FuSe
an(other) OCaml implementation of binary sessions

Luca Padovani

target API

Linear type theory for asynchronous session types (Gay & Vasconcelos 2010)

$$
\begin{aligned}
\texttt{accept} &: A\ \texttt{service} \to A \\
\texttt{request} &: A\ \texttt{service} \to \overline{A}
\end{aligned}
$$

$$
\begin{aligned}
\texttt{send} &: \alpha \to\ !\alpha.A \to A \\
\texttt{receive} &: ?\alpha.A \to \alpha \times A
\end{aligned}
$$

$$
\begin{aligned}
\texttt{select} &: (\overline{A}_k \to [\mathsf{C}_i : \overline{A}_i]_{i \in I}) \to \oplus[\mathsf{C}_i : A_i]_{i \in I} \to A_k \\
\texttt{branch} &: \&[\mathsf{C}_i : A_i]_{i \in I} \to [\mathsf{C}_i : A_i]_{i \in I}
\end{aligned}
$$

$$
\begin{aligned}
\texttt{create} &: \texttt{unit} \to A \times \overline{A} \\
\texttt{close} &: \texttt{end} \to \texttt{unit}
\end{aligned}
$$

duality and linearity in `OCaml`

Turn duality into equality

- Dardha et al. 2012 + tweak

duality and linearity in OCaml

Turn duality into equality

- Dardha et al. 2012 + tweak

Ostrich approach to linearity

- no type-based mechanism, no monads, …
- affinity violations detected at runtime
  (Tov & Pucella 2010, Hu & Yoshida 2016)
- some (many?) violations detected by OCaml anyway

duality $\Rightarrow$ equality

Session types revisited (Dardha et al., PPDP 2012)

$$[\![T]\!] \quad = \quad \kappa[t]$$

$$\begin{aligned} [\![!t.T]\!] &= ![t \times [\![\overline{T}]\!]] \\ [\![?t.T]\!] &= ?[t \times [\![T]\!]] \end{aligned}$$

$$[\![T]\!] = \kappa[t] \quad \Rightarrow \quad [\![\overline{T}]\!] = \overline{\kappa}[t]$$

duality $\Rightarrow$ equality

Session types revisited + tweak (Dardha et al., PPDP 2012)

$$[\![T]\!] = \langle t, s \rangle$$

$$\begin{aligned} [\![!t.T]\!] &= \langle \mathbb{0}, t \times [\![\overline{T}]\!] \rangle \\ [\![?t.T]\!] &= \langle t \times [\![T]\!], \mathbb{0} \rangle \end{aligned}$$

$$[\![T]\!] = \langle t, s \rangle \Rightarrow [\![\overline{T}]\!] = \langle s, t \rangle$$

target API (encoded)

$$
\begin{aligned}
\texttt{accept} &: \langle \alpha, \beta \rangle\ \texttt{service} \to \langle \alpha, \beta \rangle \\
\texttt{request} &: \langle \alpha, \beta \rangle\ \texttt{service} \to \langle \beta, \alpha \rangle \\
\\
\texttt{send} &: \gamma \to \langle \mathbb{0}, \gamma \times \langle \beta, \alpha \rangle \rangle \to \langle \alpha, \beta \rangle \\
\texttt{receive} &: \langle \gamma \times \langle \alpha, \beta \rangle, \mathbb{0} \rangle \to \gamma \times \langle \alpha, \beta \rangle \\
\\
\texttt{select} &: (\langle \beta, \alpha \rangle \to \gamma) \to \langle \mathbb{0}, \gamma \rangle \to \langle \alpha, \beta \rangle \\
\texttt{branch} &: \langle \gamma, \mathbb{0} \rangle \to \gamma \\
\\
\texttt{create} &: \texttt{unit} \to \langle \alpha, \beta \rangle \times \langle \beta, \alpha \rangle \\
\texttt{close} &: \langle \mathbb{0}, \mathbb{0} \rangle \to \texttt{unit}
\end{aligned}
$$

DEMO

wrap-up

- GV-style sessions (Gay & Vasconcelos, JFP 2010)
- context-free session types (Thiemann & Vasconcelos, ICFP 2016)
- subtyping (Gay & Hole, Acta 2005)
- chaperone contracts (ongoing with Hernán Melgratti)

References

- FuSe details and implementation (Padovani, JFP 2017)
- context-free session types in FuSe (Padovani, ESOP 2017)