# Flexible Representation Analysis

Zhong Shao
ICFP 1997

Sukyoung Ryu

November 22, 2000

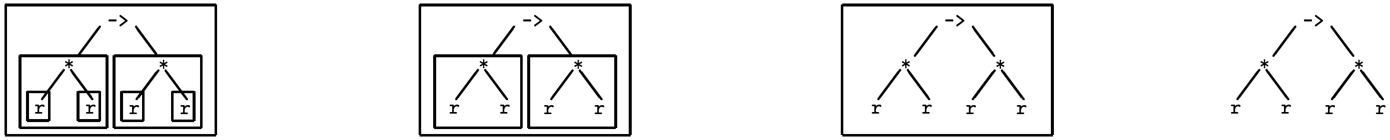
# Various Data Representations

**Fully Boxed Rep**
(data are recursively boxed at each layer)

**Partially Boxed Rep**
(e.g., function must box its arguments and results)

**Simply Boxed Rep**
(only the top-level data is boxed)

**Unboxed Rep**
(free form without any restrictions)




```
fun mv (x,y) = (x + 3.1, y + 2.7)
```

- Leroy's coercion-based approach
  does not handle recursive data types and mutable types well
- Harper and Morrisett's type-passing approach
  involves extensive runtime type analysis and code manipulations
- Shao's flexible representation
  the canonical boxed form

# Source language: SRC

$$
\begin{array}{llcl}
(\mathit{monotypes}) & \tau & ::= & t \mid \texttt{int} \mid \texttt{real} \mid \tau_1 * \tau_2 \mid \tau_1 \to \tau_2 \mid \tau\ \texttt{pack} \\
(\mathit{polytypes}) & \sigma & ::= & \tau \mid \forall t.\sigma \\
(\mathit{terms}) & e & ::= & x \mid i \mid f \mid \langle e_1, e_2 \rangle \mid \pi_1 e \mid \pi_2 e \mid e_1 e_2 \mid \lambda x.e \\
 & & \mid & \texttt{pk}\ e \mid \texttt{upk}\ e \mid \texttt{let}\ x = v\ \texttt{in}\ e \\
(\mathit{values}) & v & ::= & x \mid i \mid f \mid \langle v_1, v_2 \rangle \mid \lambda x.e \mid \texttt{pk}\ v
\end{array}
$$

$$
\frac{\triangle; \Gamma \vdash e : \tau}{\triangle; \Gamma \vdash \texttt{pk}\ e : \tau\ \texttt{pack}} \qquad \frac{\triangle; \Gamma \vdash e : \tau\ \texttt{pack}}{\triangle; \Gamma \vdash \texttt{upk}\ e : \tau}
$$

$$
\frac{e \hookrightarrow_s v}{\texttt{pk}\ e \hookrightarrow_s \texttt{pk}\ v} \qquad \frac{e \hookrightarrow_s \texttt{pk}\ v}{\texttt{upk}\ e \hookrightarrow_s v}
$$

- *incoercible tycons*

  type constructors whose values cannot be coerced because either it is too expensive or it violates the semantics

- `pack` : a representative for incoercible tycons

# Intermediate language: IL

$$
\begin{array}{llcl}
(\mathit{kinds}) & \kappa & ::= & \Omega \mid \kappa_1 \to \kappa_2 \\
(\mathit{con's}) & \mu & ::= & t \mid \texttt{Int} \mid \texttt{Real} \mid \texttt{Pack}(\mu) \mid \to (\mu_1, \mu_2) \mid \times(\mu_1, \mu_2) \\
 & & \mid & \lambda t :: \kappa.\mu \mid \mu_1[\mu_2] \\
(\mathit{types}) & \sigma & ::= & T(\mu) \mid \sigma_1 \to \sigma_2 \mid \sigma_1 \times \sigma_2 \mid \forall t :: \kappa.\sigma \\
(\mathit{terms}) & e & ::= & x \mid i \mid f \mid \texttt{Pk} \mid \texttt{Upk} \mid \langle e_1, e_2 \rangle \mid \pi_1 e \mid \pi_2 e \\
 & & \mid & \lambda x : \sigma.e \mid @e_1 e_2 \mid \Lambda t :: \kappa.e \mid e[\mu] \\
(\mathit{values}) & v & ::= & i \mid f \mid \texttt{Pk} \mid \texttt{Upk} \mid \texttt{PK}_\mu(v) \mid \langle v_1, v_2 \rangle \mid \lambda x : \sigma.e \mid \Lambda t :: \kappa.e
\end{array}
$$

$$\frac{}{\triangle; \Gamma \vdash \texttt{Pk} : \forall t :: \Omega. T(t) \to T(\texttt{Pack}(t))}$$

$$\frac{}{\triangle; \Gamma \vdash \texttt{Upk} : \forall t :: \Omega. T(\texttt{Pack}(t)) \to T(t)}$$

$$\frac{e \hookrightarrow_i v}{@(\texttt{Pk}[\mu])(e) \hookrightarrow_i \texttt{PK}_\mu(v)} \qquad \frac{e \hookrightarrow_i \texttt{PK}_\mu(v)}{@(\texttt{Upk}[\mu])(e) \hookrightarrow_i v}$$

# Translation from SRC to IL

$(const)$ $$\frac{}{\triangle;\Gamma \vdash i : \texttt{int} \Rightarrow i} \qquad \frac{}{\triangle;\Gamma \vdash f : \texttt{real} \Rightarrow f}$$

$(var)$ $$\frac{\Gamma(x) = \forall t_1 \ldots t_n.\tau \quad \rho = \{t_i \mapsto \tau_i \mid i = 1, \ldots, n\} \quad FTV(\rho(\tau)) \subseteq \triangle}{\triangle;\Gamma \vdash x : \rho(\tau) \Rightarrow x[\tau_1^m] \cdots [\tau_n^m]}$$

$(pk/upk)$ $$\frac{\triangle;\Gamma \vdash e : \tau \Rightarrow e'}{\triangle;\Gamma \vdash \texttt{pk}\ e : \tau\ \texttt{pack} \Rightarrow @(\texttt{Pk}[\tau^m])(e')} \qquad \frac{\triangle;\Gamma \vdash e : \tau\ \texttt{pack} \Rightarrow e'}{\triangle;\Gamma \vdash \texttt{upk}\ e : \tau \Rightarrow @(\texttt{Upk}[\tau^m])(e')}$$

$(pair/ith)$ $$\frac{\triangle;\Gamma \vdash e_1 : \tau_1 \Rightarrow e_1' \quad \triangle;\Gamma \vdash e_2 : \tau_2 \Rightarrow e_2'}{\triangle;\Gamma \vdash \langle e_1, e_2\rangle : \tau_1 * \tau_2 \Rightarrow \langle e_1', e_2'\rangle} \qquad \frac{\triangle;\Gamma \vdash e : \tau_1 * \tau_2 \Rightarrow e'}{\triangle;\Gamma \vdash \pi_i e : \tau_i \Rightarrow \pi_i e'} \quad (i = 1, 2)$$

$(fn/app)$ $$\frac{\triangle;\Gamma \uplus \{x : \tau_1\} \vdash e : \tau_2 \Rightarrow e'}{\triangle;\Gamma \vdash \lambda x.e : \tau_1 \to \tau_2 \Rightarrow \lambda x : \tau_1^p.e'} \qquad \frac{\triangle;\Gamma \vdash e_1 : \tau' \to \tau \Rightarrow e_1' \quad \triangle;\Gamma \vdash e_2 : \tau' \Rightarrow e_2'}{\triangle;\Gamma \vdash e_1 e_2 : \tau \Rightarrow @e_1' e_2'}$$

$(let)$ $$\frac{\triangle \uplus \{t_1, \ldots, t_n\};\Gamma \vdash v : \tau_0 \Rightarrow e_1' \quad \sigma_0 = \forall t_1, \ldots, t_n.\tau_0 \quad \triangle;\Gamma \uplus \{x : \sigma_0\} \vdash e : \tau \Rightarrow e_2'}{\triangle;\Gamma \vdash \texttt{let}\ x = v\ \texttt{in}\ e : \tau \Rightarrow @(\lambda x : \sigma_0^p.e_2')(\Lambda t_1 :: \Omega, \ldots, \Lambda t_n :: \Omega.e_1')}$$

$$\begin{aligned}
t^m &= t \\
\texttt{int}^m &= \texttt{Int} \\
\texttt{real}^m &= \texttt{Real} \\
(\tau_1 * \tau_2)^m &= \times(\tau_1^m, \tau_2^m) \\
(\tau_1 \to \tau_2)^m &= \to(\tau_1^m, \tau_2^m) \\
(\tau\ \texttt{pack})^m &= \texttt{Pack}(\tau^m)
\end{aligned}
\qquad
\begin{aligned}
\tau^p &= T(\tau^m) \\
(\forall t.\sigma)^p &= \forall t :: \Omega.\sigma^p
\end{aligned}$$

# Target language: TGT

$$
\begin{array}{llll}
(\mathit{kinds}) & \kappa & ::= & ... \\
(\mathit{con's}) & \mu & ::= & ... \mid \texttt{Boxed}(\mu) \mid \texttt{Typerec}\,\mu\,\texttt{of}\,(\mu_i|\mu_r|\mu_\rightarrow|\mu_\times|\mu_p|\mu_b) \\
(\mathit{types}) & \sigma & ::= & ... \\
(\mathit{terms}) & e & ::= & ... \mid \texttt{box} \mid \texttt{unbox} \mid \texttt{typerec}\,\mu\,\texttt{of}\,[t.\sigma](e_i|e_r|e_\rightarrow|e_\times|e_p|e_b) \\
(\mathit{values}) & v & ::= & ... \mid \texttt{box} \mid \texttt{unbox} \mid \texttt{BX}_\mu(v)
\end{array}
$$

- `Typerec`, `typerec` : for intensional type analysis
- `Boxed`, `box`, `unbox` : for boxing

$$
\begin{array}{l}
\texttt{box} : \forall t :: \Omega. T(t) \rightarrow T(\texttt{Boxed}(t)) \\
\texttt{unbox} : \forall t :: \Omega. T(\texttt{Boxed}(t)) \rightarrow T(t)
\end{array}
$$

- `wrap`, `unwrap` : for coercion

$$
\begin{array}{l}
\texttt{wrap} : \forall t :: \Omega. T(t) \rightarrow T(\texttt{Wrap}[t]) \\
\texttt{unwrap} : \forall t :: \Omega. T(\texttt{Wrap}[t]) \rightarrow T(t)
\end{array}
$$

# Translation from IL to TGT

(*base*) $$\frac{}{\triangle;\Gamma \vdash x : \Gamma(x) \Rightarrow x} \qquad \frac{}{\triangle;\Gamma \vdash c : \_ \Rightarrow c} \quad (c = i, f, \mathtt{Pk}, \mathtt{Upk})$$

(*pa/pi*) $$\frac{\triangle;\Gamma \vdash e_1 : \sigma_1 \Rightarrow e_1' \quad \triangle;\Gamma \vdash e_2 : \sigma_2 \Rightarrow e_2'}{\triangle;\Gamma \vdash \langle e_1, e_2\rangle : \sigma_1 \times \sigma_2 \Rightarrow \langle e_1', e_2'\rangle} \qquad \frac{\triangle;\Gamma \vdash e : \sigma_1 \times \sigma_2 \Rightarrow e'}{\triangle;\Gamma \vdash \pi_i e : \sigma_i \Rightarrow \pi_i e'} \quad (i = 1, 2)$$

(*fn/ap*) $$\frac{\triangle;\Gamma \uplus \{x : \sigma_1\} \vdash e : \sigma_2 \Rightarrow e'}{\triangle;\Gamma \vdash \lambda x : \sigma_1 . e : \sigma_1 \to \sigma_2 \Rightarrow \lambda x : \sigma_1^u . e'} \qquad \frac{\triangle;\Gamma \vdash e_1 : \sigma' \to \sigma \Rightarrow e_1' \quad \triangle;\Gamma \vdash e_2 : \sigma' \Rightarrow e_2'}{\triangle;\Gamma \vdash @e_1 e_2 : \sigma \Rightarrow @e_1' e_2'}$$

(*tfn*) $$\frac{\triangle \uplus \{t :: \kappa\};\Gamma \vdash e : \sigma \Rightarrow e'}{\triangle;\Gamma \vdash \Lambda t :: \kappa . e : \forall t :: \kappa . \sigma \Rightarrow \Lambda t : \kappa . e'}$$

(*tapp*) $$\frac{\triangle;\Gamma \vdash e : \forall t :: \kappa . \sigma \Rightarrow e' \quad e_c' = \mathcal{C}_{\mathcal{C}}([\mu^w/t]\sigma^u, [\mu^u/t]\sigma^u)}{\triangle;\Gamma \vdash e[\mu] : [\mu/t]\sigma \Rightarrow @e_c'(e'[\mu^w])}$$

$$\begin{array}{rcl}
t^u & = & t \\
\mathtt{Int}^u & = & \mathtt{Int} \\
\mathtt{Real}^u & = & \mathtt{Real} \\
(\times(\mu_1, \mu_2))^u & = & \times(\mu_1^u, \mu_2^u) \\
(\to(\mu_1, \mu_2))^u & = & \to(\mu_1^u, \mu_2^u) \\
(\mathtt{Pack}(\mu))^u & = & \mathtt{Pack}(\mathtt{Wrap}[\mu]) \\
(\Lambda t :: \kappa . \mu)^u & = & \Lambda t :: \kappa . \mu^u \\
(\mu_1[\mu_2])^u & = & \mu_1^u[\mu_2^u]
\end{array}
\qquad
\begin{array}{rcll}
T(\mu)^u & = & T(\mu^u) & \\
(\forall t :: \kappa . \sigma)^u & = & \forall t :: \kappa . \sigma^u & \\
(\sigma_1 \times \sigma_2)^u & = & \sigma_1^u \times \sigma_2^u & \\
(\sigma_1 \to \sigma_2)^u & = & \sigma_1^u \to \sigma_2^u & \\
 & & & \\
\mu^w & = & \mathtt{Wrap}[\mu^u] & \text{if } \triangle \rhd \mu :: \Omega \\
(\Lambda t :: \kappa . \mu)^w & = & \Lambda t :: \kappa . \mu^w & \\
\mu^w & = & (\Lambda t :: \kappa . \mu[t])^w & \text{if } \triangle \rhd \mu :: \kappa \to \kappa'
\end{array}$$

## The Coercion Generator

$$
\begin{array}{rcl}
\mathcal{C}_{\mathcal{C}}(\times(\mu_1', \mu_2'), \times(\mu_1, \mu_2)) & = & \mathcal{C}_{\mathcal{T}}(\,T(\mu_1') \times T(\mu_2'), T(\mu_1) \times T(\mu_2)) \\
\mathcal{C}_{\mathcal{C}}(\rightarrow (\mu_1', \mu_2'), \rightarrow (\mu_1, \mu_2)) & = & \mathcal{C}_{\mathcal{T}}(\,T(\mu_1') \rightarrow T(\mu_2'), T(\mu_1) \rightarrow T(\mu_2)) \\
\mathcal{C}_{\mathcal{C}}(\mu', \mu) & = & \texttt{identity}[\mu] \quad \text{if } \mu \equiv \mu' \text{ and } \triangle \triangleright \mu :: \Omega \\
\mathcal{C}_{\mathcal{C}}(\texttt{Wrap}[\mu], \mu) & = & \texttt{unwrap}[\mu] \\
\mathcal{C}_{\mathcal{C}}(\mu, \texttt{Wrap}[\mu]) & = & \texttt{wrap}[\mu]
\end{array}
$$

$$
\begin{array}{rcl}
\mathcal{C}_{\mathcal{T}}(\,T(\mu'), T(\mu)) & = & \mathcal{C}_{\mathcal{C}}(\mu', \mu) \\
\mathcal{C}_{\mathcal{T}}(\forall t :: \kappa.\sigma', \forall t :: \kappa.\sigma) & = & \lambda f : \forall t :: \kappa.\sigma'.(\Lambda t :: \kappa.@(\mathcal{C}_{\mathcal{T}}(\sigma', \sigma))(f[t])) \\
\mathcal{C}_{\mathcal{T}}(\sigma_1' \times \sigma_2', \sigma_1 \times \sigma_2) & = & \lambda x : \sigma_1' \times \sigma_2'.\ \langle @(\mathcal{C}_{\mathcal{T}}(\sigma_1', \sigma_1))(\pi_1 x), @(\mathcal{C}_{\mathcal{T}}(\sigma_2', \sigma_2))(\pi_2 x)\rangle \\
\mathcal{C}_{\mathcal{T}}(\sigma_1' \rightarrow \sigma_2', \sigma_1 \rightarrow \sigma_2) & = & \lambda f : \sigma_1' \rightarrow \sigma_2'.\lambda x : \sigma_1.\ @(\mathcal{C}_{\mathcal{T}}(\sigma_2', \sigma_2))(@f\ (@(\mathcal{C}_{\mathcal{T}}(\sigma_1, \sigma_1'))\ x))
\end{array}
$$

# $\boxed{\texttt{Wrap}}$

- Simple Boxing

$$\begin{aligned}
\texttt{Wrap}[\mu] &= \texttt{Boxed}(\texttt{Uncv}[\mu]) \\
\\
\texttt{Uncv}[\texttt{Int}] &= \texttt{Int} \\
\texttt{Uncv}[\texttt{Real}] &= \texttt{Real} \\
\texttt{Uncv}[\texttt{Pack}(\mu)] &= \texttt{Pack}(\mu) \\
\texttt{Uncv}[\texttt{Boxed}(\mu)] &= \mu \\
\texttt{Uncv}[\times(\mu_1, \mu_2)] &= \times(\texttt{Uncv}[\mu_1], \texttt{Uncv}[\mu_2]) \\
\texttt{Uncv}[\rightarrow(\mu_1, \mu_2)] &= \rightarrow(\texttt{Uncv}[\mu_1], \texttt{Uncv}[\mu_2])
\end{aligned}$$

- Partial Boxing

$$\texttt{Uncv}[\rightarrow(\mu_1, \mu_2)] = \rightarrow(\texttt{Boxed}(\texttt{Uncv}[\mu_1]), \texttt{Boxed}(\texttt{Uncv}[\mu_2]))$$

- Full Boxing

$$\texttt{Uncv}[\times(\mu_1, \mu_2)] = \times(\texttt{Boxed}(\texttt{Uncv}[\mu_1]), \texttt{Boxed}(\texttt{Uncv}[\mu_2]))$$

## $\boxed{\texttt{wrap}, \texttt{unwrap}}$

- Simple Boxing

$$
\begin{array}{lrcl}
(1) & \texttt{wrap}[\mu] & = & \texttt{box}[\texttt{Uncv}[\mu]] \circ \texttt{uncv}[\mu] \\
(2) & \texttt{unwrap}[\mu] & = & \texttt{cover}[\mu] \circ \texttt{unbox}[\texttt{Uncv}[\mu]] \\
 & & & \\
(3) & \texttt{uncv}[\texttt{Boxed}(\mu)] & = & \texttt{unbox}[\mu] \\
(4) & \texttt{uncv}[\times(\mu_1, \mu_2)] & = & \texttt{uncv}[\mu_1] \times \texttt{uncv}[\mu_2] \\
(5) & \texttt{uncv}[\rightarrow (\mu_1, \mu_2)] & = & \texttt{cover}[\mu_1] \rightarrow \texttt{uncv}[\mu_2] \\
(6) & \texttt{uncv}[\mu] & = & \texttt{identity}[\mu] \\
 & & & \\
(7) & \texttt{cover}[\texttt{Boxed}(\mu)] & = & \texttt{box}[\mu] \\
(8) & \texttt{cover}[\times(\mu_1, \mu_2)] & = & \texttt{cover}[\mu_1] \times \texttt{cover}[\mu_2] \\
(9) & \texttt{cover}[\rightarrow (\mu_1, \mu_2)] & = & \texttt{uncv}[\mu_1] \rightarrow \texttt{cover}[\mu_2] \\
(10) & \texttt{cover}[\mu] & = & \texttt{identity}[\mu]
\end{array}
$$

## Type Preservation

**Definition (valid canonical boxed form)**
If `Wrap` commutes with substitution and idempotents through all coercible tycon:

$$\begin{array}{rcl}
\texttt{Wrap}[\mu] & \equiv & \texttt{Wrap}[\texttt{Wrap}[\mu]] \\
\texttt{Wrap}[\times(\mu_1, \mu_2)] & \equiv & \texttt{Wrap}[\times(\texttt{Wrap}[\mu_1], \texttt{Wrap}[\mu_2])] \\
\texttt{Wrap}[\rightarrow (\mu_1, \mu_2)] & \equiv & \texttt{Wrap}[\rightarrow (\texttt{Wrap}[\mu_1], \texttt{Wrap}[\mu_2])]
\end{array}$$

**Theorem (type preservation)**
If `Wrap` is in the valid canonical boxed form and $\triangle; \Gamma \vdash e_i : \sigma \Rightarrow e_t$, then $\triangle^u; \Gamma^u \vdash e_t : \sigma^u$.

## Performance Measurements (Execution Time)

| Bmark | **Old** total (gc) | **New** total (gc) | **Ratio** |
|---|---|---|---|
| **boyer** | 0.97 (0.01) | 0.97 (0.00) | 1.00 |
| **fft** | 15.8 (2.96) | 3.56 (0.00) | 0.23 |
| **kb-comp** | 3.92 (0.33) | 4.10 (0.36) | 1.05 |
| **lexgen** | 5.20 (0.11) | 3.80 (0.10) | 0.73 |
| **life** | 0.63 (0.00) | 0.63 (0.00) | 1.00 |
| **mbrot** | 4.89 (0.00) | 3.53 (0.00) | 0.72 |
| **mlyacc** | 1.87 (0.14) | 1.89 (0.10) | 1.01 |
| **nucleic** | 3.18 (0.00) | 2.09 (0.00) | 0.66 |
| **sieve** | 0.46 (0.00) | 0.46 (0.00) | 1.00 |
| **simple** | 14.6 (1.59) | 10.7 (0.00) | 0.73 |
| **unzipr** | 3.03 (0.02) | 6.07 (0.17) | 2.00 |
| **vliw** | 8.87 (0.06) | 9.87 (0.06) | 1.11 |