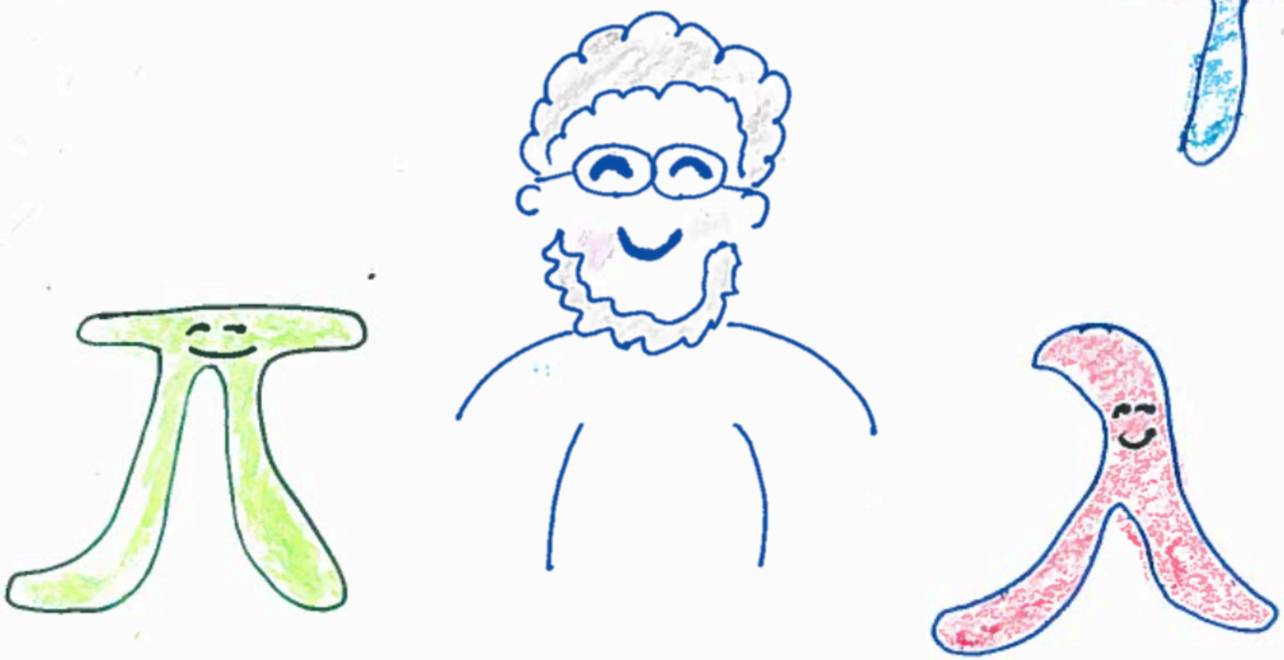
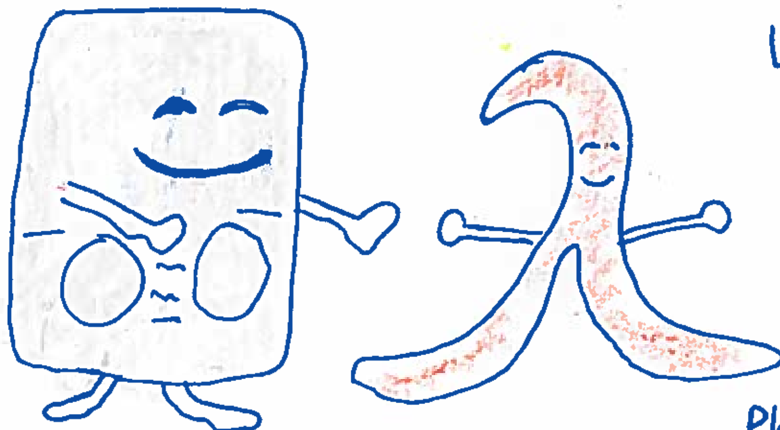
PHIL s SESSION TYPES



Imperial College London

199?

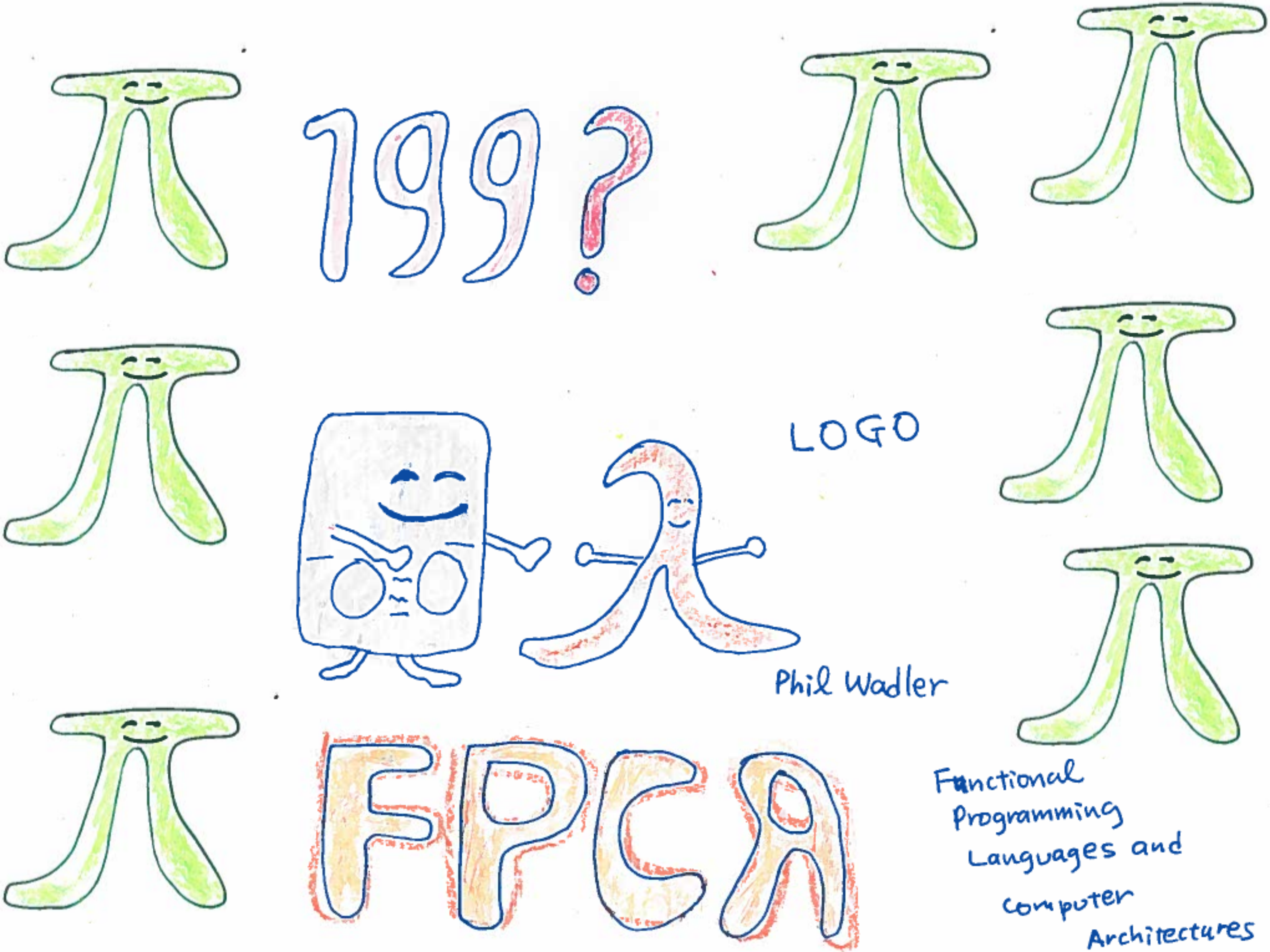


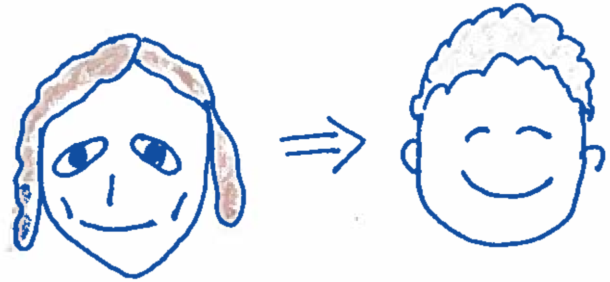
LOGO

Phil Wadler

FPCA

Functional
Programming
Languages and
Computer
Architectures

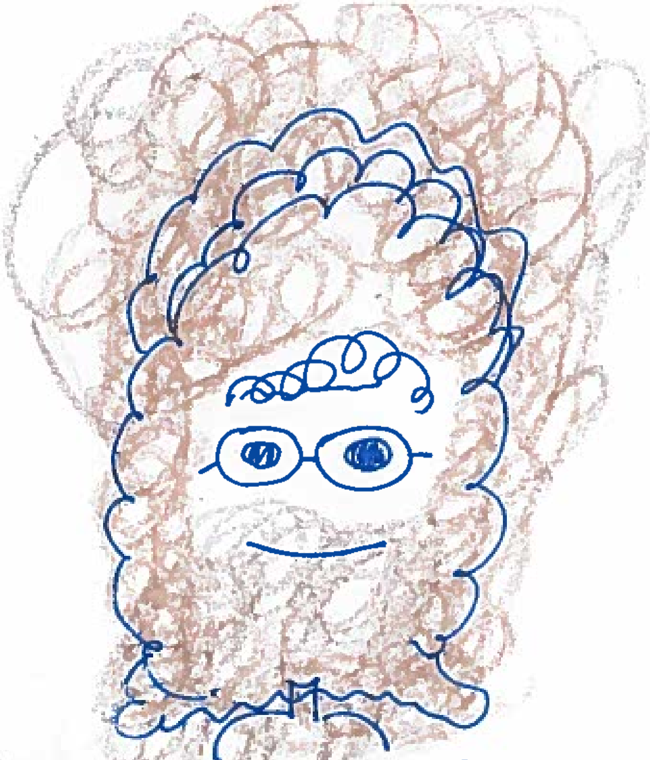

199?
LOGO
Phil Wadler
FPCA
Functional Programming Languages and Computer Architectures



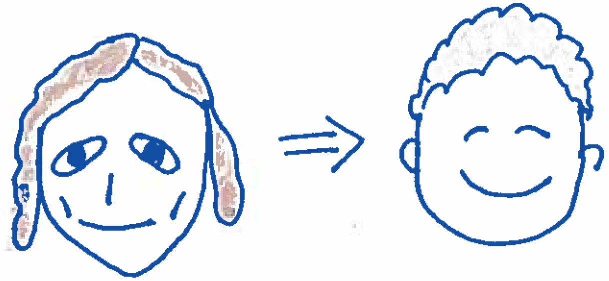
Jean-Jacque Levy

Andy Gordon

Luc Maranget



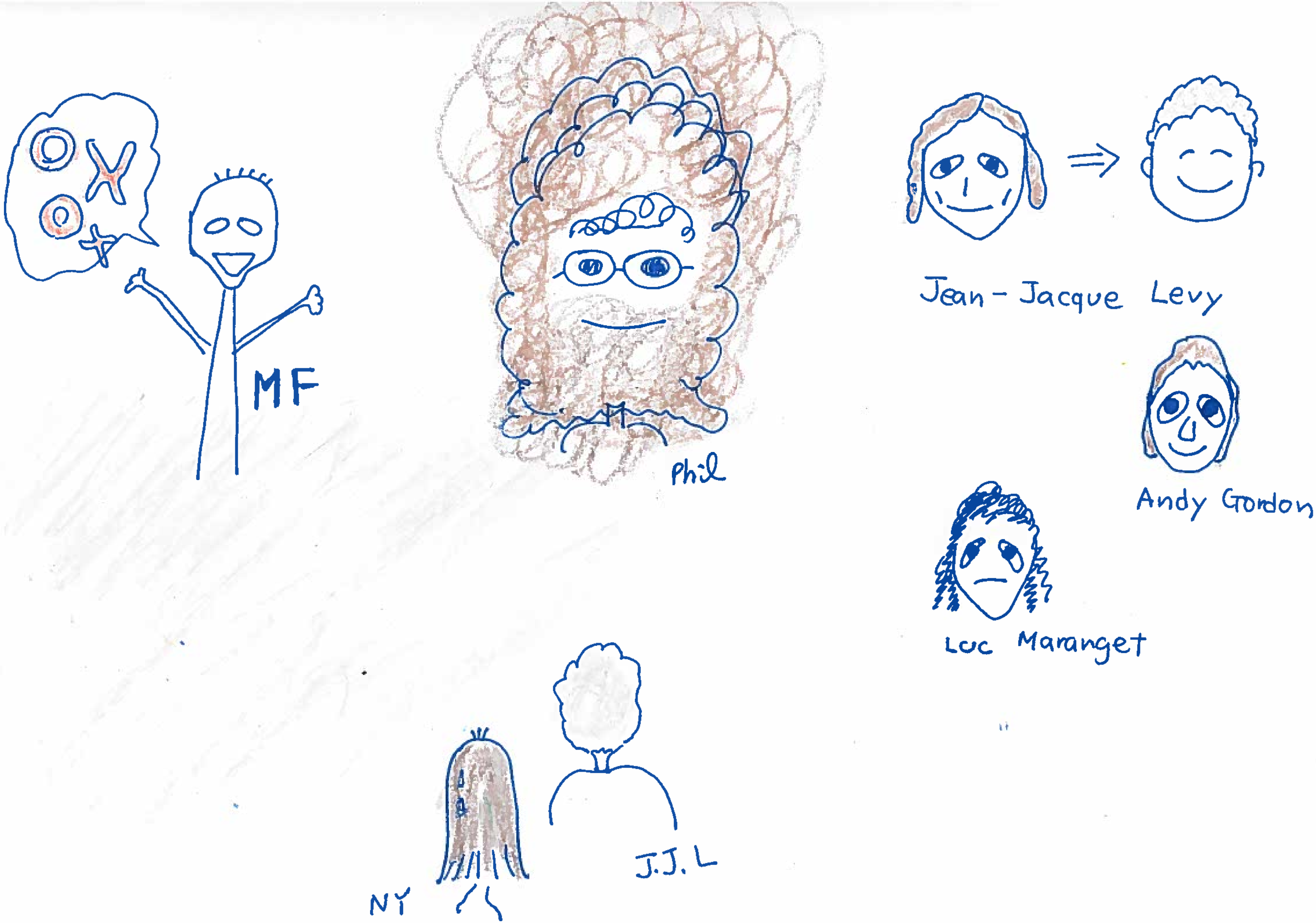
Phil



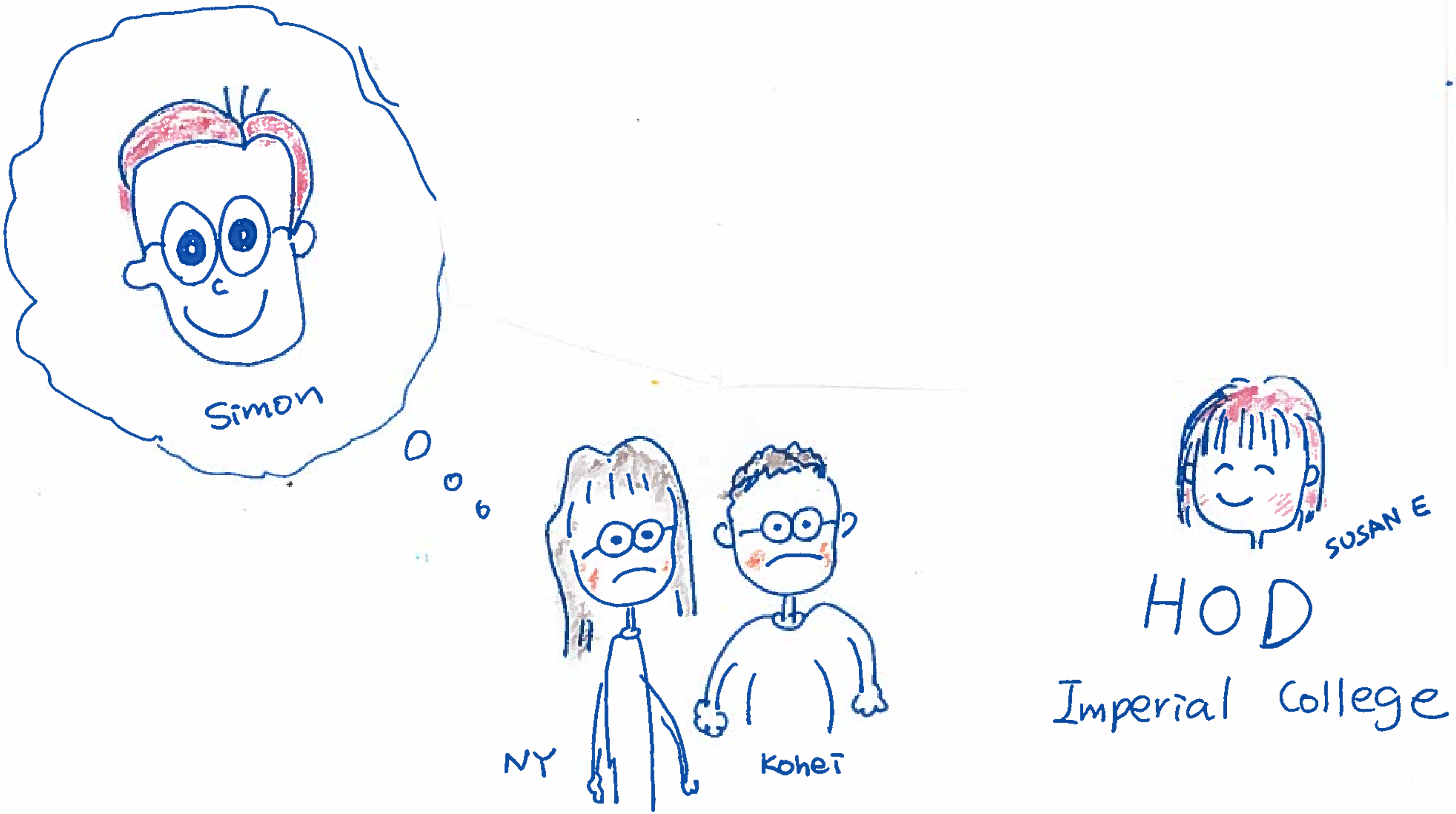
Jean-Jacque Levy

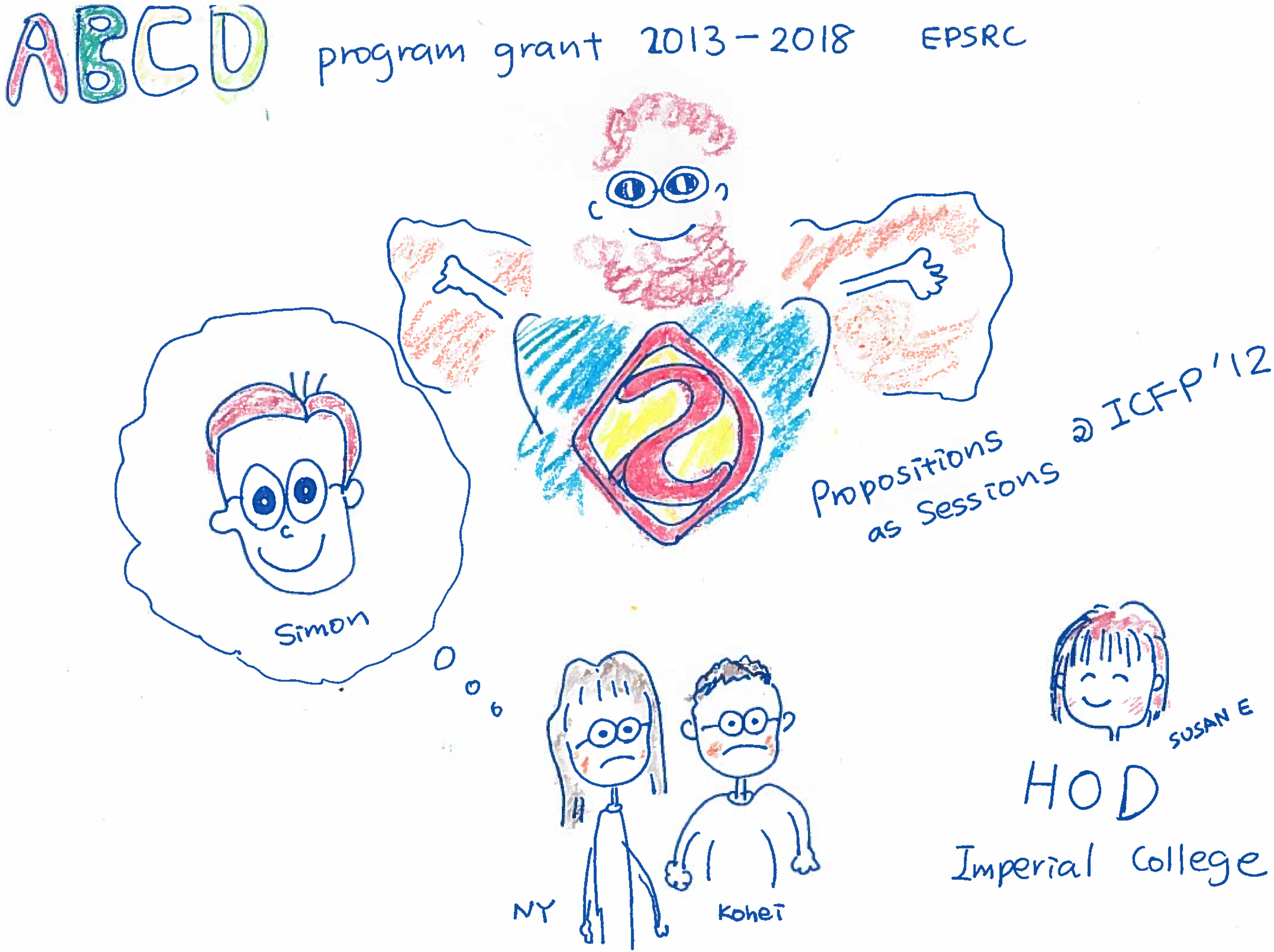
Andy Gordon

Luc Maranget


MF
Phil
Jean-Jacque Levy
Andy Gordon
Luc Maranget
NY
J.J.L

ABCD




ABCD program grant 2013-2018 EPSRC
Propositions as Sessions @ICFP'12
Simon
NY
Kohei
SUSAN E
HOD
Imperial College

Session Types and Linear Logic

WadlerFest

Bernardo Toninho, Nobuko Yoshida

11 April 2016

Session Types and Linear Logic

Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus

Session Types and Linear Logic
Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus
- Structure channel-based comm. with the notion of a session.
  - Sequence of interactive behaviours between two agents.

Session Types and Linear Logic

Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus
- Structure channel-based comm. with the notion of a session.
  - Sequence of interactive behaviours between two agents.
  - Intrinsic notion of duality (send/receive, branch/select).

Session Types and Linear Logic

Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus
- Structure channel-based comm. with the notion of a session.
  - Sequence of interactive behaviours between two agents.
  - Intrinsic notion of duality (send/receive, branch/select).
- Ensure communication safety (liveness, fidelity, ...)

Session Types and Linear Logic

Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus
- Structure channel-based comm. with the notion of a session.
  - Sequence of interactive behaviours between two agents.
  - Intrinsic notion of duality (send/receive, branch/select).
- Ensure communication safety (liveness, fidelity, . . .)

Linear Logic [Girard98]:

- A substructural logic of resources.

Session Types and Linear Logic

Context

Session Types [Honda et al93]:

- Typing discipline for π-calculus
- Structure channel-based comm. with the notion of a session.
  - Sequence of interactive behaviours between two agents.
  - Intrinsic notion of duality (send/receive, branch/select).
- Ensure communication safety (liveness, fidelity, . . .)

Linear Logic [Girard98]:

- A substructural logic of resources.
- Marriage of the dualities of classical logic with constructive aspects of intuitionistic logic.
- Far reaching applications in CS (linear λ-calculus, implicit comp. complexity, linear types, etc.)

Session Types and Linear Logic

A bit of history

Propositions as Types:

- A (deep) connection between prop. logic and λ-calculus.

Session Types and Linear Logic

A bit of history

Propositions as Types:

- A (deep) connection between prop. logic and λ-calculus.
- Proofs as programs, computation as proof simplification.

Session Types and Linear Logic

A bit of history

Propositions as Types:

- A (deep) connection between prop. logic and λ-calculus.
- Proofs as programs, computation as proof simplification.
- Not just prop. logic...

Session Types and Linear Logic

A bit of history

Propositions as Types:

- A (deep) connection between prop. logic and λ-calculus.
- Proofs as programs, computation as proof simplification.
- Not just prop. logic. . .

Concurrency Theory:

- Process Algebra (CSP [Hoare78], CCS, π-calculus [Milner80,89])
- Language-based models of message-passing concurrency.
- A plethora of typing systems (I/O types, Usage types, Linear types, . . . , Session types)

Session Types and Linear Logic
A bit of history

Linear Logic and Concurrency:

- A logic of interacting resources?

Session Types and Linear Logic

A bit of history

Linear Logic and Concurrency:

- A logic of interacting resources?
- Initial efforts explored connections to concurrency:
  - Abramsky's computational interpretation [Abramsky93]
  - Bellin and Scott's refinement to a π-calculus [BellinScott94]
  - Specification structures / Interaction cat. [Abramsky et al.95]

Session Types and Linear Logic

A bit of history

Linear Logic and Concurrency:

- A logic of interacting resources?
- Initial efforts explored connections to concurrency:
  - Abramsky's computational interpretation [Abramsky93]
  - Bellin and Scott's refinement to a π-calculus [BellinScott94]
  - Specification structures / Interaction cat. [Abramsky et al.95]
- No real "Curry-Howard isomorphism"...

Session Types and Linear Logic

A bit of history

Linear Logic and Concurrency:

- A logic of interacting resources?
- Initial efforts explored connections to concurrency:
  - Abramsky's computational interpretation [Abramsky93]
  - Bellin and Scott's refinement to a π-calculus [BellinScott94]
  - Specification structures / Interaction cat. [Abramsky et al.95]
- No real "Curry-Howard isomorphism"...

Why does it matter?

- New means of reasoning about concurrent phenomena.
- Good metalogical properties map to good program properties.
- ...

Session Types and Linear Logic

What is old is new again

ILL and Session Types – SILL [CairesPfenning10]

- Interpret Session Types as ILL propositions.
- Proofs as typing derivations for π-calculus.
- Process reduction as cut reduction/elimination.

Session Types and Linear Logic

What is old is new again

ILL and Session Types – SILL [CairesPfenning10]

- Interpret Session Types as ILL propositions.
- Proofs as typing derivations for π-calculus.
- Process reduction as cut reduction/elimination.

CLL and Session Types – CP [Wadler12,14]

- Full linear logic.
- Further from π-calculus, but matching LL precisely.
- Embeds a session-typed functional language (GV) into CP.

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$A, B \quad ::=$$

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$
\begin{array}{rcll}
A, B & ::= & A \otimes B & \text{output } A \text{ then behave as } B \\
 & & A \parr B & \text{Input } A \text{ then behave as } B
\end{array}
$$

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$
\begin{array}{rcll}
A, B & ::= & A \otimes B & \text{output } A \text{ then behave as } B \\
 & & A \mathbin{\wp} B & \text{Input } A \text{ then behave as } B \\
 & & A \oplus B & \text{Select from } A \text{ or } B \\
 & & A \mathbin{\&} B & \text{Offer choice of } A \text{ or } B
\end{array}
$$

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$
\begin{array}{lcll}
A, B & ::= & A \otimes B & \text{output } A \text{ then behave as } B \\
 & & A \parr B & \text{Input } A \text{ then behave as } B \\
 & & A \oplus B & \text{Select from } A \text{ or } B \\
 & & A \,\&\, B & \text{Offer choice of } A \text{ or } B \\
 & & !A & \text{Server Accept} \\
 & & ?A & \text{Client Request}
\end{array}
$$

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

A, B	$::=$	$A \otimes B$	output A then behave as B
		$A \parr B$	Input A then behave as B
		$A \oplus B$	Select from A or B
		$A \mathbin{\&} B$	Offer choice of A or B
		$!A$	Server Accept
		$?A$	Client Request
		$\mathbf{1}$	Unit for $\otimes$
		$\ldots$	

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$
\begin{array}{llll}
A, B & ::= & A \otimes B & \text{output } A \text{ then behave as } B \\
 & & A \mathbin{⅋} B & \text{Input } A \text{ then behave as } B \\
 & & A \oplus B & \text{Select from } A \text{ or } B \\
 & & A \mathbin{\&} B & \text{Offer choice of } A \text{ or } B \\
 & & !A & \text{Server Accept} \\
 & & ?A & \text{Client Request} \\
 & & \mathbf{1} & \text{Unit for } \otimes \\
 & & \ldots &
\end{array}
$$

Cut as Composition

$$
(\text{cut})\frac{P \vdash \Delta, x{:}A \qquad Q \vdash \Delta', x{:}A^{\perp}}{\nu x.(P \mid Q) \vdash \Delta, \Delta'}
$$

Session Types and Linear Logic

Meanings of Propositions

CLL Propositions as Sessions

$$
\begin{array}{lcll}
A, B & ::= & A \otimes B & \text{output } A \text{ then behave as } B \\
 & & A \parr B & \text{Input } A \text{ then behave as } B \\
 & & A \oplus B & \text{Select from } A \text{ or } B \\
 & & A \,\&\, B & \text{Offer choice of } A \text{ or } B \\
 & & !A & \text{Server Accept} \\
 & & ?A & \text{Client Request} \\
 & & \mathbf{1} & \text{Unit for } \otimes \\
 & & \ldots &
\end{array}
$$

Identity as Forwarding

$$
(\mathrm{id})\frac{}{x \leftrightarrow y \vdash x{:}A, y{:}A^{\perp}}
$$

Input and Output:

$$(\otimes)\frac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B}$$

Session Types and Linear Logic

Reductions

Input and Output:

$$(\otimes)\frac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B} \quad (\parr)\frac{P \vdash \Delta_3, y{:}A, x{:}B}{x(y).P \vdash \Delta_3, x{:}A \parr B}$$

Session Types and Linear Logic

Reductions

Input and Output:

$$(\otimes)\frac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B} \quad (\parr)\frac{P \vdash \Delta_3, y{:}A, x{:}B}{x(y).P \vdash \Delta_3, x{:}A \parr B}$$

Communication as principal cut reductions:

$$\frac{\dfrac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B} \quad \dfrac{R \vdash y{:}A^\perp, x{:}B^\perp}{x(y).R \vdash \Delta_3, x{:}A^\perp \parr B^\perp}}{\boldsymbol{\nu} x.(x[y].(P \mid Q) \mid x(y).R) \vdash \Delta_1, \Delta_2, \Delta_3}$$

Session Types and Linear Logic

Reductions

Input and Output:

$$(\otimes)\frac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B} \qquad (\parr)\frac{P \vdash \Delta_3, y{:}A, x{:}B}{x(y).P \vdash \Delta_3, x{:}A \parr B}$$

Communication as principal cut reductions:

$$\frac{\dfrac{P \vdash \Delta_1, y{:}A \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B} \qquad \dfrac{R \vdash y{:}A^\perp, x{:}B^\perp}{x(y).R \vdash \Delta_3, x{:}A^\perp \parr B^\perp}}{\boldsymbol{\nu} x.(x[y].(P \mid Q) \mid x(y).R) \vdash \Delta_1, \Delta_2, \Delta_3}$$

$$\Longrightarrow \frac{P \vdash \Delta_1, y{:}A \qquad \dfrac{Q \vdash \Delta_2, x{:}B \quad R \vdash \Delta_3, y{:}A^\perp, x{:}B^\perp}{\boldsymbol{\nu} x.(Q \mid R) \vdash \Delta_2, \Delta_3, y{:}A^\perp}}{\boldsymbol{\nu} y.(P \mid \boldsymbol{\nu} x.(Q \mid R)) \vdash \Delta_1, \Delta_2, \Delta_3}$$

Session Types and Linear Logic

What about the other proof conversions?

$$
\text{(cut)}\dfrac{(\otimes)\dfrac{P \vdash \Delta_1, y{:}A, z{:}C \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B, z{:}C} \quad R \vdash \Delta_3, z{:}C^{\perp}}{\nu z.(x[y].(P \mid Q) \mid R) \vdash \Delta_1, x{:}A \otimes B, \Delta_2, \Delta_3}
$$

Session Types and Linear Logic

What about the other proof conversions?

$$
\text{(cut)}\dfrac{(\otimes)\dfrac{P \vdash \Delta_1, y{:}A, z{:}C \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B, z{:}C} \quad R \vdash \Delta_3, z{:}C^{\perp}}{\boldsymbol{\nu} z.(x[y].(P \mid Q) \mid R) \vdash \Delta_1, x{:}A \otimes B, \Delta_2, \Delta_3}
$$

$$
\Longrightarrow (\otimes)\dfrac{\text{(cut)}\dfrac{P \vdash \Delta_1, y{:}A, z{:}C \quad R \vdash \Delta_3, z{:}C^{\perp}}{\boldsymbol{\nu} z.(P \mid R) \vdash \Delta_1, \Delta_3, y{:}A} \quad Q \vdash \Delta_2, x{:}B}{x[y].(\boldsymbol{\nu} z.(P \mid R) \mid Q) \vdash \Delta_1, x{:}A \otimes B, \Delta_2, \Delta_3}
$$

Session Types and Linear Logic

What about the other proof conversions?

$$
(\text{cut})\;\dfrac{(\otimes)\;\dfrac{P \vdash \Delta_1, y{:}A, z{:}C \quad Q \vdash \Delta_2, x{:}B}{x[y].(P \mid Q) \vdash \Delta_1, \Delta_2, x{:}A \otimes B, z{:}C} \quad R \vdash \Delta_3, z{:}C^{\perp}}{\boldsymbol{\nu} z.(x[y].(P \mid Q) \mid R) \vdash \Delta_1, x{:}A \otimes B, \Delta_2, \Delta_3}
$$

$$
\Longrightarrow (\otimes)\;\dfrac{(\text{cut})\;\dfrac{P \vdash \Delta_1, y{:}A, z{:}C \quad R \vdash \Delta_3, z{:}C^{\perp}}{\boldsymbol{\nu} z.(P \mid R) \vdash \Delta_1, \Delta_3, y{:}A} \quad Q \vdash \Delta_2, x{:}B}{x[y].(\boldsymbol{\nu} z.(P \mid R) \mid Q) \vdash \Delta_1, x{:}A \otimes B, \Delta_2, \Delta_3}
$$

$$
\boldsymbol{\nu} z.(x[y].(P \mid Q) \mid R) \Longrightarrow x[y].(\boldsymbol{\nu} z.(P \mid R) \mid Q) \quad \text{if } z \in fn(P)
$$

Session Types and Linear Logic

Putting it all together

CP Reduction

- One CP reduction for every principal cut reduction (one per dual prop. pair).
- One CP reduction for each commutting conversion (2 for $\otimes$, 2 for $\oplus$, none for $\mathbf{0}$, one for the rest).

Session Types and Linear Logic

Putting it all together

CP Reduction

- One CP reduction for every principal cut reduction (one per dual prop. pair).
- One CP reduction for each commutting conversion (2 for $\otimes$, 2 for $\oplus$, none for 0, one for the rest).

Metatheorems

- If $P \vdash \Delta$ and $P \Longrightarrow Q$ then $Q \vdash \Delta$.
- If $P \vdash \Delta$ there exists Q such that $P \Longrightarrow^* Q$ and Q is not a cut.

Session Types and Linear Logic

Why does it matter?

- Safety/liveness properties “for free”.
- A solid foundation to build on:
  - Encodings of λ-calculi into π-calculus / CP [Toninho et al.12,Wadler12,LindleyMorris15].
  - Value-dependent / refinement session types [Toninho11].
  - Multiparty sessions [Carbone et al.,15].
  - Dynamic monitoring [Jia et al.16]
  - “Better” designed languages [Wadler12,Toninho et al13]
  - ...

Session Types and Linear Logic

Why does it still matter?

Loads still to do (fortunately)!

- Curry-Howard iso. gave us Haskell, ML, ...
- Need a "real" language that puts this all together!

Session Types and Linear Logic

Why does it still matter?

Loads still to do (fortunately)!

- Curry-Howard iso. gave us Haskell, ML, ...
- Need a “real” language that puts this all together!
- A better approach to logic and multiparty sessions.

Session Types and Linear Logic

Why does it still matter?

Loads still to do (fortunately)!

- Curry-Howard iso. gave us Haskell, ML, ...
- Need a "real" language that puts this all together!
- A better approach to logic and multiparty sessions.
- True non-determinism?
- More π-calculus-like behaviours?

Session Types and Linear Logic

Why does it still matter?

Loads still to do (fortunately)!

- Curry-Howard iso. gave us Haskell, ML, ...
- Need a "real" language that puts this all together!
- A better approach to logic and multiparty sessions.
- True non-determinism?
- More π-calculus-like behaviours?
- Dependent types?
- etc...

Session Types and Linear Logic

Conclusion

- The connections of linear logic and session types:
  - Linear propositions as session types.
  - Proofs as processes.
  - Communication and proof conversion.

Session Types and Linear Logic

Conclusion

- The connections of linear logic and session types:
  - Linear propositions as session types.
  - Proofs as processes.
  - Communication and proof conversion.
- Logic gives us comm. safety / liveness for free.

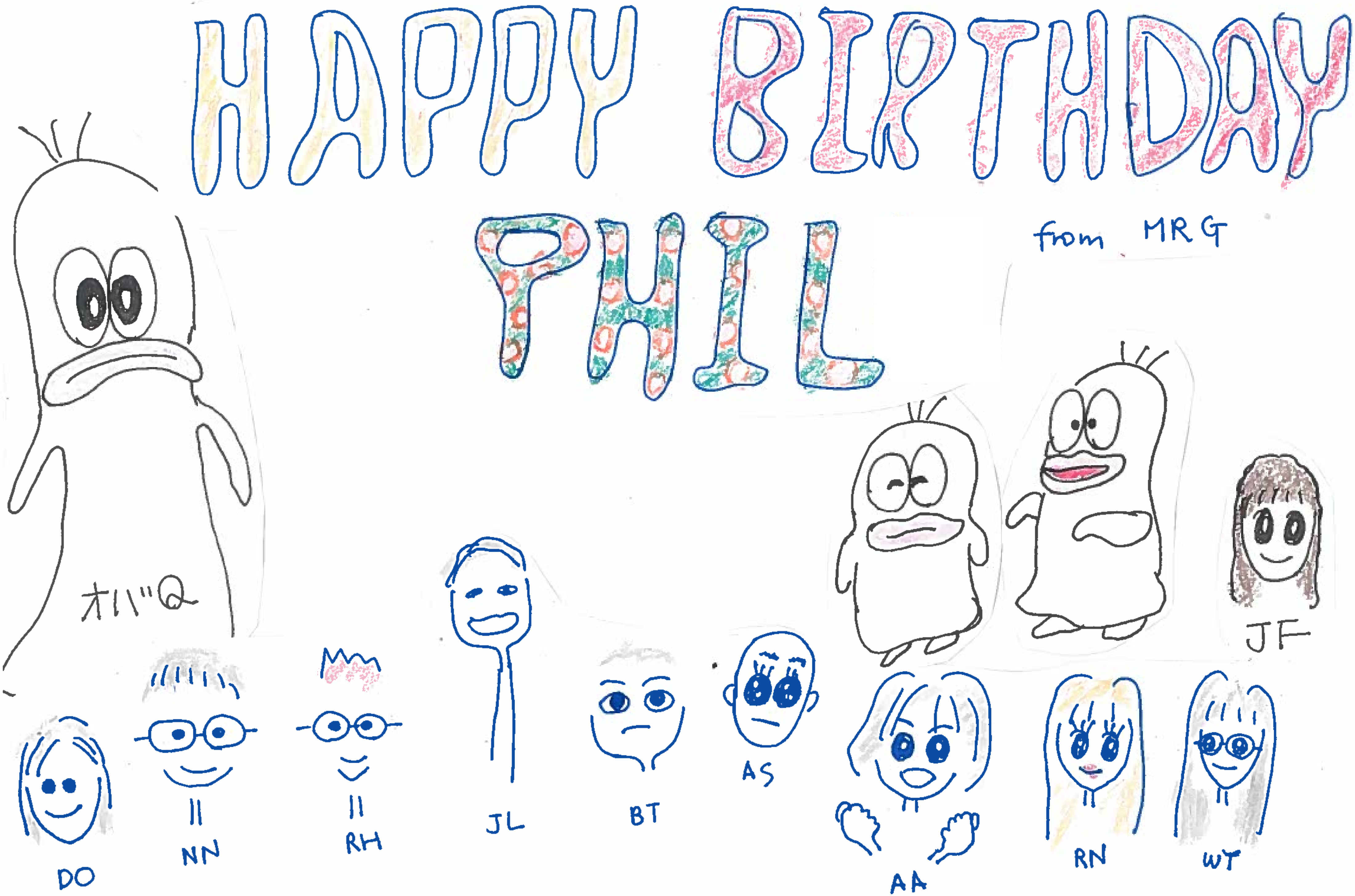
Session Types and Linear Logic

Conclusion

- The connections of linear logic and session types:
  - Linear propositions as session types.
  - Proofs as processes.
  - Communication and proof conversion.
- Logic gives us comm. safety / liveness for free.
- . . . but also a general and powerful framework to reason about concurrency!

Session Types and Linear Logic
Conclusion

- The connections of linear logic and session types:
  - Linear propositions as session types.
  - Proofs as processes.
  - Communication and proof conversion.
- Logic gives us comm. safety / liveness for free.
- ... but also a general and powerful framework to reason about concurrency!
- Only scratched the surface!


HAPPY BIRTHDAY
PHIL
from MRG
オバQ
JF
DO
NN
RH
JL
BT
AS
AA
RN
WT