Non-Idempotent Intersection Types For Global State

Miguel Ramos

LIACC, DCC, Universidade do Porto

IRIF, CNRS, Université Paris Cité

ITRS'24

Lambda-Calculi

Lambda-Calculi

The Lambda-Calculus

$$t, u ::= x \mid \lambda x.t \mid tu$$

$$(\lambda x.t)u \rightsquigarrow_\beta t\{x \backslash u\}$$

Lambda-Calculi

The Lambda-Calculus

$$t, u ::= x \mid \lambda x.t \mid tu$$

$$(\lambda x.t)u \rightsquigarrow_\beta t\{x\backslash u\}$$

Plotkin's CBN

$$\frac{}{(\lambda x.t)u \rightsquigarrow t\{x\backslash u\}}$$

$$\frac{t \rightsquigarrow t'}{tu \rightsquigarrow t'u}$$

Lambda-Calculi

The Lambda-Calculus

$$t, u ::= x \mid \lambda x.t \mid tu$$

$$(\lambda x.t)u \rightsquigarrow_\beta t\{x \backslash u\}$$

Plotkin's CBN

$$\frac{}{(\lambda x.t)u \rightsquigarrow t\{x \backslash u\}}$$

$$\frac{t \rightsquigarrow t'}{tu \rightsquigarrow t'u}$$

Plotkin's CBV

$$\boxed{v} ::= x \mid \lambda x.t$$

$$\frac{}{(\lambda x.t)\boxed{v} \rightsquigarrow t\{x \backslash \boxed{v}\}}$$

$$\frac{t \rightsquigarrow t'}{tu \rightsquigarrow t'u} \qquad \frac{t \rightsquigarrow t'}{\boxed{v}t \rightsquigarrow \boxed{v}t'}$$

Core CBV Lambda-Calculus

$$
\begin{aligned}
v &::= x \mid \lambda x.t \\
t &::= v \mid \boxed{vt}
\end{aligned}
$$

Core CBV Lambda-Calculus

$$
\begin{array}{rcl}
v & ::= & x \mid \lambda x.t \\
t & ::= & v \mid \boxed{vt} \mid \cdots \\
 & \vdots &
\end{array}
$$

Core CBV Lambda-Calculus

$$\begin{array}{rcl} v & ::= & x \mid \lambda x.t \\ t & ::= & v \mid vt \mid \cdots \\ & \vdots & \end{array}$$

$$\frac{}{(\lambda x.t)v \rightsquigarrow t\{x \backslash v\}} \qquad \frac{t \rightsquigarrow t'}{vt \rightsquigarrow vt'}$$

Core CBV Lambda-Calculus

$$
\begin{array}{rcl}
v & ::= & x \mid \lambda x.t \\
t & ::= & v \mid \boxed{vt} \mid \cdots \\
 & \vdots &
\end{array}
$$

$$
\frac{}{(\lambda x.t)v \rightsquigarrow t\{x \backslash v\}} \qquad \frac{t \rightsquigarrow t'}{vt \rightsquigarrow vt'}
$$

$$
tu := (\lambda x.xu)t
$$

Global State
(Operational Semantics)

Mapping from Locations to Values

$$\ell_1 \mapsto v_1 \quad \cdots \quad \ell_i \mapsto v_i \quad \cdots \quad \ell_n \mapsto v_n$$

Global State
(Operational Semantics)

Mapping from Locations to Values

$$\ell_1 \mapsto v_1 \quad \cdots \quad \ell_i \mapsto v_i \quad \cdots \quad \ell_n \mapsto v_n$$

State and Configurations

Global State
(Operational Semantics)

Mapping from Locations to Values

$$\ell_1 \mapsto v_1 \quad \cdots \quad \ell_i \mapsto v_i \quad \cdots \quad \ell_n \mapsto v_n$$

State and Configurations

$$s \quad ::= \quad \epsilon \mid \mathtt{upd}_\ell(v, s)$$

Global State
(Operational Semantics)

Mapping from Locations to Values

$$\ell_1 \mapsto v_1 \quad \cdots \quad \ell_i \mapsto v_i \quad \cdots \quad \ell_n \mapsto v_n$$

State and Configurations

$$\begin{aligned} s &::= \epsilon \mid \mathtt{upd}_\ell(v, s) \\ c &::= (t, s) \end{aligned}$$

Global State
(Operational Semantics)

Mapping from Locations to Values

$$\ell_1 \mapsto v_1 \quad \cdots \quad \ell_i \mapsto v_i \quad \cdots \quad \ell_n \mapsto v_n$$

State and Configurations

$$\begin{array}{lcl} s & ::= & \epsilon \mid \mathrm{upd}_\ell(v, s) \\ c & ::= & (t, s) \end{array}$$

$$\frac{}{((\lambda x.t)v, s) \rightsquigarrow (t\{x \backslash v\}, s)} \qquad \frac{(t, s) \rightsquigarrow (t', s)}{(vt, s) \rightsquigarrow (vt', s)}$$

Global State
(Operational Semantics)

Interacting with the State

$$t \quad ::= \quad v \mid vt$$

Global State
(Operational Semantics)

Interacting with the State

$$t \quad ::= \quad v \mid vt \mid \mathtt{get}_{\ell}(\lambda x.t)$$

Global State
(Operational Semantics)

Interacting with the State

$$t \quad ::= \quad v \mid vt \mid \texttt{get}_\ell(\lambda x.t)$$

Get Value from Location

$$\frac{}{(\texttt{get}_\ell(\lambda x.t), s) \rightsquigarrow (t\{x \backslash \texttt{lkp}_\ell(s)\}, s)}$$

Global State
(Operational Semantics)

Interacting with the State

$$t \quad ::= \quad v \mid vt \mid \mathtt{get}_\ell(\lambda x.t) \mid \mathtt{set}_\ell(v, t)$$

Get Value from Location

$$\frac{}{(\mathtt{get}_\ell(\lambda x.t), s) \rightsquigarrow (t\{x \backslash \mathtt{lkp}_\ell(s)\}, s)}$$

Global State
(Operational Semantics)

Interacting with the State

$$t \quad ::= \quad v \mid vt \mid \texttt{get}_\ell(\lambda x.t) \mid \texttt{set}_\ell(v, t)$$

Get Value from Location

$$\frac{}{(\texttt{get}_\ell(\lambda x.t), s) \rightsquigarrow (t\{x \backslash \texttt{lkp}_\ell(s)\}, s)}$$

Set Value to Location

$$\frac{}{(\texttt{set}_\ell(v, t), s) \rightsquigarrow (t, \texttt{upd}_\ell(v, s))}$$

Global State

(*Monadic* Operational Semantics)

Global State
(*Monadic* Operational Semantics)

Algebraic Theory for Global State Monad

1. ... $\cdots$ 4. $\texttt{set}_\ell(v, \texttt{get}_\ell(\lambda x.t)) = \texttt{set}_\ell(v, t\{x \backslash v\})$ $\cdots$ 7. ...

Global State

(*Monadic* Operational Semantics)

Algebraic Theory for Global State Monad

1. ... $\cdots$ 4. $\texttt{set}_\ell(v, \texttt{get}_\ell(\lambda x.t)) = \texttt{set}_\ell(v, t\{x \backslash v\})$ $\cdots$ 7. ...

Algebraic Theory for Array Monad

$$
\begin{array}{lrcll}
1. & \texttt{lkp}_\ell(\texttt{upd}_\ell(v, s)) & = & v & \\
2. & \texttt{upd}_\ell(\texttt{lkp}_\ell(s), s) & = & s & \\
3. & \texttt{upd}_\ell(v', \texttt{upd}_\ell(v, s)) & = & \texttt{upd}_\ell(v', s) & \\
4. & \texttt{upd}_{\ell'}(v', \texttt{upd}_\ell(v, s)) & = & \texttt{upd}_\ell(v, \texttt{upd}_{\ell'}(v', s)) & \ell \neq \ell'
\end{array}
$$

Global State
(*Monadic* Operational Semantics)

$$(\texttt{set}_\ell(v, \texttt{get}_\ell(\lambda x.x)), \texttt{upd}_\ell(v', \epsilon))$$

Global State
(*Monadic* Operational Semantics)

$$(\texttt{set}_\ell(v, \texttt{get}_\ell(\lambda x.x)), \texttt{upd}_\ell(v', \epsilon)) \rightsquigarrow (\texttt{get}_\ell(\lambda x.x), \texttt{upd}_\ell(v, \epsilon))$$

Global State
(*Monadic* Operational Semantics)

$$(\texttt{set}_\ell(v, \texttt{get}_\ell(\lambda x.x)), \texttt{upd}_\ell(v', \epsilon)) \rightsquigarrow (\texttt{get}_\ell(\lambda x.x), \texttt{upd}_\ell(v, \epsilon)) \rightsquigarrow (v, \texttt{upd}_\ell(v, \epsilon))$$

Global State

(*Monadic* Operational Semantics)

$$(\mathtt{set}_\ell(v, \mathtt{get}_\ell(\lambda x.x)), \mathtt{upd}_\ell(v', \epsilon)) \rightsquigarrow (\mathtt{get}_\ell(\lambda x.x), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (v, \mathtt{upd}_\ell(v, \epsilon))$$

Assuming that $\mathtt{lkp}_\ell(s)$ never "fails" ...

Global State
(*Monadic* Operational Semantics)

$$(\mathtt{set}_\ell(v, \mathtt{get}_\ell(\lambda x.x)), \mathtt{upd}_\ell(v', \epsilon)) \rightsquigarrow (\mathtt{get}_\ell(\lambda x.x), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (v, \mathtt{upd}_\ell(v, \epsilon))$$

Assuming that $\mathtt{lkp}_\ell(s)$ never "fails" ...

$$\frac{}{(\mathtt{get}_\ell(\lambda x.t), \mathtt{upd}_\ell(v, s)) \rightsquigarrow (t\{x \backslash v\}, s)}$$

Intersection Types

$$\begin{aligned} \sigma &::= \tau \mid \sigma \wedge \sigma \\ \tau &::= \omega \mid \sigma \to \tau \end{aligned}$$

Intersection Types

$$\begin{array}{rcl} \sigma & ::= & \tau \mid \sigma \wedge \sigma \\ \tau & ::= & \omega \mid \sigma \to \tau \end{array}$$

$$(\sigma_1 \wedge \sigma_2) \wedge \sigma_3 \quad = \quad \sigma_1 \wedge (\sigma_2 \wedge \sigma_3)$$

Intersection Types

$$\begin{array}{rcl} \sigma & ::= & \tau \mid \sigma \wedge \sigma \\ \tau & ::= & \omega \mid \sigma \to \tau \end{array}$$

$$\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \end{array}$$

Intersection Types

$$
\begin{aligned}
\sigma &::= \tau \mid \sigma \wedge \sigma \\
\tau &::= \omega \mid \sigma \to \tau
\end{aligned}
$$

$$
\begin{aligned}
(\sigma_1 \wedge \sigma_2) \wedge \sigma_3 &= \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\
\sigma_1 \wedge \sigma_2 &= \sigma_2 \wedge \sigma_1 \\
\sigma \wedge \sigma &= \sigma
\end{aligned}
$$

Intersection Types

$$\begin{array}{rcl} \sigma & ::= & \tau \mid \sigma \wedge \sigma \\ \tau & ::= & \omega \mid \sigma \to \tau \end{array}$$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \\ \sigma \wedge \sigma & = & \sigma \end{array}\right\} \quad \begin{array}{rcl} \sigma & ::= & \{\tau_i\}_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & \{\,\} \mid \sigma \to \tau \end{array}$$

Intersection Types

$$\begin{array}{rcl}\sigma & ::= & \tau \mid \sigma \wedge \sigma \\ \tau & ::= & \omega \mid \sigma \to \tau\end{array}$$

$$\left.\begin{array}{rcl}(\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \\ \sigma \wedge \sigma & = & \sigma\end{array}\right\} \quad \begin{array}{rcl}\sigma & ::= & \{\tau_i\}_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & \{\,\} \mid \sigma \to \tau\end{array}$$

Typability $\Leftrightarrow$ Termination

Intersection Types

$$\begin{array}{rcl} \sigma & ::= & \tau \mid \sigma \wedge \sigma \\ \tau & ::= & \omega \mid \sigma \to \tau \end{array}$$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \\ \sigma \wedge \sigma & = & \sigma \end{array}\right\} \quad \begin{array}{rcl} \sigma & ::= & \{\tau_i\}_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & \{\,\} \mid \sigma \to \tau \end{array}$$

Typability $\Leftrightarrow$ Termination

$$x : \{\{\tau\} \to \sigma, \tau\} \vdash xx : \tau$$

Non-Idempotent Intersection Types

$$\sigma \wedge \sigma \neq \sigma$$

Non-Idempotent Intersection Types

$$\sigma \wedge \sigma \neq \sigma$$

$$\begin{aligned} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 &= \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 &= \sigma_2 \wedge \sigma_1 \end{aligned}$$

Non-Idempotent Intersection Types

$$\sigma \wedge \sigma \neq \sigma$$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \end{array}\right\} \begin{array}{rcl} m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & [\,] \mid m \to \tau \end{array}$$

Non-Idempotent Intersection Types

$$\sigma \wedge \sigma \neq \sigma$$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \end{array}\right\} \begin{array}{rcl} m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & [\,] \mid m \rightarrow \tau \end{array}$$

Typability $\Leftrightarrow$ Termination

Non-Idempotent Intersection Types

$$\sigma \wedge \sigma \neq \sigma$$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \end{array}\right\} \begin{array}{rcl} m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & [\,] \mid m \rightarrow \tau \end{array}$$

Typability $\Leftrightarrow$ Termination

$$x : [[\tau] \rightarrow [\tau] \rightarrow \tau, \tau, \tau] \vdash xxx : \tau$$

Non-Idempotent Intersection Types

$\sigma \wedge \sigma \neq \sigma$

$$\left.\begin{array}{rcl} (\sigma_1 \wedge \sigma_2) \wedge \sigma_3 & = & \sigma_1 \wedge (\sigma_2 \wedge \sigma_3) \\ \sigma_1 \wedge \sigma_2 & = & \sigma_2 \wedge \sigma_1 \end{array}\right\} \begin{array}{rcl} m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\ \tau & ::= & [\,] \mid m \to \tau \end{array}$$

Typability $\Leftrightarrow$ Termination

$$x : [[\tau] \to [\tau] \to \tau, \tau, \tau] \vdash xxx : \tau$$

Resource Aware $\Rightarrow$ Combinatorial Arguments

Type System for Core CBV Lambda-Calculus

$$
\begin{array}{lcl}
m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\
\tau & ::= & m \rightarrow m
\end{array}
$$

Type System for Core CBV Lambda-Calculus

$$m \quad ::= \quad [\tau_i]_{i\in I} \text{ where } I \text{ is a finite set}$$
$$\tau \quad ::= \quad m \rightarrow m$$

$$\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma \vdash v : m \rightarrow m' \qquad \Delta \vdash t : m}{\Gamma \sqcup \Delta \vdash vt : m'}\ \texttt{app}$$

$$\frac{\Gamma; x : m \vdash t : m'}{\Gamma \vdash \lambda x.t : m \rightarrow m'}\ \texttt{abs} \qquad \frac{(\Gamma_i \vdash v : \tau_i)_{i\in I}}{\sqcup_{i\in I}\Gamma_i \vdash v : [\tau_i]_{i\in I}}\ \texttt{many}$$

Type System for Core CBV Lambda-Calculus

$$
\begin{array}{rcl}
m & ::= & [\tau_i]_{i \in I} \text{ where } I \text{ is a finite set} \\
\tau & ::= & m \to m
\end{array}
$$

$$
\frac{}{x : [\tau] \vdash x : \tau}\ \texttt{ax} \qquad \frac{\Gamma \vdash v : m \to m' \qquad \Delta \vdash t : m}{\Gamma \sqcup \Delta \vdash vt : m'}\ \texttt{app}
$$

$$
\frac{\Gamma; x : m \vdash t : m'}{\Gamma \vdash \lambda x.t : m \to m'}\ \texttt{abs} \qquad \frac{(\Gamma_i \vdash v : \tau_i)_{i \in I}}{\sqcup_{i \in I} \Gamma_i \vdash v : [\tau_i]_{i \in I}}\ \texttt{many}
$$

Typability $\Leftrightarrow$ Termination

$\Phi \triangleright \emptyset \vdash t : [\,]$ iff t is terminating (in exactly $|\Phi|$ steps)

Non-Idempotent Intersection Types for Global State

Monadic Intersection Types

Non-Idempotent Intersection Types for Global State

Monadic Intersection Types

$\epsilon \mathbin{\&} \mathtt{upd}_\ell(v, s)$

$$\sigma \quad ::= \quad \epsilon \mid \mathtt{upd}_\ell(m, \sigma)$$

Non-Idempotent Intersection Types for Global State

Monadic Intersection Types

$$\epsilon \;\&\; \mathtt{upd}_\ell(v, s) \qquad \sigma \;::=\; \epsilon \mid \mathtt{upd}_\ell(m, \sigma)$$

$$(v, s) \qquad \pi \;::=\; m \times \sigma$$

Non-Idempotent Intersection Types for Global State

Monadic Intersection Types

$\epsilon \,\&\, \mathtt{upd}_{\ell}(v, s)$	σ	$::=$	$\epsilon \mid \mathtt{upd}_{\ell}(m, \sigma)$
(v, s)	π	$::=$	$m \times \sigma$
$\lambda x.t$	τ	$::=$	$m \to \mu$

Non-Idempotent Intersection Types for Global State

Monadic Intersection Types

$$
\begin{array}{ccl}
\epsilon \;\&\; \mathtt{upd}_\ell(v, s) & \sigma & ::= \;\; \epsilon \mid \mathtt{upd}_\ell(m, \sigma) \\
(v, s) & \pi & ::= \;\; m \times \sigma \\
\lambda x.t & \tau & ::= \;\; m \to \mu \\
\mathtt{get}_\ell(\lambda x.t) \;\&\; \mathtt{set}_\ell(v, t) & \mu & ::= \;\; \sigma \Rightarrow \pi
\end{array}
$$

Type System for Core CBV Lambda-Calculus with Global State

Type System for Core CBV Lambda-Calculus with Global State

$$\frac{\Gamma; x : m \vdash t : \sigma \Rightarrow \pi}{\Gamma \vdash \mathtt{get}_\ell(\lambda x.t) : \mathtt{upd}_\ell(m \sqcup m', \sigma) \Rightarrow \pi}\ \mathtt{get} \qquad \frac{\Gamma \vdash v : m \qquad \Delta \vdash t : \mathtt{upd}_\ell(m, \sigma) \Rightarrow \pi}{\Gamma \sqcup \Delta \vdash \mathtt{set}_\ell(v, t) : \sigma \to \pi}\ \mathtt{set}$$

Type System for Core CBV Lambda-Calculus with Global State

$$\frac{\Gamma; x : m \vdash t : \sigma \Rightarrow \pi}{\Gamma \vdash \mathtt{get}_\ell(\lambda x.t) : \mathtt{upd}_\ell(m \sqcup m', \sigma) \Rightarrow \pi}\ \mathtt{get} \qquad \frac{\Gamma \vdash v : m \qquad \Delta \vdash t : \mathtt{upd}_\ell(m, \sigma) \Rightarrow \pi}{\Gamma \sqcup \Delta \vdash \mathtt{set}_\ell(v, t) : \sigma \to \pi}\ \mathtt{set}$$

$$\frac{}{\emptyset \vdash \epsilon : \epsilon}\ \mathtt{e\text{-}state} \qquad \frac{\Gamma \vdash v : m \qquad \Delta \vdash s : \sigma}{\Gamma \sqcup \Delta \vdash \mathtt{upd}_\ell(v, s) : \mathtt{upd}_\ell(m, \sigma)}\ \mathtt{upd\text{-}state}$$

Type System for Core CBV Lambda-Calculus with Global State

$$\frac{\Gamma; x : m \vdash t : \sigma \Rightarrow \pi}{\Gamma \vdash \mathtt{get}_\ell(\lambda x.t) : \mathtt{upd}_\ell(m \sqcup m', \sigma) \Rightarrow \pi}\ \mathtt{get} \qquad \frac{\Gamma \vdash v : m \quad \Delta \vdash t : \mathtt{upd}_\ell(m, \sigma) \Rightarrow \pi}{\Gamma \sqcup \Delta \vdash \mathtt{set}_\ell(v, t) : \sigma \to \pi}\ \mathtt{set}$$

$$\frac{}{\emptyset \vdash \epsilon : \epsilon}\ \mathtt{e\text{-}state} \qquad \frac{\Gamma \vdash v : m \quad \Delta \vdash s : \sigma}{\Gamma \sqcup \Delta \vdash \mathtt{upd}_\ell(v, s) : \mathtt{upd}_\ell(m, \sigma)}\ \mathtt{upd\text{-}state}$$

$$\frac{\Gamma \vdash t : \sigma \Rightarrow \pi \quad \Delta \vdash s : \sigma}{\Gamma \sqcup \Delta \vdash (t, s) : \pi}\ \mathtt{conf}$$

Problem: No Subject Expansion

$$(\texttt{set}_\ell(v', t), \texttt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \texttt{upd}_\ell(v', \epsilon))$$

Problem: No Subject Expansion

$$(\mathtt{set}_\ell(v', t), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \mathtt{upd}_\ell(v', \epsilon))$$

$$\dfrac{\Phi_t \triangleright \emptyset \vdash t : \mathtt{upd}_\ell(m', \epsilon) \Rightarrow \pi \qquad \dfrac{\Phi_{v'} \triangleright \emptyset \vdash v' : m' \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \texttt{e-state}}{\emptyset \vdash \mathtt{upd}_\ell(v', \epsilon) : \mathtt{upd}_\ell(m', \epsilon)}\ \texttt{upd-state}}{\emptyset \vdash (t, \mathtt{upd}_\ell(v', \epsilon)) : \pi}\ \texttt{conf}$$

Problem: No Subject Expansion

$$(\texttt{set}_\ell(v', t), \texttt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \texttt{upd}_\ell(v', \epsilon))$$

$$\dfrac{\Phi_t \triangleright \emptyset \vdash t : \texttt{upd}_\ell(m', \epsilon) \Rightarrow \pi \qquad \dfrac{\Phi_{v'} \triangleright \emptyset \vdash v' : m' \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \texttt{e-state}}{\emptyset \vdash \texttt{upd}_\ell(v', \epsilon) : \texttt{upd}_\ell(m', \epsilon)}\ \texttt{upd-state}}{\emptyset \vdash (t, \texttt{upd}_\ell(v', \epsilon)) : \pi}\ \texttt{conf}$$

$$\dfrac{\dfrac{\Phi_{v'} \qquad \Phi_t}{\emptyset \vdash \texttt{set}_\ell(v', t) : \boxed{\epsilon} \Rightarrow \pi}\ \texttt{set} \qquad \dfrac{\dfrac{}{\emptyset \vdash v : [\,]}\ \texttt{many} \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \texttt{e-state}}{\emptyset \vdash \texttt{upd}_\ell(v, \epsilon) : \boxed{\texttt{upd}_\ell([\,], \epsilon)}}\ \texttt{upd-state}}{\emptyset \vdash (\texttt{set}_\ell(v', t), \texttt{upd}_\ell(v, \epsilon)) : \boxed{?}}\ \texttt{conf}$$

Reason: Loss of Information

Array Monad

1.	$\mathtt{lkp}_\ell(\mathtt{upd}_\ell(v, s))$	$=$	v	
2.	$\mathtt{upd}_\ell(\mathtt{lkp}_\ell(s), s)$	$=$	s	
3.	$\mathtt{upd}_\ell(v', \mathtt{upd}_\ell(v, s))$	$=$	$\mathtt{upd}_\ell(v', s)$	
4.	$\mathtt{upd}_{\ell'}(v', \mathtt{upd}_\ell(v, s))$	$=$	$\mathtt{upd}_\ell(v, \mathtt{upd}_{\ell'}(v', s))$	$\ell \neq \ell'$

Reason: Loss of Information

Array *with Log* Monad

1.	$\mathtt{lkp}_\ell(\mathtt{upd}_\ell(v, s))$	$=$	v	
2.	~~$\mathtt{upd}_\ell(\mathtt{lkp}_\ell(s), s)$~~	$=$	~~s~~	
3.	~~$\mathtt{upd}_\ell(v', \mathtt{upd}_\ell(v, s))$~~	$=$	~~$\mathtt{upd}_\ell(v', s)$~~	
4.	$\mathtt{upd}_{\ell'}(v', \mathtt{upd}_\ell(v, s))$	$=$	$\mathtt{upd}_\ell(v, \mathtt{upd}_{\ell'}(v', s))$	$\ell \neq \ell'$

Dropping Equations 2 & 3 = Adding a Log

Problem Solved: Subject Expansion Holds

$$(\mathtt{set}_\ell(v', t), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \mathtt{upd}_\ell(v', \mathtt{upd}_\ell(v, \epsilon)))$$

Problem Solved: Subject Expansion Holds

$$(\mathtt{set}_\ell(v', t), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \mathtt{upd}_\ell(v', \mathtt{upd}_\ell(v, \epsilon)))$$

$$\dfrac{\Phi_t \triangleright \emptyset \vdash t : \mathtt{upd}_\ell(m', \mathtt{upd}_\ell(m, \epsilon)) \Rightarrow \pi \qquad \dfrac{\Phi_{v'} \qquad \dfrac{\Phi_v \triangleright \emptyset \vdash v : m \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \mathtt{e\text{-}state}}{\emptyset \vdash \mathtt{upd}_\ell(v, \epsilon) : \mathtt{upd}_\ell(m, \epsilon)}\ \mathtt{upd\text{-}state}}{\emptyset \vdash \mathtt{upd}_\ell(v', \epsilon) : \mathtt{upd}_\ell(m', \mathtt{upd}_\ell(m, \epsilon))}\ \mathtt{upd\text{-}state}}{\emptyset \vdash (t, \mathtt{upd}_\ell(v', \epsilon)) : \pi}\ \mathtt{conf}$$

Problem Solved: Subject Expansion Holds

$$(\mathtt{set}_\ell(v', t), \mathtt{upd}_\ell(v, \epsilon)) \rightsquigarrow (t, \mathtt{upd}_\ell(v', \mathtt{upd}_\ell(v, \epsilon)))$$

$$\dfrac{\Phi_t \triangleright \emptyset \vdash t : \mathtt{upd}_\ell(m', \mathtt{upd}_\ell(m, \epsilon)) \Rightarrow \pi \qquad \dfrac{\Phi_{v'} \qquad \dfrac{\Phi_v \triangleright \emptyset \vdash v : m \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \texttt{e-state}}{\emptyset \vdash \mathtt{upd}_\ell(v, \epsilon) : \mathtt{upd}_\ell(m, \epsilon)}\ \texttt{upd-state}}{\emptyset \vdash \mathtt{upd}_\ell(v', \epsilon) : \mathtt{upd}_\ell(m', \mathtt{upd}_\ell(m, \epsilon))}\ \texttt{upd-state}}{\emptyset \vdash (t, \mathtt{upd}_\ell(v', \epsilon)) : \pi}\ \texttt{conf}$$

$$\dfrac{\dfrac{\Phi_{v'} \qquad \Phi_t}{\emptyset \vdash \mathtt{set}_\ell(v', t) : \boxed{\mathtt{upd}_\ell(m, \epsilon)} \Rightarrow \pi}\ \texttt{set} \qquad \dfrac{\Phi_v \qquad \dfrac{}{\emptyset \vdash \epsilon : \epsilon}\ \texttt{e-state}}{\emptyset \vdash \mathtt{upd}_\ell(v, \epsilon) : \boxed{\mathtt{upd}_\ell(m, \epsilon)}}\ \texttt{upd-state}}{\emptyset \vdash (\mathtt{set}_\ell(v', t), \mathtt{upd}_\ell(v, \epsilon)) : \boxed{\pi}}\ \texttt{conf}$$

Main Result

Typability ⇔ Termination

$\Phi \triangleright \emptyset \vdash (t, s) : [\,] \times \sigma$ iff (t, s) is terminating (in exactly $|\Phi|$ step)

Main Result

Typability $\Leftrightarrow$ Termination

$\Phi \triangleright \emptyset \vdash (t, s) : [\,] \times \sigma^{\dagger}$ iff (t, s) is terminating (in exactly $|\Phi|$ step)

† all locations are mapped to the empty multiset type

Main Result

Typability ⇔ Termination

$\Phi \triangleright \emptyset \vdash (t, s) : [\,] \times \sigma^{\dagger}$ iff (t, s) is terminating* (in exactly $|\Phi|$ step)

† all locations are mapped to the empty multiset type

* without getting *stuck*

The End

References

- Sandra Alves, Delia Kesner, and Miguel Ramos.
  Quantitative global memory.
- Ugo de'Liguoro and Riccardo Treglia.
  Intersection types for a λ-calculus with global store.
- Francesco Gavazzo, Riccardo Treglia, and Gabriele Vanoni.
  Monadic intersection types, relationally.
- Gordon D. Plotkin and John Power.
  Tensors of comodels and models for operational semantics.