Propositions as Sessions

Austeja Elvina Brasiunaite
Sam Lindley
J. Garrett Morris
Philip Wadler
University of Edinburgh

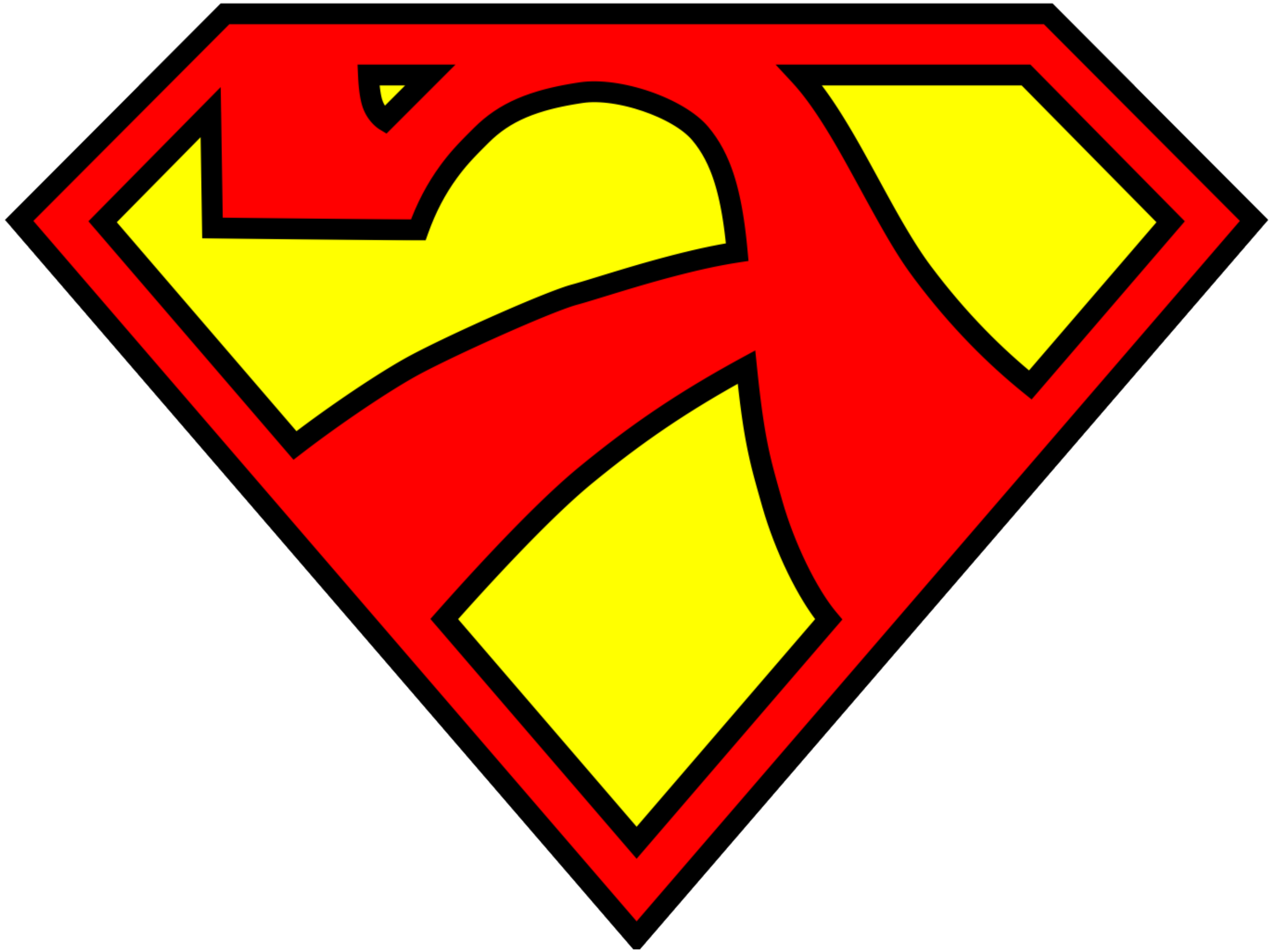
AIPL, Heriot Watt
Tuesday 19 August 2014

Kohei Honda, 1959–2012

EPSRC Programme Grant EP/K034413/1

From Data Types to Session Types: A Basis for Concurrency and Distribution (ABCD)

Simon Gay, Nobuko Yoshida, Philip Wadler



Propositions as Types

propositions	*as*	types
proofs	*as*	programs
normalisation of proofs	*as*	evaluation of programs

Propositions as Types is robust

propositions	*as*	types
proofs	*as*	programs
normalisation of proofs	*as*	evaluation of programs

Intuitionistic Natural Deduction	$\leftrightarrow$	Simply-Typed Lambda Calculus
Quantification over propositions	$\leftrightarrow$	Polymorphism
Quantification over individuals	$\leftrightarrow$	Dependent types
Modal Logic	$\leftrightarrow$	Monads (state, exceptions)
Classical-Intuitionistic Embedding	$\leftrightarrow$	Continuation Passing Style

. . . but there's a missing link

propositions	*as*	types
proofs	*as*	programs
normalisation of proofs	*as*	evaluation of programs

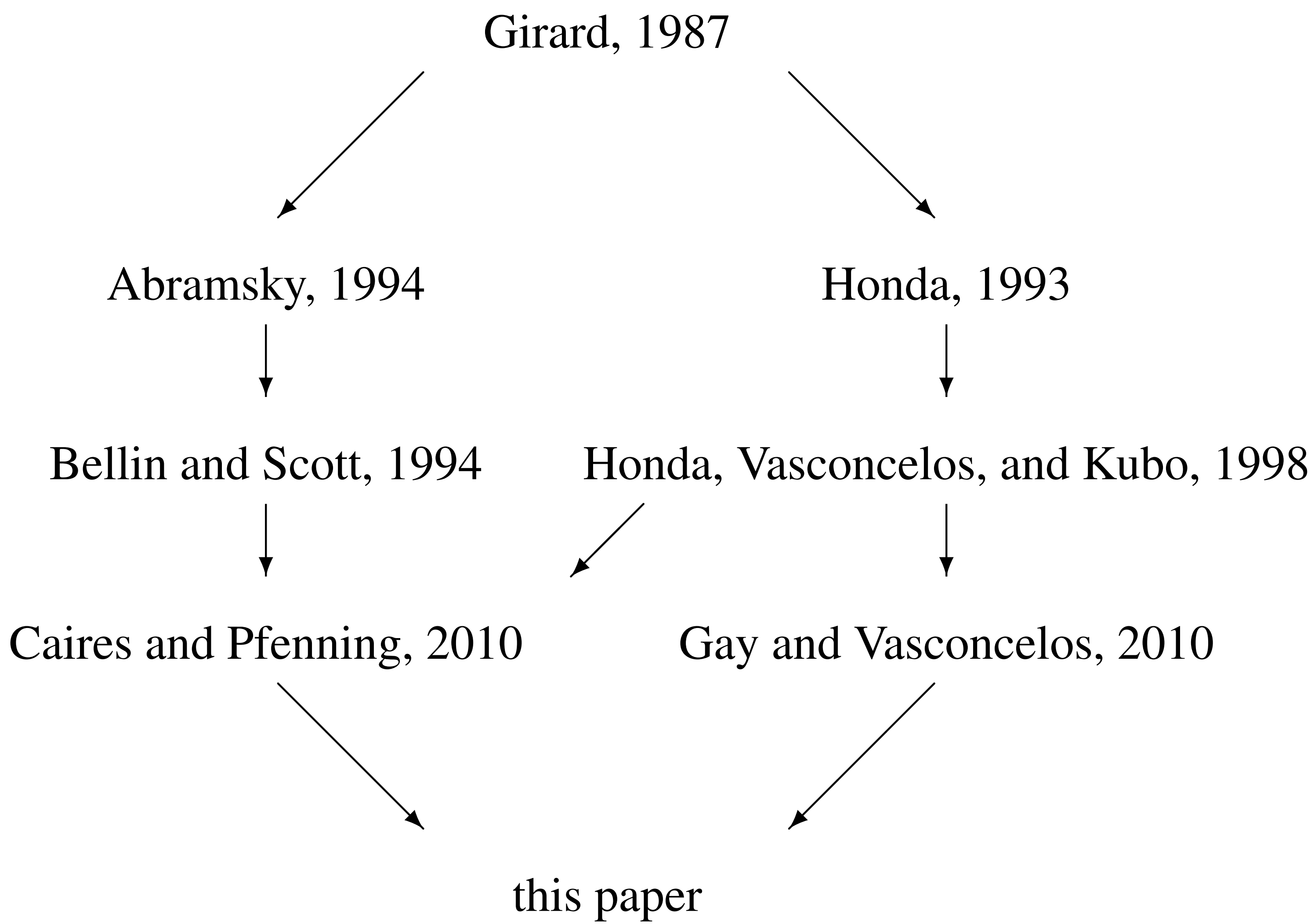
Intuitionistic Natural Deduction	$\leftrightarrow$	Simply-Typed Lambda Calculus
Quantification over propositions	$\leftrightarrow$	Polymorphism
Quantification over individuals	$\leftrightarrow$	Dependent types
Modal Logic	$\leftrightarrow$	Monads (state, exceptions)
Classical-Intuitionistic Embedding	$\leftrightarrow$	Continuation Passing Style
???	$\leftrightarrow$	Process Calculus

Propositions as Sessions

propositions	*as*	types
proofs	*as*	programs
normalisation of proofs	*as*	evaluation of programs

propositions	*as*	session types
proofs	*as*	processes
cut elimination	*as*	communication

Lines of development



ILL vs. CLL

- Caires and Pfenning, 2010: Intuitionistic Linear Logic

$$\frac{\Gamma;\ \Delta \vdash P :: y : A \qquad \Gamma;\ \Delta' \vdash Q :: x : B}{\Gamma;\ \Delta, \Delta' \vdash \nu y.\, x\langle y\rangle.(P \mid Q) :: x : A \otimes B}\ \otimes\text{-R}$$

$$\frac{\Gamma;\ \Delta \vdash P :: y : A \quad \Gamma;\ \Delta', x : B \vdash Q :: z : C}{\Gamma;\ \Delta, \Delta', x : A \multimap B \vdash \nu y.x\langle y\rangle.(P \mid Q) :: z : C}\ \multimap\text{-L}$$

$$\frac{\Gamma;\ \Delta, y : A \vdash R :: x : B}{\Gamma;\ \Delta \vdash x(y).R :: x : A \multimap B}\ \multimap\text{-R} \qquad \frac{\Gamma;\ \Delta, y : A, x : B \vdash R :: z : C}{\Gamma;\ \Delta, x : A \otimes B \vdash x(y).R :: z : C}\ \otimes\text{-L}$$

- this paper: Classical Linear Logic

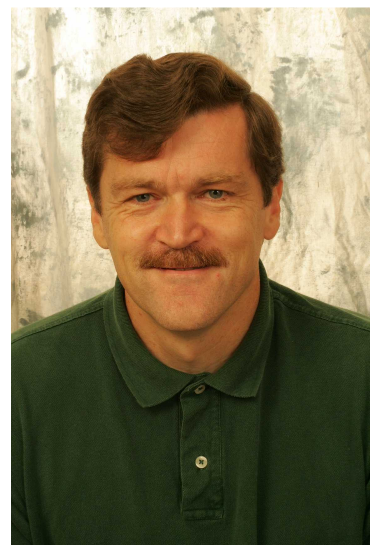
$$\frac{P \vdash \Gamma, y : A \qquad Q \vdash \Delta, x : B}{x[y].(P \mid Q) \vdash \Gamma, \Delta, x : A \otimes B}\ \otimes \qquad \frac{R \vdash \Theta, y : A, x : B}{x(y).R \vdash \Theta, x : A \parr B}\ \parr$$

Part I

CP

Classical Processes

Caires-Pfenning



Part II

GV

Good Variation

Gay-Vasconcelos

Part III

Demo

Session Types

$S ::=$

$!T.S$	output value of type T then behave as S
$?T.S$	input value of type T then behave as S
$\oplus\{l_i : S_i\}_{i \in I}$	select from behaviours S_i with label l_i
$\&\{l_i : S_i\}_{i \in I}$	offer choice of behaviours S_i with label l_i
$\mathsf{end}_!$	terminator, convenient for use with output
$\mathsf{end}_?$	terminator, convenient for use with input

Each session S has a dual $\overline{S}$:

$$\begin{aligned} \overline{!T.S} &= ?T.\overline{S} & \overline{?T.S.} &= !T.\overline{S} \\ \overline{\oplus(l_i : S_i)_{i \in I}} &= \&(l_i : \overline{S}_i)_{i \in I} & \overline{\&(l_i : S_i)_{i \in I}} &= \oplus(l_i : \overline{S}_i)_{i \in I} \\ \overline{\mathsf{end}_!} &= \mathsf{end}_? & \overline{\mathsf{end}_?} &= \mathsf{end}_! \end{aligned}$$

Types

$$T, U, V ::=$$

S	session (linear)
$T \otimes U$	tensor product (linear)
$T \multimap U$	function (linear)
$T \to U$	function (unlimited)
Unit	unit (unlimited)

Each type is classified as linear or unlimited:

$$\mathsf{lin}(S) \quad \mathsf{lin}(T \otimes U) \quad \mathsf{lin}(T \multimap U)$$

$$\mathsf{un}(T \to U) \quad \mathsf{un}(\mathsf{Unit})$$

Part IV

Conclusions