nu r\, P \;\mid\; x \;\mid\; \mathbf{0}$$

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\mid\; \overline{a}\langle\tilde{Q}\rangle.P \;\mid\; P_1 \parallel P_2 \;\mid\; \nu r\, P \;\mid\; x \;\mid\; \mathbf{0}$$

In the *asynchronous* variant (AHO^n) outputs have no continuations

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\mid\; \overline{a}\langle\tilde{Q}\rangle.P \;\mid\; P_1 \parallel P_2 \;\mid\; \nu r\, P \;\mid\; x \;\mid\; \mathbf{0}$$

In the *asynchronous* variant (AHO^n) outputs have no continuations

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\mid\; \overline{a}\langle\tilde{Q}\rangle.P \;\mid\; P_1 \parallel P_2 \;\mid\; \nu r\, P \;\mid\; x \;\mid\; \mathbf{0}$$

In the *asynchronous* variant (AHO^n) outputs have no continuations

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\mid\; \overline{a}\langle\tilde{Q}\rangle.P \;\mid\; P_1 \parallel P_2 \;\mid\; \nu r\, P \;\mid\; x \;\mid\; \mathbf{0}$$

In the *asynchronous* variant (AHO^n) outputs have no continuations

The variant with *abstraction passing* extends SHO^n with λ-like *abstractions* and *applications*:

The Languages

Synchronous pure process passing of arity *n* (SHO^n)

$$P, Q ::= a(\tilde{x}).P \;\;|\;\; \overline{a}\langle\tilde{Q}\rangle.P \;\;|\;\; P_1 \parallel P_2 \;\;|\;\; \nu r\, P \;\;|\;\; x \;\;|\;\; \mathbf{0}$$

In the *asynchronous* variant (AHO^n) outputs have no continuations

The variant with *abstraction passing* extends SHO^n with λ-like *abstractions* and *applications*:

$$P, Q ::= \cdots \;\;|\;\; (x)P \;\;|\; P_1\lfloor P_2 \rfloor$$

Semantics

- A Labeled Transition System (LTS) that enforces a closer look into synchronizations
- Two kinds of Internal behavior:
  - *internal* synchronizations τ
  - *public* synchronizations $a\tau$

The LTS for SHO^n

The LTS for SHO^n

$$\textsc{Inp}\ \frac{}{a(\widetilde{x}).\,P \xrightarrow{a(\widetilde{x})} P} \qquad \textsc{Out}\ \frac{}{\overline{a}\langle\widetilde{Q}\rangle.\,P \xrightarrow{\overline{a}\langle\widetilde{Q}\rangle} P} \qquad \textsc{Act1}\ \frac{P_1 \xrightarrow{\alpha} P_1' \quad \mathsf{cond}(\alpha, P_2)}{P_1 \parallel P_2 \xrightarrow{\alpha} P_1' \parallel P_2}$$

$$\textsc{Open}\ \frac{P \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}''\rangle} P' \quad r \neq a,\ r \in \mathsf{fn}(\widetilde{P}'') - \widetilde{s}}{\nu r\, P \xrightarrow{(\nu r\widetilde{s})\overline{a}\langle\widetilde{P}''\rangle} P'} \qquad \textsc{Res}\ \frac{P \xrightarrow{\alpha} P' \quad r \notin \mathsf{n}(\alpha)}{\nu r\, P \xrightarrow{\alpha} \nu r\, P'}$$

$$\textsc{Tau1}\ \frac{P_1 \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}\rangle} P_1' \quad P_2 \xrightarrow{a(\widetilde{x})} P_2' \quad \widetilde{s} \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \parallel P_2 \xrightarrow{a\tau} \nu\widetilde{s}\,(P_1' \parallel P_2'\{\widetilde{P}/\widetilde{x}\})} \qquad \textsc{IntRes}\ \frac{P \xrightarrow{a\tau} P'}{\nu a\, P \xrightarrow{\tau} \nu a\, P}$$

The LTS for SHO^n

$$\textsc{Inp}\ \frac{}{a(\widetilde{x}).\,P \xrightarrow{a(\widetilde{x})} P} \qquad \textsc{Out}\ \frac{}{\overline{a}\langle\widetilde{Q}\rangle.\,P \xrightarrow{\overline{a}\langle\widetilde{Q}\rangle} P} \qquad \textsc{Act1}\ \frac{P_1 \xrightarrow{\alpha} P_1' \quad \mathsf{cond}(\alpha, P_2)}{P_1 \parallel P_2 \xrightarrow{\alpha} P_1' \parallel P_2}$$

$$\textsc{Open}\ \frac{P \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P''}\rangle} P' \quad r \neq a,\, r \in \mathsf{fn}(\widetilde{P''}) - \widetilde{s}}{\nu r\, P \xrightarrow{(\nu r\widetilde{s})\overline{a}\langle\widetilde{P''}\rangle} P'} \qquad \textsc{Res}\ \frac{P \xrightarrow{\alpha} P' \quad r \notin \mathsf{n}(\alpha)}{\nu r\, P \xrightarrow{\alpha} \nu r\, P'}$$

$$\textsc{Tau1}\ \frac{P_1 \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}\rangle} P_1' \quad P_2 \xrightarrow{a(\widetilde{x})} P_2' \quad \widetilde{s} \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \parallel P_2 \xrightarrow{a\tau} \nu\widetilde{s}\,(P_1' \parallel P_2'\{\widetilde{P}/\widetilde{x}\})} \qquad \textsc{IntRes}\ \frac{P \xrightarrow{a\tau} P'}{\nu a\, P \xrightarrow{\tau} \nu a\, P}$$

The LTS for SHO^n

$$\textsc{Inp}\ \frac{}{a(\widetilde{x}).\,P \xrightarrow{a(\widetilde{x})} P} \qquad \textsc{Out}\ \frac{}{\overline{a}\langle\widetilde{Q}\rangle.\,P \xrightarrow{\overline{a}\langle\widetilde{Q}\rangle} P} \qquad \textsc{Act1}\ \frac{P_1 \xrightarrow{\alpha} P_1' \quad \mathsf{cond}(\alpha, P_2)}{P_1 \parallel P_2 \xrightarrow{\alpha} P_1' \parallel P_2}$$

$$\textsc{Open}\ \frac{P \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P''}\rangle} P' \quad r \neq a,\, r \in \mathsf{fn}(\widetilde{P''}) - \widetilde{s}}{\nu r\, P \xrightarrow{(\nu r\widetilde{s})\overline{a}\langle\widetilde{P''}\rangle} P'} \qquad \textsc{Res}\ \frac{P \xrightarrow{\alpha} P' \quad r \notin \mathsf{n}(\alpha)}{\nu r\, P \xrightarrow{\alpha} \nu r\, P'}$$

$$\textsc{Tau1}\ \frac{P_1 \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}\rangle} P_1' \quad P_2 \xrightarrow{a(\widetilde{x})} P_2' \quad \widetilde{s} \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \parallel P_2 \xrightarrow{a\tau} \nu\widetilde{s}\,(P_1' \parallel P_2'\{\widetilde{P}/\widetilde{x}\})} \qquad \textsc{IntRes}\ \frac{P \xrightarrow{a\tau} P'}{\nu a\, P \xrightarrow{\tau} \nu a\, P'}$$

The LTS for SHO^n

$$\textsc{Inp}\ \frac{}{a(\widetilde{x}).\,P \xrightarrow{a(\widetilde{x})} P} \qquad \textsc{Out}\ \frac{}{\overline{a}\langle\widetilde{Q}\rangle.\,P \xrightarrow{\overline{a}\langle\widetilde{Q}\rangle} P} \qquad \textsc{Act1}\ \frac{P_1 \xrightarrow{\alpha} P_1' \quad \mathsf{cond}(\alpha, P_2)}{P_1 \parallel P_2 \xrightarrow{\alpha} P_1' \parallel P_2}$$

$$\textsc{Open}\ \frac{P \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}''\rangle} P' \quad r \neq a,\ r \in \mathsf{fn}(\widetilde{P}'') - \widetilde{s}}{\nu r\, P \xrightarrow{(\nu r\widetilde{s})\overline{a}\langle\widetilde{P}''\rangle} P'} \qquad \textsc{Res}\ \frac{P \xrightarrow{\alpha} P' \quad r \notin \mathsf{n}(\alpha)}{\nu r\, P \xrightarrow{\alpha} \nu r\, P'}$$

$$\textsc{Tau1}\ \frac{P_1 \xrightarrow{(\nu\widetilde{s})\overline{a}\langle\widetilde{P}\rangle} P_1' \quad P_2 \xrightarrow{a(\widetilde{x})} P_2' \quad \widetilde{s} \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \parallel P_2 \xrightarrow{a\tau} \nu\widetilde{s}\,(P_1' \parallel P_2'\{\widetilde{P}/\widetilde{x}\})} \qquad \textsc{IntRes}\ \frac{P \xrightarrow{a\tau} P'}{\nu a\, P \xrightarrow{\tau} \nu a\, P}$$

Rule for the variant with abstraction passing:

$$\textsc{App}\ \frac{}{(x)P\lfloor Q\rfloor \xrightarrow{\tau} P\{Q/x\}}$$

The notion of encoding (Informally)

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

Encoding: a translation plus syntactic/semantic conditions

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

Encoding: a translation plus syntactic/semantic conditions

Syntactic: Compositionality - Name invariance

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

Encoding: a translation plus syntactic/semantic conditions

Syntactic: Compositionality - Name invariance
Semantic: Operational Correspondence - Adequacy

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

Encoding: a translation plus syntactic/semantic conditions

Syntactic: Compositionality - Name invariance
Semantic: Operational Correspondence - Adequacy
Divergence Reflection

The notion of encoding (Informally)

Language: an algebra of processes, an LTS, and a weak behavioral equivalence

Translation: a function from language $\mathcal{L}_1$ to language $\mathcal{L}_2$

Encoding: a translation plus syntactic/semantic conditions

Syntactic: Compositionality - Name invariance
Semantic: Operational Correspondence - Adequacy
Divergence Reflection

Encodings are *composable*: the composition of two encodings is an encoding


Encoding Synchronous into
Asynchronous

Synchronous into Asynchronous

Synchronous into Asynchronous

- Encoding SHO^n into AHO^{n+1} is easy

Synchronous into Asynchronous

- Encoding SHO^n into AHO^{n+1} is easy

  Send all the objects as they are, and use an ***extra parameter*** to send the continuation of the output

Synchronous into Asynchronous

- Encoding SHO^n into AHO^{n+1} is easy

  Send all the objects as they are, and use an ***extra parameter*** to send the continuation of the output

- Challenge: to encode SHO^n into AHO^n

Synchronous into Asynchronous

- Encoding SHO^n into AHO^{n+1} is easy

  Send all the objects as they are, and use an *extra parameter* to send the continuation of the output

- Challenge: to encode SHO^n into AHO^n

  Our solution: Send the first n-1 objects as they are, and use the n-th object to send BOTH the last object AND the continuation

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle\, k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\,\rangle \parallel \bar{l})$$

$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle\, k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\,\rangle \parallel \overline{l})$$

$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$
$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$

$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$
$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*
  - The continuation is triggered only once

Encoding Synchronous into Asynchronous

The basic case: SHO[1] into AHO[1]

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$
$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*
  - The continuation is triggered only once

Encoding Synchronous into Asynchronous

The basic case: SHO[1] into AHO[1]

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\,(\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$

$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*
  - The continuation is triggered only once
  - A trigger on *k* is always available

Encoding Synchronous into Asynchronous

The basic case: SHO^1 into AHO^1

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$
$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*
  - The continuation is triggered only once
  - A trigger on *k* is always available

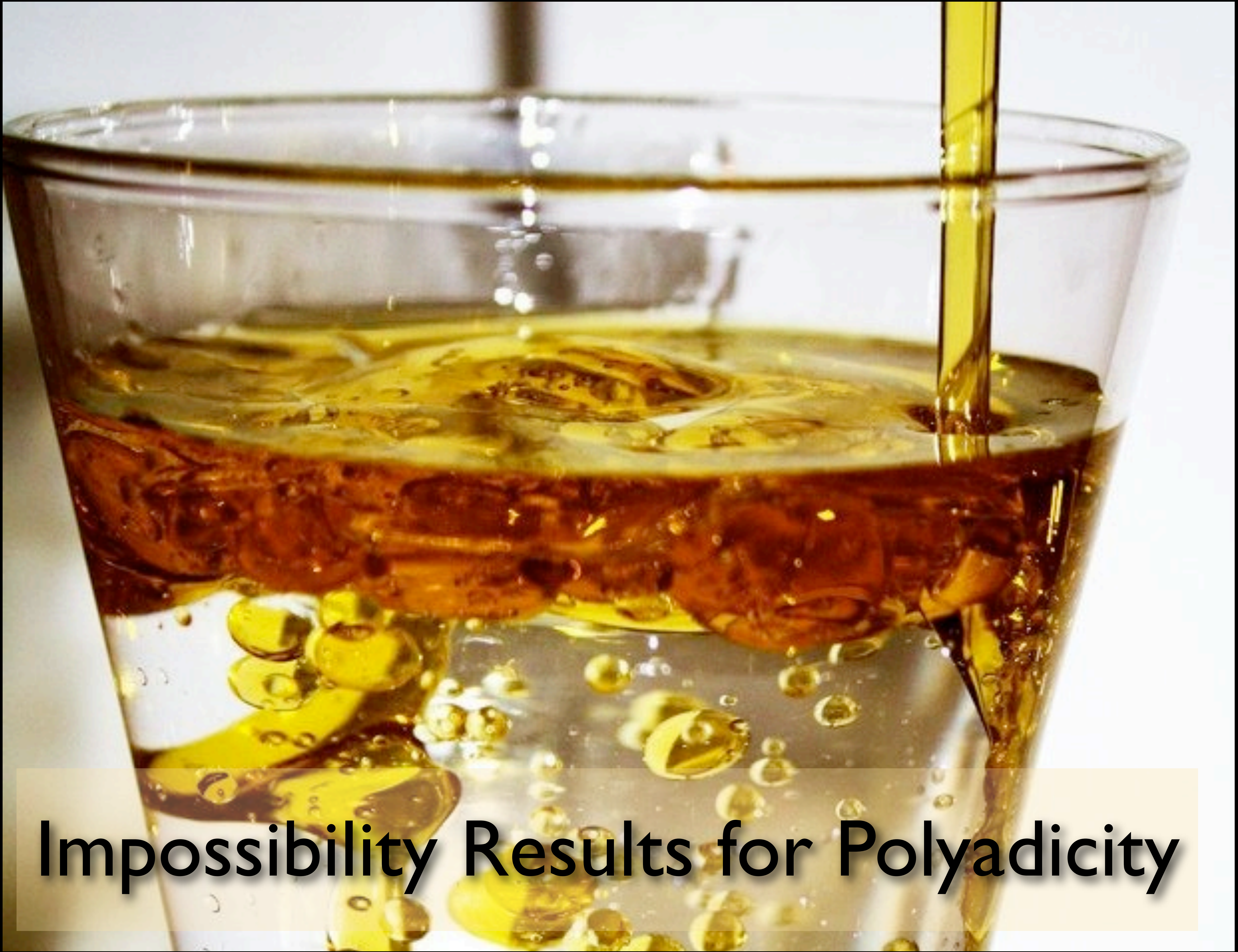
Encoding Synchronous into Asynchronous

The basic case: SHO[1] into AHO[1]

$$[\![\overline{a}\langle P\rangle.S]\!] = \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l})$$
$$[\![a(x).R]\!] = a(x).(x \parallel [\![R]\!])$$

The encoding is a homomorphism for the other operators.
Guarded choice is a derived construct in SHO^n

- Object and continuation together in a guarded sum
- Two triggers: *k* for object *P* and *l* for continuation *S*
  - The continuation is triggered only once
  - A trigger on *k* is always available
- The generalization to the *n*-adic case is immediate



Impossibility Results for Polyadicity

Restricted names are like oil and water

- They do not really “mix” after communications --- “hollow” extrusions
- This separation prevents private link establishment

Restricted names are like oil and water

- They do not really "mix" after communications --- "hollow" extrusions
- This separation prevents private link establishment

Our approach to separation:

- *Disjoint form*: our way of formalizing separation of restricted names *after* a public synchronization
- *Stability conditions*: when/how processes remain in disjoint form along computations

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q$$

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q \xrightarrow{a\tau} \nu\tilde{n}\,(P \parallel Q\{{}^{R_1, R_2}/{}_{x_1, x_2}\})$$

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q \xrightarrow{a\tau} \nu\tilde{n}\,(P \parallel Q\{{}^{R_1, R_2}/{x_1, x_2}\})$$
$$= \nu\tilde{n}\,(P \parallel C[R_1, R_2])$$

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\begin{aligned}\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q &\xrightarrow{a\tau} \nu\tilde{n}\,(P \parallel Q\{^{R_1, R_2}/_{x_1, x_2}\}) \\ &= \;\; \nu\tilde{n}\,(P \parallel C[R_1, R_2])\end{aligned}$$

The scope expands but this is a hollow extrusion

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q \xrightarrow{a\tau} \nu\tilde{n}\,(P \parallel Q\{R_1, R_2/x_1, x_2\})$$
$$= \nu\tilde{n}\,(P \parallel C[R_1, R_2])$$

The scope expands but this is a hollow extrusion

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\begin{aligned}\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q \xrightarrow{a\tau} & \ \nu\tilde{n}\,(P \parallel Q\{^{R_1, R_2}/_{x_1, x_2}\}) \\ = & \ \nu\tilde{n}\,(P \parallel C[R_1, R_2])\end{aligned}$$

The scope expands but this is a hollow extrusion

Even if R_1, R_2 are *inside* C, they do not share private names

Disjoint Forms

Two biadic processes that do not share private names
They can communicate through a public name:

$$\nu\tilde{n}\,(\overline{a}\langle R_1, R_2\rangle.P) \parallel a(x_1, x_2).Q \xrightarrow{a\tau} \nu\tilde{n}\,(P \parallel Q\{R_1, R_2/x_1, x_2\})$$
$$= \nu\tilde{n}\,(P \parallel C[R_1, R_2])$$

The scope expands but this is a hollow extrusion

Even if R_1, R_2 are *inside* C, they do not share private names

The private names of C and those of P, R_1, R_2 are *disjoint*

Disjoint Forms

Definition 10 (Disjoint Form) *Let* $T \equiv \nu\widetilde{n}(P \parallel C[\widetilde{R}])$ *be a* SHO^m *process where*

1. $\widetilde{n}$ *is a set of names such that* $\widetilde{n} \subseteq \mathsf{fn}(P, \widetilde{R})$ *and* $\widetilde{n} \cap \mathsf{fn}(C) = \emptyset$*;*
2. C *is a k-ary (guarded, multihole) context;*
3. $\widetilde{R}$ *contains k closed processes.*

We then say that T *is in* k-adic disjoint form with respect to $\widetilde{n}$, $\widetilde{R}$, and P.

Disjoint Forms

Definition 10 (Disjoint Form) *Let* $T \equiv \nu\widetilde{n}(P \parallel C[\widetilde{R}])$ *be a* SHO^m *process where*

1. $\widetilde{n}$ *is a set of names such that* $\widetilde{n} \subseteq \mathsf{fn}(P, \widetilde{R})$ *and* $\widetilde{n} \cap \mathsf{fn}(C) = \emptyset$*;*
2. C *is a* k*-ary (guarded, multihole) context;*
3. $\widetilde{R}$ *contains* k *closed processes.*

We then say that T *is in* k-adic disjoint form with respect to $\widetilde{n}$, $\widetilde{R}$, and P.

Disjoint Forms

Definition 10 (Disjoint Form) *Let* $T \equiv \nu\widetilde{n}(P \parallel C[\widetilde{R}])$ *be a* SHO^m *process where*

1. $\widetilde{n}$ *is a set of names such that* $\widetilde{n} \subseteq \mathsf{fn}(P, \widetilde{R})$ *and* $\widetilde{n} \cap \mathsf{fn}(C) = \emptyset$*;*
2. C *is a k-ary (guarded, multihole) context;*
3. $\widetilde{R}$ *contains k closed processes.*

We then say that T *is in* k-adic disjoint form with respect to $\widetilde{n}$, $\widetilde{R}$, and P.

Stability Conditions:

Disjoint forms are preserved by internal synchronizations and certain output actions

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

Proof Sketch:

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

Proof Sketch:

1. Assume such an encoding exists

The impossibility result

Theorem. *There is no encoding of* SHO^2 *into* SHO^1

Proof Sketch:

1. Assume such an encoding exists

2. Take a process *P* that makes a public (biadic) synchronization.

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

<u>Proof Sketch:</u>

1. Assume such an encoding exists

2. Take a process *P* that makes a public (biadic) synchronization.

   Two processes with different behavior are sent; the receiver non-deterministically discards one and executes the other.

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

<u>*Proof Sketch:*</u>

1. Assume such an encoding exists

2. Take a process *P* that makes a public (biadic) synchronization.

   Two processes with different behavior are sent; the receiver non-deterministically discards one and executes the other.

3. Show that the encoding of *P* mimics such communication and gets into <u>*monadic disjoint form (MDF)*</u>

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

Proof Sketch:

1. Assume such an encoding exists
2. Take a process *P* that makes a public (biadic) synchronization.

   Two processes with different behavior are sent; the receiver non-deterministically discards one and executes the other.
3. Show that the encoding of *P* mimics such communication and gets into *monadic disjoint form (MDF)*
4. Show that the MDF is preserved along relevant computations

The impossibility result

Theorem. *There is no encoding of SHO^2 into SHO^1*

Proof Sketch:

1. Assume such an encoding exists
2. Take a process *P* that makes a public (biadic) synchronization.

   Two processes with different behavior are sent; the receiver non-deterministically discards one and executes the other.
3. Show that the encoding of *P* mimics such communication and gets into *monadic disjoint form (MDF)*
4. Show that the MDF is preserved along relevant computations
5. Using a causality analysis, show that the (limited) structure of the MDF causes the encoding of *P* to exhibit behavior that *P* doesn't have: contradiction.

The hierarchy

Theorem. *There is no encoding of* SHO^n *into* SHO^{n-1}, *for every* $n>1$

The hierarchy

Theorem. *There is no encoding of* SHO^n *into* SHO^{n-1}, *for every* $n>1$

- Proofs follow by an extension of all notions and auxiliary results
- The hierarchy holds also for asynchronous calculi


The power of abstraction passing

Abstraction passing

- Sending functions as in the λ-calculus
- It is specific to the higher-order setting --- not present in the name passing
- We only consider abstractions of order 1: functions from processes to processes

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$\begin{aligned}
[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] &= a(z).\,([\![R]\!] \parallel \nu m\, n\, c\,(\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \\
&\qquad \parallel c.\, z\lfloor [\![P_2]\!] \rfloor)) \\
[\![a(x_1, x_2).\, Q]\!] &= \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\,[\![Q]\!]))
\end{aligned}$$

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\,([\![R]\!] \parallel \nu m\, n\, c\,(\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!] \rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\, [\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing ($SHO^1{}_a$):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\, ([\![R]\!] \parallel \nu m\, n\, c\, (\overline{n} \parallel z\lfloor n.\, (\overline{c} \parallel \overline{m}) + m.\, ([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!] \rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\, (\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\, (x_1 \parallel b(x_2).\, [\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\,([\![R]\!] \parallel \nu m\, n\, c\,(\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!]\rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\,[\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*
- A private link on *b* is realized upon application

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\,([\![R]\!] \parallel \nu m\, n\, c\,(\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!] \rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\,[\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*
- A private link on *b* is realized upon application

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$\llbracket \overline{a}\langle P_1, P_2\rangle .\, R \rrbracket = a(z).\, (\llbracket R \rrbracket \parallel \nu m\, n\, c\, (\overline{n} \parallel z\lfloor n.\, (\overline{c} \parallel \overline{m}) + m.\, (\llbracket P_1 \rrbracket \parallel \overline{m})\rfloor \parallel c.\, z\lfloor \llbracket P_2 \rrbracket \rfloor))$$

$$\llbracket a(x_1, x_2).\, Q \rrbracket = \nu b\, (\overline{a}\langle (y)\overline{b}\langle y\rangle \rangle \parallel b(x_1).\, (x_1 \parallel b(x_2).\, \llbracket Q \rrbracket))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*
- A private link on *b* is realized upon application

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\,([\![R]\!] \parallel \nu m\, n\, c\,(\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!] \rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle \rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\,[\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*
- A private link on *b* is realized upon application
- Communication objects P_1, P_2 are “activated” from the encoding of output

Private links with abstraction passing

Encoding SHO^2 into SHO^1 with abstraction passing (SHO^1_a):

$$[\![\overline{a}\langle P_1, P_2\rangle.\, R]\!] = a(z).\,([\![R]\!] \parallel \nu m\, n\, c\, (\overline{n} \parallel z\lfloor n.\,(\overline{c} \parallel \overline{m}) + m.\,([\![P_1]\!] \parallel \overline{m})\rfloor \parallel c.\, z\lfloor [\![P_2]\!] \rfloor))$$

$$[\![a(x_1, x_2).\, Q]\!] = \nu b\,(\overline{a}\langle (y)\overline{b}\langle y\rangle\rangle \parallel b(x_1).\,(x_1 \parallel b(x_2).\,[\![Q]\!]))$$

- The receiver takes the initiative and sends an abstraction with a restricted name *b*
- A private link on *b* is realized upon application
- Communication objects P_1, P_2 are “activated” from the encoding of output

Abstraction passing goes beyond process passing

Theorem. *For every $m, n > 1$, there is no encoding of $SHO^n{}_a$ into SHO^m*

Abstraction passing goes beyond process passing

Theorem. *For every $m, n > 1$, there is no encoding of SHO^n_a into SHO^m*

Proof:

- Suppose there is an encoding A[[-]]: $SHO^n_a \rightarrow SHO^m$
- We know there is an encoding B[[-]]: $SHO^{m+1} \rightarrow SHO^n_a$
- By composability of encodings, we have the encoding
  A◦B[[-]]: $SHO^{m+1} \rightarrow SHO^m$
- However, such an encoding doesn't exist: contradiction

Asynchronous Communication

Abstraction Passing

Polyadic Communication

Asynchronous Communication

Abstraction Passing

Polyadic Communication

Asynchronous Communication

Abstraction Passing

Polyadic Communication

Asynchronous Communication

Abstraction Passing

Polyadic Communication

On the Expressiveness of Polyadic and Synchronous Communication in Higher-Order Process Calculi

Ivan Lanese   Jorge A. Pérez   Davide Sangiorgi   Alan Schmitt

ICALP 2010, Bordeaux.

The notion of encoding (Formally)

The notion of encoding (Formally)

Definition 5 (Syntactic Conditions) *Let* $[\![\cdot]\!] : \mathcal{P}_{\mathsf{s}} \to \mathcal{P}_{\mathsf{t}}$ *be a translation of* $\mathcal{L}_{\mathsf{s}}$ *into* $\mathcal{L}_{\mathsf{t}}$. *We say that* $[\![\cdot]\!]$ *is:*

1. compositional *if for every k-ary operator* op *of* $\mathcal{L}_{\mathsf{s}}$ *and for all* $S_1, \ldots, S_k$ *with* $\mathsf{fn}(S_1, \ldots, S_k) = N$, *there exists a k-ary context* $C^N_{\mathsf{op}} \in \mathcal{P}_{\mathsf{t}}$ *that depends on* N *and* op *such that* $[\![\mathsf{op}(S_1, \ldots, S_k)]\!] = C^N_{\mathsf{op}}[[\![S_1]\!], \ldots, [\![S_k]\!]]$;
2. name invariant *if* $[\![\sigma(P)]\!] = \sigma([\![P]\!])$, *for any injective renaming of names* σ.

The notion of encoding (Formally)

Definition 5 (Syntactic Conditions) *Let* $[\![\cdot]\!] : \mathcal{P}_\mathsf{s} \to \mathcal{P}_\mathsf{t}$ *be a translation of* $\mathcal{L}_\mathsf{s}$ *into* $\mathcal{L}_\mathsf{t}$. *We say that* $[\![\cdot]\!]$ *is:*

1. compositional *if for every k-ary operator* op *of* $\mathcal{L}_\mathsf{s}$ *and for all* $S_1, \ldots, S_k$ *with* $\mathsf{fn}(S_1, \ldots, S_k) = N$, *there exists a k-ary context* $C^N_{\mathsf{op}} \in \mathcal{P}_\mathsf{t}$ *that depends on* N *and* op *such that* $[\![\mathsf{op}(S_1, \ldots, S_k)]\!] = C^N_{\mathsf{op}}[[\![S_1]\!], \ldots, [\![S_k]\!]]$;
2. name invariant *if* $[\![\sigma(P)]\!] = \sigma([\![P]\!])$, *for any injective renaming of names* σ.

Definition 6 (Semantic Conditions) *Let* $[\![\cdot]\!] : \mathcal{P}_\mathsf{s} \to \mathcal{P}_\mathsf{t}$ *be a translation of* $\mathcal{L}_\mathsf{s}$ *into* $\mathcal{L}_\mathsf{t}$. *We say that* $[\![\cdot]\!]$ *is:*

1. complete *if for every* $S, S' \in \mathcal{P}_\mathsf{s}$ *and* $\alpha \in \mathcal{A}_\mathsf{s}$ *such that* $S \stackrel{\alpha}{\Longrightarrow}_\mathsf{s} S'$, *it holds that* $[\![S]\!] \stackrel{\beta}{\Longrightarrow}_\mathsf{t} \approx_\mathsf{t} [\![S']\!]$, *where* $\beta \in \mathcal{A}_\mathsf{t}$ *and* $\mathsf{sig}(\alpha) = \mathsf{sig}(\beta)$;
2. sound *if for every* $S \in \mathcal{P}_\mathsf{s}$, $T \in \mathcal{P}_\mathsf{t}$, $\beta \in \mathcal{A}_\mathsf{t}$ *such that* $[\![S]\!] \stackrel{\beta}{\Longrightarrow}_\mathsf{t} T$ *there exists an* $S' \in \mathcal{P}_\mathsf{s}$ *and an* $\alpha \in \mathcal{A}_\mathsf{s}$ *such that* $S \stackrel{\alpha}{\Longrightarrow}_\mathsf{s} S'$, $T \Rightarrow\approx_\mathsf{t} [\![S']\!]$, *and* $\mathsf{sig}(\alpha) = \mathsf{sig}(\beta)$;
3. adequate *if for every* $S, S' \in \mathcal{P}_\mathsf{s}$, *if* $S \approx_\mathsf{s} S'$ *then* $[\![S]\!] \approx_\mathsf{t} [\![S']\!]$;
4. diverge-reflecting *if for every* $S \in \mathcal{P}_\mathsf{s}$, $[\![S]\!]$ *diverges only if* S *diverges.*

Example: Sync into Async

A synchronous "duplicator" process:

$$\overline{a}\langle P\rangle.S \parallel a(x).(x \parallel x) \xrightarrow{a\tau} S \parallel P \parallel P$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \parallel a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \parallel a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \ & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \ \parallel \ a(x).(x \parallel [\![x \parallel x]\!]) \\
= \ & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \ \parallel \ a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \ & \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \ ) \\
\xrightarrow{\tau} \ & \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \ ) \\
\xrightarrow{\tau} \ & \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \ ) \\
\xrightarrow{\tau} \ & \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \ & [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \; & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \; & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \; & \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \; & [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& \llbracket \overline{a}\langle P\rangle.S \rrbracket \parallel \llbracket a(x).(x \parallel x) \rrbracket \\
= \;& \nu k\, l\, (\overline{a}\langle k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel \llbracket x \parallel x \rrbracket) \\
= \;& \nu k\, l\, (\overline{a}\langle k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\overline{l} \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k}) \\
& \qquad\quad \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k}) \\
& \qquad\quad \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, (\llbracket S\rrbracket \parallel \overline{k} \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k}) \\
& \qquad\qquad\qquad\qquad \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, l\, (\llbracket S\rrbracket \parallel \llbracket P\rrbracket \parallel \overline{k} \parallel k.(\llbracket P\rrbracket \parallel \overline{k}) + l.(\llbracket S\rrbracket \parallel \overline{k})\;) \\
\xrightarrow{\tau} \;& \nu k\, (\llbracket S\rrbracket \parallel \llbracket P\rrbracket \parallel \llbracket P\rrbracket \parallel \overline{k}) \\
\approx \;& \llbracket S\rrbracket \parallel \llbracket P\rrbracket \parallel \llbracket P\rrbracket
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
=\ & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \ \parallel\ a(x).(x \parallel [\![x \parallel x]\!]) \\
=\ & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \ \parallel\ a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau}\ & \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau}\ & \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau}\ & \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau}\ & \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx\ & [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \; & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \; & \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \overline{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \; & \nu k\, l\, (\overline{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad\quad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \; & \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \; & [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad\qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \;) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$

$$
\begin{aligned}
& [\![\overline{a}\langle P\rangle.S]\!] \parallel [\![a(x).(x \parallel x)]\!] \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel [\![x \parallel x]\!]) \\
= \;& \nu k\, l\, (\overline{a}\langle k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\rangle \parallel \bar{l}) \;\parallel\; a(x).(x \parallel x \parallel x) \\
\xrightarrow{a\tau} \;& \nu k\, l\, (\bar{l} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k}) \\
& \qquad\qquad \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, l\, ([\![S]\!] \parallel [\![P]\!] \parallel \overline{k} \parallel k.([\![P]\!] \parallel \overline{k}) + l.([\![S]\!] \parallel \overline{k})\ ) \\
\xrightarrow{\tau} \;& \nu k\, ([\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!] \parallel \overline{k}) \\
\approx \;& [\![S]\!] \parallel [\![P]\!] \parallel [\![P]\!]
\end{aligned}
$$