Existential Types

- Provide a form of first-class information hiding (abstract data types)
- Example:

p = **pack** {Nat, (5, λx:Nat. succ x)}
as $\exists\alpha.\ \alpha \times (\alpha \to \alpha)$
open p **as** {β,x} **in** ((snd x) (fst x))

Existential Types

Syntax:

$$\begin{array}{lcl} T & ::= & \ldots \mid \exists \alpha.\ T \\ e & ::= & \mathit{pack}\ \{T, e\}\ \mathit{as}\ T \mid \mathit{open}\ e\ \mathit{as}\ \{\alpha, x\}\ \mathit{in}\ e \end{array}$$

Evaluation contexts:

$$E ::= \ldots \mid \mathit{pack}\ \{T, E\}\ \mathit{as}\ T \mid \mathit{open}\ E\ \mathit{as}\ \{\alpha, x\}\ \mathit{in}\ e$$

Reduction rule:

$$\mathit{open}\ (\mathit{pack}\ \{T_1, v\}\ \mathit{as}\ T_2)\ \mathit{as}\ \{\alpha, x\}\ \mathit{in}\ e \longrightarrow [\alpha \mapsto T_1][x \mapsto v]e$$

Type rules:

$$\frac{\Gamma \vdash e : [\alpha \mapsto U]T}{\Gamma \vdash \mathit{pack}\ \{U, e\}\ \mathit{as}\ \exists \alpha.T} \qquad \frac{\Gamma \vdash e_1 : \exists \alpha.\ T_1 \qquad \Gamma, \alpha, x : T_1 \vdash e_2 : T_2}{\Gamma \vdash \mathit{open}\ e_1\ \mathit{as}\ \{\alpha, x\}\ \mathit{in}\ e_2 : T_2}$$

Encoding Existentials with Universals

$$\exists\alpha.\ T \equiv \forall\beta.\ (\forall\alpha.T \to \beta) \to \beta$$

$$\textit{pack}\ \{T_1, e\}\ \textit{as}\ \exists\alpha.T_2 \equiv \Lambda\beta.\ \lambda f : (\forall\alpha.\ T_2 \to \beta).(f\,[T_1]\ e)$$

$$\textit{open}\ e_1\ \textit{as}\ \{\alpha, x\}\ \textit{in}\ e_2 \equiv (e_1[T_2]\ (\Lambda\alpha.\ \lambda x : T_1.\ e_2))$$

where $e_2 : T_2$ and $e_1 : \exists\alpha.\ T_1$

Using Existentials to Model Objects

Counter $= \exists\alpha.\{\mathit{state} : \alpha, \mathit{methods} : \{\mathit{get} : \alpha \to \mathit{Nat}, \mathit{inc} : \alpha \to \alpha\}\}$
c $= \{$Nat, $\{$ state $= 5$,
methods $= \{$get $= \lambda$x:Nat. x, inc $= \lambda$x:Nat. succ x$\}\}$
as Counter
open c **as** $\{\alpha$,body$\}$ **in** body.methods.get(body.state)
$>$ 5 : Nat

sendinc $= \lambda$c:Counter.
open c **as** $\{\alpha$,body$\}$ **in**
$\{$ α, $\{$ state $=$ body.methods.inc(body.state),
methods $=$ body.methods $\}\}$ **as** Counter
$>$ sendinc : Counter $\to$ Counter