Persistance and Type Abstraction

Luca Cardelli

David MacQueen

Type expressions

$$\sigma ::= \text{int} \mid \text{bool} \mid t \mid \sigma \times \sigma \mid \sigma + \sigma \mid \sigma \to \sigma \mid \forall t.\sigma$$

Polymorphism

$$\lambda x.\, x$$

$$\lambda x : \text{int}.\, x \qquad : \text{int} \to \text{int}$$

$$\lambda x : \text{bool}.\, x \qquad : \text{bool} \to \text{bool}$$

$$\text{Id} = \Lambda t.\, \lambda x : t.\, x \qquad : \forall t.\, t \to t$$

$$\text{Id}[\text{int}] = \lambda x : \text{int}.\, x \quad : \text{int} \to \text{int}$$

$$\text{Id}[\text{bool}] = \lambda x : \text{bool}.\, x \quad : \text{bool} \to \text{bool}$$

Existential Types and Packages

$\sigma ::= \exists t.\sigma(t)$ "signature"

$$\underline{\text{pack}}\ [t=\tau\,;\ e:\sigma(t)] \quad : \quad \exists t.\sigma(t)$$

$$\text{if} \quad e:\sigma(\tau)$$

$$\underline{\text{open}}\ A\ \underline{\text{as}}\ t, x\ \underline{\text{in}}\ e \quad : \quad \rho$$

$$\text{if} \quad A : \exists t.\sigma(t)$$

$$\text{and} \quad x:\sigma(t) \quad \Rightarrow \quad e:\rho$$

Example

$$
\begin{aligned}
I7 = \underline{\text{pack}}\ [\,t = \text{int};\ & \\
& \lambda(n,m).\,((n-m) \bmod 7 = 0) : t \times t \to \text{bool}\,] \\
& \quad : \exists t.\ t \times t \to \text{bool}
\end{aligned}
$$

$\underline{\text{open}}\ I7\ \underline{\text{as}}\ T, eq\ \underline{\text{in}}\ \ldots\ (\text{if}\ \boxed{eq(3,13)}\ \text{then} \ldots)\ \ldots$

Packages and Abstraction

$$A = \mathrm{pack}[t = \tau ; e : \sigma(t)] \quad : \exists t . \sigma(t)$$

$$\underline{\mathrm{open}}\ A\ \ \underline{\mathrm{as}}\ \ s, x\ \ \underline{\mathrm{in}}\ \ e' \quad : \rho$$

(1) transparent witness:

$s \equiv \tau$

s may appear in ρ (as τ)

(2) hypothetical witness:

s <u>opaque</u> (e.g. $s \not\equiv \tau$)

s may not appear in ρ

(3) abstract witness:

s opaque

s may appear in ρ (abbrev. $\mathrm{witness}(A)$)

Dynamics

$\underline{\text{dynamic}}(\tau, e) \quad : \text{dynamic} \qquad (\sim \exists t.\, t)$

$\underline{\text{coerce}}\ e\ \underline{\text{to}}\ \tau \quad : \tau$

---

value a = dynamic(Int, 3) $\langle$Int, 3$\rangle$

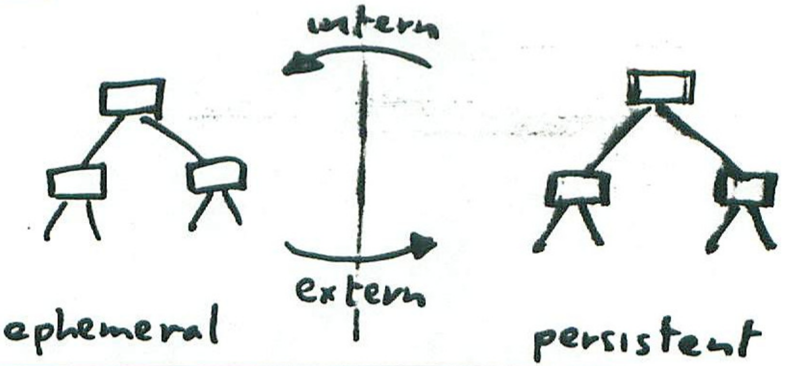
value f = dynamic(Int → Int, succ) $\langle$Int → Int, {closure

coerce a to Int ⇒ 3

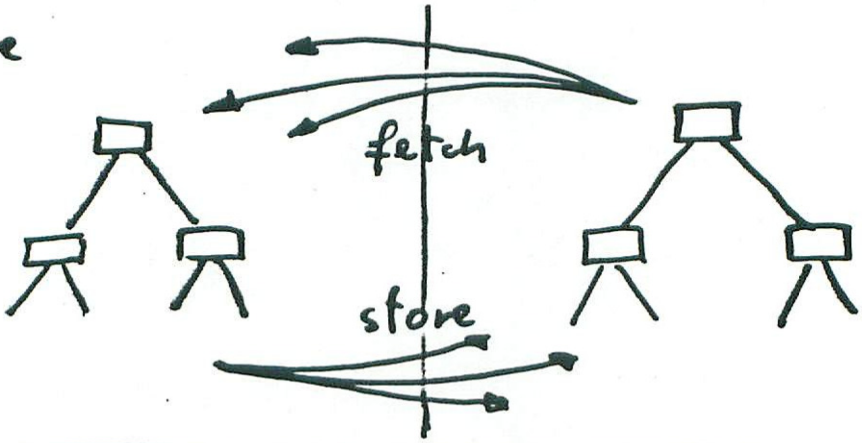
(coerce f to Int → Int)(3) ⇒ 4

coerce a to Int → Int ⇒

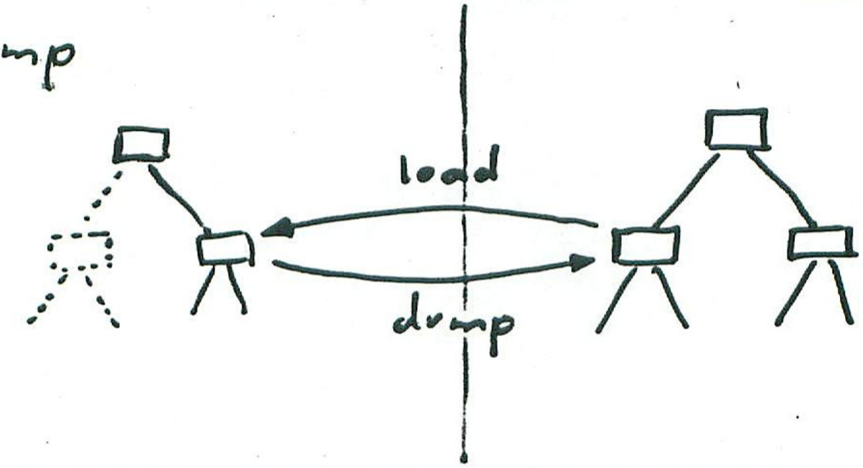
Intern/Extern



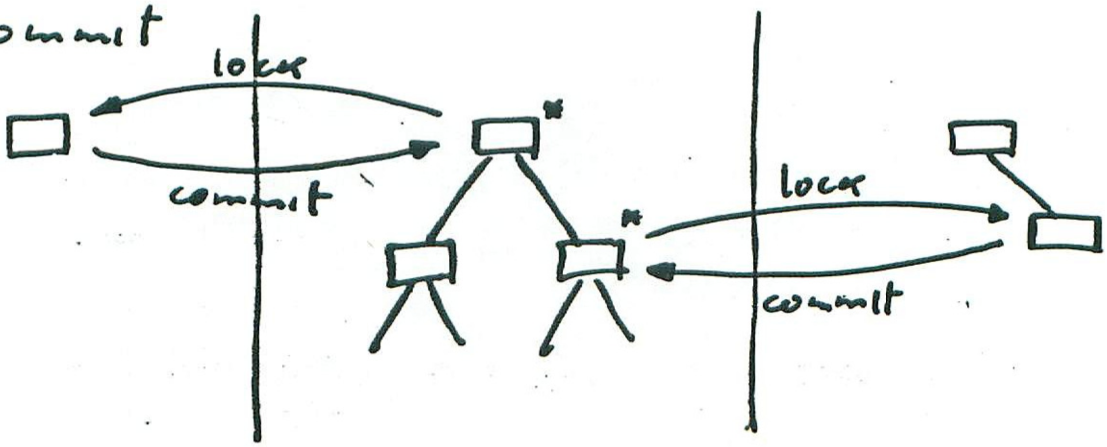
Fetch-Store



Load-Dump



Lock-Commit



```
signature S = ∃t. t × (t → int)
abstraction A = pack[t=int, (3,succ): t×(t→int)] : S
extern("abstraction", dynamic(S,A))
value x = open A as t,p in dynamic(t, fst(p)) : dynam
extern("object", x)
```

```
signature S = ∃t. t × (t → int)
abstraction A = coerce intern("abstraction") to S
open A as t,p
  in let x = coerce intern("object") to t
     in ... snd(p)(x) ...
```