Using session types as an effect system

"Effects in a pi"

Dominic Orchard, Nobuko Yoshida

Imperial College London

dorchard.co.uk

PLACES 2015 - April 18th 2015

<u>Effect systems</u> describe side-effect *behaviour*

λ-calculus as prototype

$$(\text{app})\ \frac{\Gamma \vdash M : \sigma \xrightarrow{\mathsf{H}} \tau,\ \mathsf{F} \qquad \Gamma \vdash N : \sigma,\ \mathsf{G}}{\Gamma \vdash M\ N : \tau,\ \mathsf{F} \bullet \mathsf{G} \bullet \mathsf{H}}$$

e.g. $\Gamma \vdash r2 := r1 + 1 : \mathbf{unit},\ \{\textit{read}\ R_1, \textit{write}\ R_2\}$

<u>Session types</u> describe communication *behaviour*

π-calculus as prototype

$$(\text{recv})\ \frac{\Gamma, x : \tau; \Delta, c : S \quad \vdash P}{\Gamma; \Delta, c : ?[\tau].S \vdash c?(\mathrm{x}).\mathrm{P}}$$

e.g. $c : ?[\mathbf{int}].![\mathbf{int}] \ \vdash\ c?(x).c!<x+1>$

Are they related?

λ-calculus

+ simple types

+ state

+ effect systems

embedding

this work*

π-calculus

+ sessions

+ session types

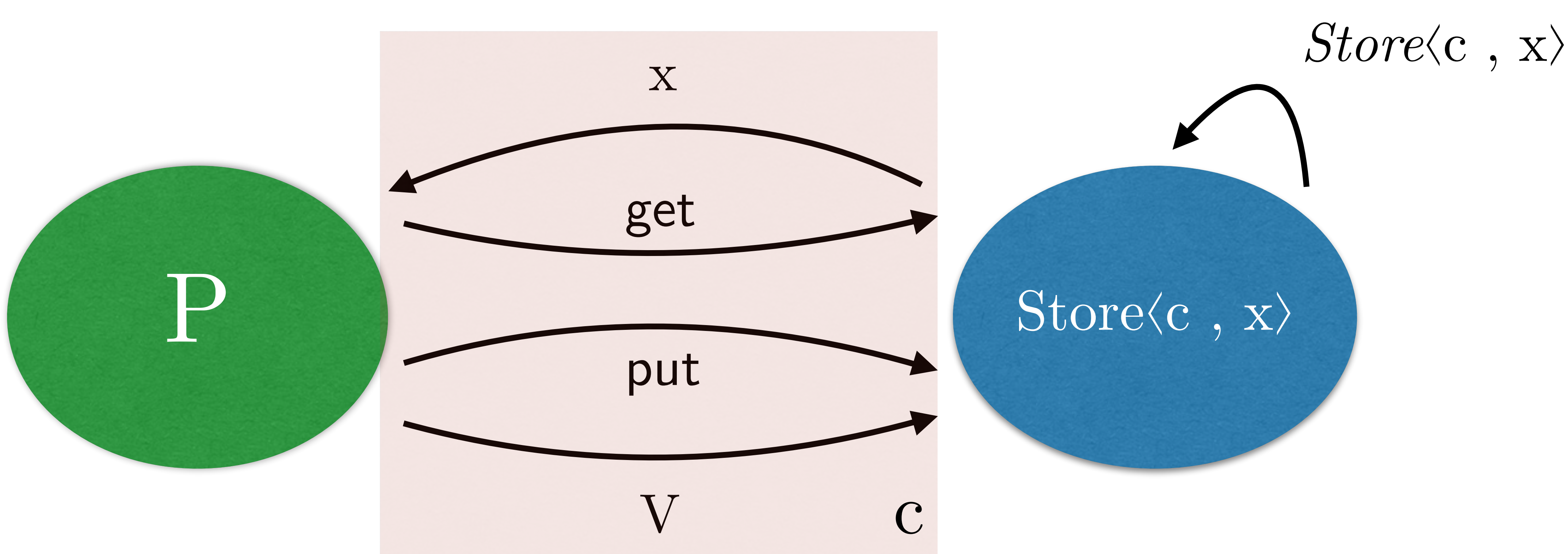
T

- Expressive power of session types
- Session types generalise causal effect systems

P

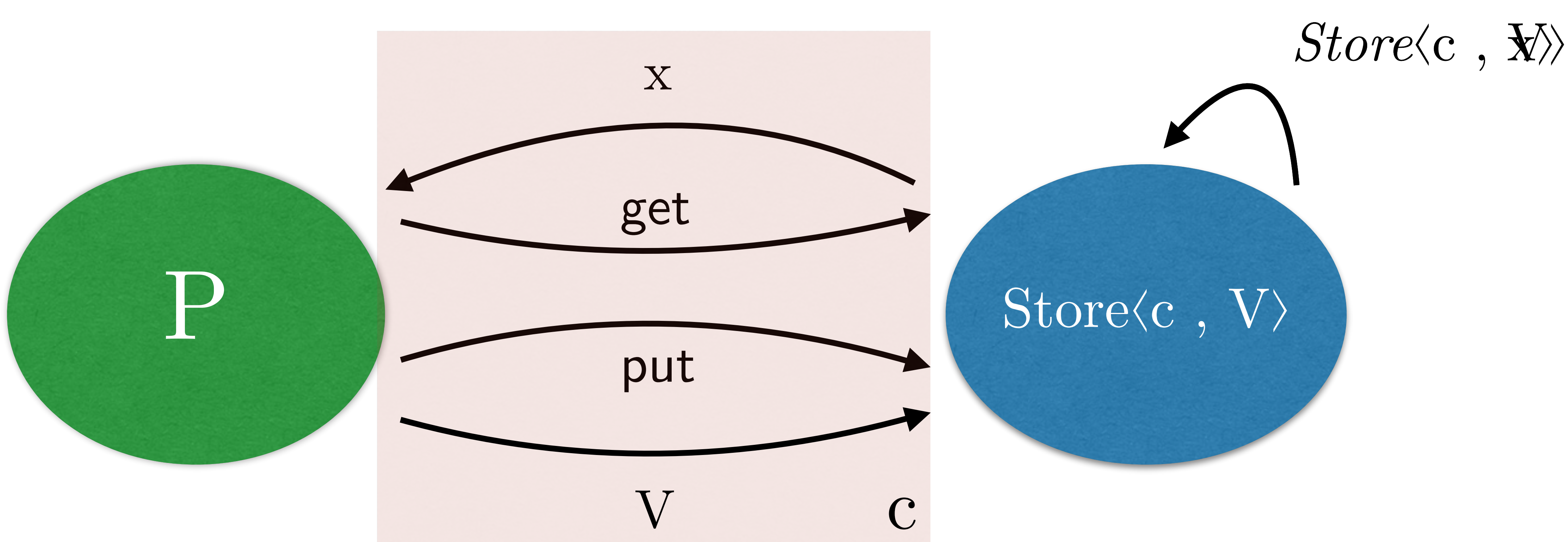
- π-calculus with effect system for free
- Concurrent effect semantics via π-calculus
- Compile into π-calculus

Variable agent



Variable agent

$$
\begin{aligned}
\mathbf{def}\ \mathit{Store}(\mathrm{c}, \mathrm{x}) = \mathrm{c} \triangleright \{ & \mathsf{get} : \mathrm{c}!\langle \mathrm{x} \rangle.\mathit{Store}\langle \mathrm{c}, \mathrm{x} \rangle, \\
& \mathsf{put} : \mathrm{c}?(\mathrm{y}).\mathit{Store}\langle \mathrm{c}, \mathrm{y} \rangle, \\
& \mathsf{stop} : \mathbf{0} \} \\
\mathbf{in}\ \mathit{Store}\langle \mathrm{c}\ ,\ \mathrm{i} \rangle &
\end{aligned}
$$



Variable agent

$$\begin{aligned}\textbf{def } \mathit{Store}(\mathrm{c}, \mathrm{x}) = \mathrm{c} \triangleright \{&\mathsf{get} : \mathrm{c}!\langle \mathrm{x}\rangle.\mathit{Store}\langle \mathrm{c}, \mathrm{x}\rangle,\\ &\mathsf{put} : \mathrm{c}?(\mathrm{y}).\mathit{Store}\langle \mathrm{c}, \mathrm{y}\rangle,\\ &\mathsf{stop} : \mathbf{0}\}\\ \textbf{in } \mathit{Store}\langle \mathrm{c}\ , \mathrm{i}\rangle\end{aligned}$$

Server

$$\mathrm{get}\ (\mathrm{c})(\mathrm{x}).\mathrm{P} = (\mathrm{c} \triangleleft \mathsf{get}).\mathrm{c}?(\mathrm{x}).\mathrm{P}$$

$$\mathrm{put}\ (\mathrm{c})\langle \mathrm{V}\rangle.\mathrm{P} = (\mathrm{c} \triangleleft \mathsf{put}).\mathrm{c}!\langle \mathrm{V}\rangle.\mathrm{P}$$

$$\mathrm{stop} = (\mathrm{c} \triangleleft \mathsf{stop}).\mathbf{0}$$

Client

e.g. increment store

$$\textbf{def } \mathit{Store}(\mathrm{c}, \mathrm{x}) = \ldots \textbf{ in } \ \mathbf{(}\mathrm{get}(\mathrm{c})(\mathrm{x}).\mathrm{put}(\mathrm{c})\langle \mathrm{x}+1\rangle.\mathrm{stop}\ \mid \mathit{Store}\langle \mathrm{c}\ , \mathrm{i}\rangle)$$

λ-calculus + simple types

+ state

+ effect systems

embedding

embedding

π-calculus + sessions

+ session types

$\Gamma \vdash \mathrm{M} : \tau,\ \mathsf{F}$

Effect calculus

abs $$\frac{\Gamma, x : \sigma \vdash M : \tau,\ \mathsf{F}}{\Gamma \vdash \lambda x.M : \sigma \xrightarrow{\mathsf{F}} \tau,\ \varnothing}$$

$$\frac{\Gamma \vdash M : \sigma,\ \mathsf{F} \qquad \Gamma, x : \sigma \vdash N : \tau,\ \mathsf{G}}{\Gamma \vdash \mathbf{let}\ x = M\ \mathbf{in}\ N : \tau,\ \mathsf{F} \bullet \mathsf{G}}$$

app $$\frac{\Gamma \vdash M : \sigma \xrightarrow{\mathsf{H}} \tau,\ \mathsf{F} \qquad \Gamma \vdash N : \sigma,\ \mathsf{G}}{\Gamma \vdash M\ N : \tau,\ \mathsf{F} \bullet \mathsf{G} \bullet \mathsf{H}}$$

var $$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma,\ \varnothing}$$

monoid $(\mathsf{F}, \bullet, \varnothing)$

Effect calculus for state

$$\frac{\Gamma \vdash V : \tau,\ [\,]}{\Gamma \vdash \mathsf{put}\ V : (),\ [\mathrm{put}\ \boldsymbol{\tau}]} \qquad \frac{}{\Gamma \vdash \mathsf{get} : \tau,\ [\mathrm{get}\ \boldsymbol{\tau}]}$$

$$(\mathrm{List}\ \{\mathrm{put}\ \boldsymbol{t}, \mathrm{get}\ \boldsymbol{t} \mid \boldsymbol{t} \in \tau\},\ +\!+,\ [\,])$$

e.g. increment store

$$\Gamma \vdash \mathbf{let}\ x = \mathsf{get}\ \mathbf{in}\ \mathsf{put}\ (x+1) : \mathrm{int},\ [\mathrm{get\ int},\ \mathrm{put\ int}]$$

$\Gamma\ ;\ \Delta \vdash \mathrm{P}$

π-calculus with sessions

$$\text{(recv)}\ \frac{\Gamma, x:\tau; \Delta, c:S \vdash P}{\Gamma; \Delta, c:?[\tau].S \vdash c?(x).P}$$

$$\text{(branch)}\ \frac{\Gamma; \Delta, c:S_i \vdash P_i}{\Gamma; \Delta, c:\&[\tilde{l}:\tilde{S}] \vdash c \triangleright \{\tilde{l}:\tilde{P}\}}$$

e.g. increment store

$$\mathrm{get}\ (\mathrm{c})(\mathrm{x}).\mathrm{P} = \mathrm{c} \lhd \mathrm{get}\ .\ \mathrm{c}?(\mathrm{x}).\mathrm{P}$$

$$\mathrm{put}\ (\mathrm{c})\langle \mathrm{V}\rangle.\mathrm{P} = \mathrm{c} \lhd \mathrm{put}\ .\ \mathrm{c}!\langle \mathrm{V}\rangle.\mathrm{P}$$

$$\mathrm{c} : \oplus[\mathrm{get} : ?[\mathbf{int}].\ \oplus[\mathrm{put} : ![\mathbf{int}].\mathbf{end}\ ]] \ \vdash\ \mathrm{get}(\mathrm{c})(\mathrm{x}).\mathrm{put}(\mathrm{c})\langle \mathrm{x}+1\rangle.\mathbf{0}$$

cf. effects $[\mathrm{get\ int,\ put\ int}]$

Sessions as effects

- ***<u>Effect handler</u>*** **process [e.g., variable agent]**

  [cf. Bauer, Pretnar "Progamming with algebraic effects and handlers."]

- ***<u>Effect channel</u>*** [a session channel for communicating with handler]

  … **whose session type <u>is</u> (encoding of) effect annotation**

- **"Threading" effect channel through control flow of encoding**

  [cf. state $\langle e, s\rangle \rightarrow \langle e', s'\rangle$ **or monadic semantics** $a \rightarrow M\ b$ **]**

State effect annotations as session types

$$[\![\ [\,]\ ]\!] = \mathbf{end}$$

$$[\![\ (\mathrm{get}\ \tau) : \mathsf{F}\ ]\!] = \oplus[\mathsf{get} : ?[\![\tau]\!].\ [\![\ \mathsf{F}\ ]\!]\ ]$$

$$[\![\ (\mathrm{get}\ \tau) : \mathsf{F}\ ]\!] = \oplus[\mathsf{put} : ![\![\tau]\!].\ [\![\ \mathsf{F}\ ]\!]\ ]$$

Embedding

(mid)

$$[\![\Gamma \vdash M : \tau, \mathsf{F}]\!]_r^{\mathit{eff}} = [\![\Gamma]\!], r : ![\![\tau]\!], \mathit{eff} : [\![\mathsf{F}]\!] \vdash \ldots$$
$$\nu ei, eo\ .\ ((\Gamma \vdash M : \tau, \mathsf{F})_r^{ei,eo} \mid \underline{ei}!\langle \mathit{eff} \rangle . eo(c))$$

(top)

$$[\![\Gamma \vdash M : \tau, \mathsf{F}]\!]_r =$$
$$[\![\Gamma]\!], r : ![\![\tau]\!] \vdash \nu \mathit{eff}\ .\ ([\![\Gamma \vdash M : \tau, \mathsf{F}]\!]_r^{\mathit{eff}} \mid H(\underline{\mathit{eff}}))$$

(low)

$$(\Gamma \vdash M : \tau, \mathsf{F})_r^{ei,eo} = \forall \mathsf{g}\ .$$
$$[\![\Gamma]\!], r : ![\![\tau]\!], ei : ?[\![\mathsf{F} \bullet \mathsf{g}]\!], \underline{eo} : ![\![\mathsf{g}]\!] \vdash \ldots$$

receive effect channel

send effect channel

Embedding (zeroth-order)

$$(\!|\ \Gamma \vdash x : \tau,\ \varnothing\ |\!)_r^{ei,eo} \quad = \quad ei?(c).r!\langle x\rangle.\underline{eo}!\langle c\rangle$$

where $\forall \mathsf{g}\ .\ [\![\ \Gamma\ ]\!];\ \ r : ![\![\tau]\!],\ ei : ?[\![\ \mathsf{g}\ ]\!],\ \underline{eo} : ![\![\ \mathsf{g}\ ]\!] \vdash ei?(c).r!\langle x\rangle.\underline{eo}!\langle c\rangle$

$$(\!|\ \Gamma \vdash \mathbf{let}\ x = M\ \mathbf{in}\ N : \tau,\ \mathsf{F} \bullet \mathsf{G}\ |\!)_r^{ei,eo} \quad =$$

$$\nu\ q,\ a\ .\ (\,(\!|\ M\ |\!)_q^{ei,\,a}\ \mid\ \underline{q}?(x).(\!|\ N\ |\!)_r^{a,\,eo}\,)$$

where $\forall \mathsf{h}\ .\quad q : ![\![\sigma]\!],\ \ ei : ?[\![\ \mathsf{F} \bullet \mathsf{h}\ ]\!],\ \underline{a} : ![\![\ \mathsf{h}\ ]\!] \vdash (\!|\ M\ |\!)_q^{ei,\,a}$

$\mathsf{h} \rightarrow \mathsf{G} \bullet \mathsf{h}'$

$\forall \mathsf{h}'\ .\quad x : [\![\sigma]\!];\ r : ![\![\tau]\!],\ a : ?[\![\ \mathsf{G} \bullet \mathsf{h}'\ ]\!],\ \ \underline{eo} : ![\![\ \mathsf{h}'\ ]\!] \vdash (\!|\ N\ |\!)_r^{a,\,eo}$

Embedding (zeroth-order)

$$(\!|\ \Gamma \vdash x : \tau,\ \varnothing\ |\!)^{ei,eo}_{r} \quad = \quad ei?(c).r!\langle x\rangle.\underline{eo}!\langle c\rangle$$

where $\quad \forall \mathsf{g}\ .\ \ [\![\ \Gamma\ ]\!];\ \ r : ![\![\tau]\!],\ ei : ?[\![\ \mathsf{g}\ ]\!],\ \underline{eo} : ![\![\ \mathsf{g}\ ]\!] \vdash ei?(c).r!\langle x\rangle.\underline{eo}!\langle c\rangle$

$$(\!|\ \Gamma \vdash \mathbf{let}\ x = M\ \mathbf{in}\ N : \tau,\ \mathsf{F} \bullet \mathsf{G}\ |\!)^{ei,eo}_{r} \ =$$

$$\nu\ q,\ a\ .\ \left((\!|\ M\ |\!)^{ei,\,a}_{q}\ \mid\ \underline{q}?(x).(\!|\ N\ |\!)^{a,\,eo}_{r}\right)$$

where $\quad \forall \mathsf{h}\ .\quad q : ![\![\sigma]\!],\ ei : ?[\![\ \mathsf{F} \bullet \mathsf{G} \bullet \mathsf{h}'\ ]\!],\ \underline{a} : ![\![\ \mathsf{G} \bullet \mathsf{h}'\ ]\!] \vdash (\!|\ M\ |\!)^{ei,\,a}_{q}$

$\qquad \mathsf{h} \rightarrow \mathsf{G} \bullet \mathsf{h}'$

$\forall \mathsf{h}'\ .\quad x : [\![\sigma]\!];\ r : ![\![\tau]\!],\ a : ?[\![\ \mathsf{G} \bullet \mathsf{h}'\ ]\!],\ \ \underline{eo} : ![\![\ \mathsf{h}'\ ]\!] \vdash (\!|\ N\ |\!)^{a,\,eo}_{r}$

Embedding (higher-order)

Must embed *latent effects* $\sigma \xrightarrow{F} \tau$

$$[\![\ \sigma \rightarrow \tau \ ]\!] = ![?[\![\ \sigma \ ]\!]] \ . \ ![\ ! \ [\![\tau]\!] \ ] \ . \ \mathbf{end}$$

$$[\![\ \sigma \xrightarrow{F} \tau \ ]\!] = ![?[\![\ \sigma \ ]\!]] \ . \ ![?[\![\ F \bullet G \ ]\!]] \ . \ ![![\![\ G \ ]\!]] \ . \ ![\ ! \ [\![\tau]\!] \ ] \ . \ \mathbf{end}$$

send channel which
can receive effect channel
for latent effects

send channel which
can send effect channel
for **continuation**

Embedding (higher-order)

$$(\!|\ \Gamma \vdash \lambda x\ .\ M : \sigma \xrightarrow{\mathsf{F}} \tau, \varnothing\ |\!)^{ei,eo}_{r} \quad =$$

$$\nu\ y\ .\ (ei?(c).\underline{eo}!\langle c\rangle.r\,!\langle y\rangle.{}^{*}y?(p,\ a,\ b,\ q).p?(x).(\!|\ M\ |\!)^{a,b}_{q})$$

$\forall \mathsf{g}, \mathsf{h}\ .$

$$r : ![\ ?[\![\ \sigma\ ]\!],\ ?[\![\ \mathsf{F} \bullet \mathsf{h}\ ]\!],\ ![\![\ \mathsf{h}\ ]\!],\ ![\![\tau]\!]\ ],$$

$$ei : ?[\![\ \mathsf{g}\ ]\!],\ \underline{eo} : ![\![\ \mathsf{g}\ ]\!]\ \vdash \nu\ y\ .\ (ei?\ \ldots.)$$

$$(\!|\ \Gamma \vdash\ M\ N : \tau, \mathsf{F} \bullet \mathsf{G} \bullet \mathsf{H}\ |\!)^{ei,eo}_{r} \ =$$

$$\nu\ q,s,a,b,p\ .\ ((\!|\ M\ |\!)^{ei,\,a}_{q} \mid (\!|\ N\ |\!)^{a,\,b}_{s} \mid \underline{q}?(y).\underline{s}?(arg).y!\langle p,\ b,\ eo,\ r\rangle.p!\langle arg\rangle)$$

Example

$$[\![\, \Gamma \,]\!],\ r : ![\![\mathrm{int}]\!],\ \mathit{eff} : [\![\ [\mathrm{get\ int},\ \mathrm{put\ int}]\ ]\!] \vdash$$

$$[\![\ \Gamma \vdash \mathbf{let}\ x = \mathrm{get}\ \mathbf{in}\ \mathrm{put}\ (x + 1) : \mathrm{int},\ [\mathrm{get\ int},\ \mathrm{put\ int}]\ ]\!]_r^{\mathit{eff}}$$

$$[\![\ \Gamma \vdash \mathbf{let}\ x \ldots\ ]\!]_r = \nu \mathit{eff}\,.\,([\![\ \Gamma \vdash \mathbf{let}\ x \ldots\ ]\!]_r^{\mathit{eff}} \ \mid\ \mathrm{Var}(\mathit{eff}, 0))$$

Soundness

$$\Gamma \vdash M \equiv N : \tau, F \implies$$

$$[\![\Gamma]\!]; (r :![\![\tau]\!].\mathbf{end}, e : [\![F]\!]) \vdash [\![M]\!]_r^e \approx [\![N]\!]_r^e$$

An application

- Effect-informed optimisations, e.g. implicit parallelism

if $\Gamma \vdash M : \sigma, \varnothing$ and $\Gamma \vdash N : \mathrm{t}, \mathrm{F}$

then $(\!| \, \mathbf{let}\, x \leftarrow M \,\mathbf{in}\, (\mathbf{let}\, y \leftarrow N \,\mathbf{in}\, P) \, |\!)_r^{\mathsf{ei},\mathsf{eo}}$

$$= \nu\, q, s, \mathsf{eb}.\, (\llbracket M \rrbracket_q \mid (\!| \, N \, |\!)_s^{\mathsf{ei},\mathsf{eb}} \mid \overline{q}?(x).\overline{s}?(y).(\!| \, P \, |\!)_r^{\mathsf{eb},\mathsf{eo}})$$

- Use this to give semantics of *concurrent effects*
  - e.g., *non-interference, atomicity via sessions*

Conclusion

- Sessions and session types expressive enough to encode effects with a **causal** effect system
    - Per effect notion [e.g., state, counting, I/O]:

      effect mapping, handler, encoding operations
- Extended to **case** and **fix**
- Set-based effect systems recovered by transforming causal

Thanks!

More details in our PLACES'15 *paper*; *see* dorchard.co.uk