Formally Optimal Boxing

Fritz Henglein and Jesper Jorgensen

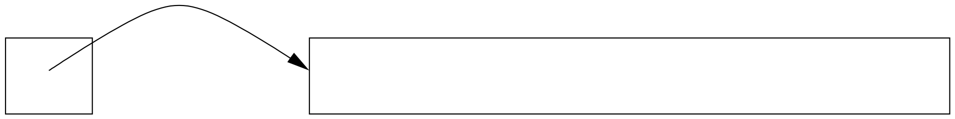
presenter : poisson

November 2, 2000

□ Introduction

- representation analysis : optimize the run-time representation of elements of data types.

- boxed representation :



- boxing analysis : minimize the need for run-time boxing/unboxing.

  - *coercion* : explicit boxing/unboxing.

  - *completion* : program translation by coercion.

□ Introduction

- completion coherence
  - have the same observational behavior
- Goal : find out an “optimal” completion
  - completion that is better than any other coherent completions

□ Overview

- some equality rules.
  - ex)
$$\begin{aligned} \langle \mathtt{unbox} \to \mathtt{box}\rangle \lambda x.e &= \lambda x.\langle \mathtt{box}\rangle([x \mapsto \langle \mathtt{unbox}\rangle x]e) \\ (\langle \mathtt{unbox} \to \mathtt{box}\rangle e_1)e_2 &= \langle \mathtt{box}\rangle(e_1(\langle \mathtt{unbox}\rangle e_2)) \end{aligned}$$

- rewriting rules
  - $\langle \mathtt{box}; \mathtt{unbox}\rangle e \Longrightarrow e$
    $\langle \mathtt{unbox}; \mathtt{box}\rangle e \Longrightarrow e$
  - $e' \Longrightarrow^* e''$: e'' is better than e'
  - if $e' \Longrightarrow e''$ for all e' then e'' is an "optimal" completion

□ Example

$$
\begin{aligned}
&(\langle \texttt{unbox} \to \texttt{box}\rangle(\lambda x.x+1)) \\
&\quad ((\langle \texttt{unbox} \to \texttt{box}\rangle(\lambda x.x+1))\langle \texttt{box}\rangle 1) \\
&= \\
&\langle \texttt{box}\rangle((\lambda x.x+1)\langle \texttt{unbox}\rangle\langle \texttt{box}\rangle((\lambda x.x+1)\langle \texttt{unbox}\rangle\langle \texttt{box}\rangle 1)) \\
&\Longrightarrow \\
&\langle \texttt{box}\rangle((\lambda x.x+1)((\lambda x.x+1)1))
\end{aligned}
$$

$$
\begin{aligned}
&(\langle \texttt{unbox} \to \texttt{box}\rangle(\lambda x.x+1)) \\
&\quad ((\langle \texttt{unbox} \to \texttt{box}\rangle(\lambda x.x+1))\langle \texttt{box}\rangle 1) \\
&= \\
&(\lambda x.\langle \texttt{box}\rangle(\langle \texttt{unbox}\rangle x+1)) \\
&\quad ((\lambda x.\langle \texttt{box}\rangle(\langle \texttt{unbox}\rangle x+1))\langle \texttt{box}\rangle 1)
\end{aligned}
$$

□ Example

$$(\langle \texttt{box} \to \texttt{unbox}\rangle(\lambda x. \cdots x \cdots))$$
$$\quad((\langle \texttt{box} \to \texttt{unbox}\rangle(\lambda x. \cdots x \cdots))1)$$
$$=$$
$$(\lambda x. \langle \texttt{unbox}\rangle(\cdots \langle \texttt{box}\rangle x \cdots))$$
$$\quad((\lambda x. \langle \texttt{unbox}\rangle(\cdots \langle \texttt{box}\rangle x \cdots))1)$$

$$(\langle \texttt{box} \to \texttt{unbox}\rangle(\lambda x. \cdots x \cdots))$$
$$\quad((\langle \texttt{box} \to \texttt{unbox}\rangle(\lambda x. \cdots x \cdots))1)$$
$$=$$
$$\langle \texttt{unbox}\rangle((\lambda x. \cdots x \cdots)\langle \texttt{box}\rangle\langle \texttt{unbox}\rangle((\lambda x. \cdots x \cdots)\langle \texttt{box}\rangle 1))$$
$$\Longrightarrow$$
$$\langle \texttt{unbox}\rangle((\lambda x. \cdots x \cdots)((\lambda x. \cdots x \cdots)\langle \texttt{box}\rangle 1))$$

□ Syntax of Core-XML

$$
\begin{array}{lcll}
e & ::= & x \mid \lambda x : \tau . e \mid e\ e \mid \texttt{let}\ x : \sigma = e\ \texttt{in}\ e & \mathit{Expression} \\
 & & \mid \Lambda \alpha . e \mid e\{\tau\} & \\
\tau & ::= & \alpha \mid \tau \to \tau & \mathit{Type} \\
\sigma & ::= & \forall \alpha . \sigma & \mathit{TypeScheme}
\end{array}
$$

□ Typing rules for Core-XML

$$\frac{\Gamma + x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \to \tau_2} \qquad \frac{\Gamma \vdash e_1 : \sigma \quad \Gamma + x : \sigma \vdash e_2 : \tau}{\Gamma \vdash \texttt{let } x : \sigma = e_1 \texttt{ in } e_2 : \tau}$$

$$\frac{\Gamma(x) = \sigma}{\Gamma \vdash x : \sigma} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 \ e_2 : \tau_2}$$

$$\frac{\Gamma \vdash e : \sigma}{\Gamma \vdash \Lambda \alpha . e : \forall \alpha . \sigma} \text{if} \alpha \notin FV(\Gamma) \qquad \frac{\Gamma \vdash e : \forall \alpha . \sigma}{\Gamma \vdash e\{\tau\} : [\alpha \mapsto \tau]\sigma}$$

□ Representation Types

$$
\begin{array}{rcll}
\rho & ::= & \upsilon \mid \pi & \mathit{RepType} \\
\upsilon & ::= & \alpha \mid \rho \rightarrow \rho & \mathit{UnboxedType} \\
\pi & ::= & \beta \mid [\upsilon] & \mathit{BoxedType} \\
\sigma & ::= & \rho \mid \forall \beta . \sigma & \mathit{PolyType}
\end{array}
$$

□ Coercion formation rules

$$\vdash \mathtt{box}_v : v \leadsto [v] \qquad \vdash \mathtt{unbox}_v : [v] \leadsto v$$

$$\vdash \iota_\rho : \rho \leadsto \rho \qquad \frac{\vdash c : v \leadsto v'}{\vdash [c] : [v] \leadsto [v']}$$

$$\frac{\vdash c : \rho_1 \leadsto \rho_1' \qquad \vdash c' : \rho_2 \leadsto \rho_2'}{\vdash c \to c' : \rho_1' \to \rho_2 \leadsto \rho_1 \to \rho_2'} \qquad \frac{\vdash c : \rho \leadsto \rho' \qquad \vdash c' : \rho' \leadsto \rho''}{\vdash c; c' : \rho \leadsto \rho''}$$

□ Typing rules for the explicitly boxed Core-XML

$$\frac{\Gamma + x : \rho_1 \vdash e : \rho_2}{\Gamma \vdash \lambda x : \rho_1 . e : \rho_1 \to \rho_2} \qquad \frac{\Gamma \vdash e_1 : \rho_1 \to \rho_2 \quad \Gamma \vdash e_2 : \rho_1}{\Gamma \vdash e_1\ e_2 : \rho_2}$$

$$\frac{\Gamma \vdash e_1 : \sigma \quad \Gamma + x : \sigma \vdash e_2 : \rho}{\Gamma \vdash \texttt{let}\ x : \sigma = e_1\ \texttt{in}\ e_2 : \rho} \qquad \frac{\Gamma(x) = \sigma}{\Gamma \vdash x : \sigma}$$

$$\frac{\Gamma \vdash e : \sigma}{\Gamma \vdash \Lambda\beta . e : \forall\beta.\sigma}\text{if}\beta \notin FV(\Gamma) \qquad \frac{\Gamma \vdash e : \forall\beta.\sigma}{\Gamma \vdash e\{\pi\} : [\beta \mapsto \pi]\sigma}$$

$$\frac{\Gamma \vdash e : \rho \quad \vdash c : \rho \leadsto \rho'}{\Gamma \vdash \langle c \rangle e : \rho'}$$

□ Equality rules for coercion

$$
\begin{array}{rcll}
(c;c');c'' & = & c;(c';c'') & (1)\\
c;\iota & = & c & (2)\\
\iota;c & = & c & (3)\\
(c_1 \to c_2);(c'_1 \to c'_2) & = & (c'_1;c_1) \to (c_2;c'_2) & (4)\\
\iota \to \iota & = & \iota & (5)\\
[c_1];[c_2] & = & [c_1;c_2] & (6)\\
[\iota] & = & \iota & (7)\\
c;\mathtt{box}_{v'} & = & \mathtt{box}_v;[c] & (8)\\
\mathtt{unbox}_v;c & = & [c];\mathtt{unbox}_{v'} & (9)\\
 & & & \\
\mathtt{box}_v;\mathtt{unbox}_v & = & \iota & (\phi)\\
\mathtt{unbox}_v;\mathtt{box}_v & = & \iota & (\psi)
\end{array}
$$

□ Coercion coherence

- Two coercion c and c' are *A-equal*, written $A \vdash c = c'$, if $c = c'$ is derivable from equality axioms A and the equations (1) - (9).

- Congruence

  – $c \cong c'$, if $c : \rho \leadsto \rho' \Leftrightarrow c'\rho \leadsto \rho'$

  – $e \cong e'$, if $e : \rho \Leftrightarrow e' : \rho$

- Coherence coercion

  – c and c' are congruent iff they are $\phi\psi$-equal;
  i.e., $c \cong c'$ iff $\phi\psi \vdash c = c'$

□ Coercion reduction

$$
\begin{array}{rcll}
(c; c'); c'' & = & c; (c'; c'') & \\
c; \iota & \Longrightarrow & c & \\
\iota; c & \Longrightarrow & c & \\
\iota \to \iota & \Longrightarrow & \iota & \\
(c_1 \to c_2); (c'_1 \to c'_2) & \Longrightarrow & (c'_1; c_1) \to (c_2; c'_2) & \\
[\iota] & \Longrightarrow & \iota & \\
[c_1]; [c_2] & \Longrightarrow & [c_1; c_2] & \\
\mathtt{box}; [c] & \Longrightarrow & c; \mathtt{box} & \\
[c]; \mathtt{unbox} & \Longrightarrow & \mathtt{unbox}; c & \\
 & & & \\
\mathtt{box}; \mathtt{unbox} & \Longrightarrow & \iota & (\phi^{\to}) \\
\mathtt{unbox}; \mathtt{box} & \Longrightarrow & \iota & (\psi^{\to}) \\
\mathtt{unbox}; c; \mathtt{box} & \Longrightarrow & [c] & (\psi^{\to})
\end{array}
$$

□ Optimal coercion

- A coercion c is *(formally) optimal* if all congruent coercions $c' = c$ can be reduced to c by $\phi\psi$-reduction

- The coercion reduction system in previous slide makes unique optimal coercion

□ Equality rules for expressions

$$
\begin{array}{rcll}
\langle\iota\rangle e & = & e & (10) \\
\langle c\rangle\langle d\rangle e & = & \langle d; c\rangle e & (11) \\
\\
\langle c \to d\rangle \lambda x.e & = & \lambda x.\langle d\rangle([x \mapsto \langle c\rangle x]e) & (12) \\
(\langle c \to d\rangle e)\ e' & = & \langle d\rangle(e(\langle c\rangle e')) & (13) \\
\langle d\rangle \texttt{let}\ x = \Lambda\overline{\beta}.\langle c\rangle e & = & \texttt{let}\ x = \Lambda\overline{\beta}.e & (14) \\
\texttt{in}\ [x^i \mapsto x\{\pi\}]e' & & \texttt{in}\ \langle d\rangle([x^i \mapsto \langle c\rangle x\{\pi\}]e') & \\
\langle c\rangle x\{\overline{\pi}\} & = & x\{\overline{\pi'}\} & (15)
\end{array}
$$

□ Expression coherence

- e and e' are *A-equal*, written $A \vdash e = e'$, if $e = e'$ is derivable from equality axioms A and the equations (1) - (11).

- Coherence of completions

  – e and e' are congruent iff they are $E\phi\psi$-equal;
  i.e., $e \cong e'$ iff $E\phi\psi \vdash e = e'$

  – E = (12) - (15)

□ Optimal completions

Unfortunately, $\phi\psi$-reduction on E-congruence classes is not Church-Rosser; that is, there are congruent completions that have no common reduct.

$$
\begin{aligned}
e_1 \equiv\ & \texttt{let}\ id : \forall\beta.\beta \to \beta = \Lambda\beta.\lambda y : \beta.y\ \texttt{in} \\
& \quad (\lambda x : \texttt{int}.x + \langle\texttt{unbox}\rangle(id\{[\texttt{int}]\}\langle\texttt{box}\rangle x)) \\
& \quad\quad (\texttt{if}\ \ldots\ \texttt{then}\ 2\ \texttt{else}\ \langle\texttt{unbox}\rangle(id\{[\texttt{int}]\}\langle\texttt{box}\rangle 5))
\end{aligned}
$$

$$
\begin{aligned}
e_2 \equiv\ & \texttt{let}\ id : \forall\beta.\beta \to \beta = \Lambda\beta.\lambda y : \beta.y\ \texttt{in} \\
& \quad (\lambda x : [\texttt{int}].\langle\texttt{unbox}\rangle x + \langle\texttt{unbox}\rangle(id\{[\texttt{int}]\}x)) \\
& \quad\quad (\texttt{if}\ \ldots\ \texttt{then}\ \langle\texttt{box}\rangle 2\ \texttt{else}\ (id\{[\texttt{int}]\}\langle\texttt{box}\rangle 5))
\end{aligned}
$$

We can reduce e_1 into e_2 by adding $\langle\texttt{unbox}; \texttt{box}\rangle$ before 2.

□ positive and negative coercions

$$\iota_\tau : + \quad \texttt{box}_v : + \quad \frac{c : - \quad d : +}{c \to d : +} \quad \frac{c : + \quad d : +}{c; d : +} \quad \frac{c : +}{[c] : +}$$

$$\iota_\tau : - \quad \texttt{unbox}_v : - \quad \frac{c : + \quad d : -}{c \to d : -} \quad \frac{c : - \quad d : -}{c; d : -} \quad \frac{c : -}{[c] : -}$$

$\langle c^+ \to d^+ \rangle$ can be factored into
$\langle d^+ \to \iota \rangle^- \langle \iota \to c^+ \rangle^+$ or $\langle \iota \to d^+ \rangle^+ \langle c^+ \to \iota \rangle^-$.

□ Polarized Equality (E_p) equations

$$
\begin{array}{rcll}
\langle c^{+} \to d^{-} \rangle \lambda x.e & = & \lambda x.\langle d^{-} \rangle ([x \mapsto \langle c^{+} \rangle x]e) & (12^{-}) \\
\lambda x.\langle d^{+} \rangle ([x \mapsto \langle c^{-} \rangle x]e) & = & \langle c^{-} \to d^{+} \rangle \lambda x.e & (12^{+}) \\
\\
\langle d^{-} \rangle (e(\langle c^{+} \rangle e')) & = & (\langle c^{+} \to d^{-} \rangle e)\ e' & (13^{-}) \\
(\langle c^{-} \to d^{+} \rangle e)\ e' & = & \langle d^{+} \rangle (e(\langle c^{-} \rangle e')) & (13^{+}) \\
\\
\langle d^{-} \rangle \texttt{let}\ x = \Lambda\overline{\beta}.\langle c^{-} \rangle e & = & \texttt{let}\ x = \Lambda\overline{\beta}.e & (14^{-}) \\
\texttt{in}\ [x^{i} \mapsto x\{\pi\}]e' & & \texttt{in}\ \langle d^{-} \rangle ([x^{i} \mapsto \langle c^{-} \rangle x\{\pi\}]e') & \\
\texttt{let}\ x = \Lambda\overline{\beta}.e & = & \langle d^{+} \rangle \texttt{let}\ x = \Lambda\overline{\beta}.\langle c^{+} \rangle e & (14^{+}) \\
\texttt{in}\ \langle d^{+} \rangle ([x^{i} \mapsto \langle c^{+} \rangle x\{\pi\}]e') & & \texttt{in}\ [x^{i} \mapsto x\{\pi\}]e' & \\
\\
\langle \widehat{\rho}(\overline{c}^{-}, \overline{\iota}) \rangle x\{\overline{\pi}\} & = & \langle \widehat{\rho}(\overline{\iota}, \overline{c}^{-}) \rangle x\{\overline{\pi}'\} & (15^{-}) \\
\langle \widehat{\rho}(\overline{\iota}, \overline{c}^{+}) \rangle x\{\overline{\pi}\} & = & \langle \widehat{\rho}(\overline{c}^{+}, \overline{\iota}) \rangle x\{\overline{\pi}'\} & (15^{+})
\end{array}
$$

□ ψ-free and ϕ-free completion

- a completion e is ψ-free if every congruent completion $e' \cong e$ ψ-reduces to e under $E\phi$-equality; i.e, $E\phi \vdash e' \Longrightarrow_{\psi}^{*} e$
  - keeping all data "maximally" boxed representation
  - $\psi^{\rightarrow} E_p^{\leftarrow}$ reduction

- a completion e is ϕ-free if every congruent completion $e' \cong e$ ϕ-reduces to e under $E\psi$-equality; i.e, $E\psi \vdash e' \Longrightarrow_{\phi}^{*} e$
  - keeping all data "minimally" boxed representation
  - $\phi^{\rightarrow} E_p^{\rightarrow}$ reduction

□ Optimal ψ-free/ϕ-free completion

- A completion e is a *(formally) optimal ψ-free completion* if e is ψ-free and every congruent ψ-free completion e' ϕ-reduces to e modulo E; i.e., $E \vdash e' \Longrightarrow_{\phi}^{*} e$

- A completion e is a *(formally) optimal ϕ-free completion* if e is ϕ-free and every congruent ϕ-free completion e' ψ-reduces to e modulo E; i.e., $E \vdash e' \Longrightarrow_{\psi}^{*} e$

Program	Completion	box	unbox
insert-sort	opt. ψ-free norm.	17	171
	opt. ϕ-free norm.	289	307
	Leroy	156	307
flip-list	opt. ψ-free norm.	20	20
	opt. ϕ-free norm.	30	35
	Leroy	20	20
leroy	opt. ψ-free norm.	709	709
	opt. ϕ-free norm.	446	446
	Leroy	1219	1219
poulsen	opt. ψ-free norm.	3	3
	opt. ϕ-free norm.	3	3
	Leroy	3	3
sieve	opt. ψ-free norm.	99	847
	opt. ϕ-free norm.	436	748
	Leroy	411	748
poly	opt. ψ-free norm.	100	150
	opt. ϕ-free norm.	100	150
	Leroy	620	670