In implementations of ML, Scheme, or other functional languages with static scoping, functions $\lambda x.\, e$ are paired with the environments ρ in which they were defined. The pair $\langle \lambda x.\, e,\ \rho \rangle$ is called a *closure*. The environment ρ tells how to evaluate the free variables of $\lambda x.\, e$. Usually the environment is a partial function from variables to values that is defined on (at least) all the free variables of $\lambda x.\, e$. It is obtained from the syntactic context in which $\lambda x.\, e$ appears.

When a function $\lambda x.\, e$ is called on some argument a, the argument is bound to the formal parameter x, and this new binding is appended to the environment ρ in the closure, then the body is evaluated in that environment.

The process of converting a function to take an environment as an extra argument is called *closure conversion*. In this lecture we will see why this works by proving a theorem that shows how closures adequately represent static scoping.

Closures and Environments

Formally, a *value* of λ-CBV is a closed CBV-irreducible λ-term, which is just a closed term of the form $\lambda x.\, e$. Let **Val** denote the set of λ-CBV values, and let **L** denote the set of all closed λ-terms (not necessarily CBV-irreducible).

Now we define a new language whose values are closures $\langle \lambda x.\, e,\ \rho \rangle$, where $\lambda x.\, e$ is permitted to have free variables, provided they are all in the domain of ρ. Because the definitions of closures and environments depend on each other, we must define them by mutual induction. We denote the set of closures and the set of environments by **Cl** and **Env**, respectively. They are defined to be the smallest sets such that

$$\begin{aligned} \mathbf{Env} &= \{\text{partial functions } \rho : \mathbf{Var} \to \mathbf{Cl} \text{ with finite domain}\}, \\ \mathbf{Cl} &= \{\langle \lambda x.\, e,\ \rho \rangle \mid \rho \in \mathbf{Env},\ FV(\lambda x.\, e) \subseteq \operatorname{dom} \rho\}. \end{aligned}$$

This definition may seem circular, but actually it does have a basis. We have $\bot \in \mathbf{Env}$, where $\bot$ is the null environment with domain $\varnothing$, therefore $\langle \lambda x.\, e,\ \bot \rangle \in \mathbf{Cl}$, where $\lambda x.\, e$ is closed. Once we know that **Cl** and **Env** are nonempty, we can form some nonnull environments $\rho : \mathbf{Var} \to \mathbf{Cl}$ and closures $\langle \lambda x.\, e,\ \rho \rangle$ where $\lambda x.\, e$ is not closed, provided ρ is defined on all free variables of $\lambda x.\, e$. This allows us to form even more closures and environments, and so on. After countably many steps, we reach a fixpoint.

Permitting environments to be partial functions is essential, but the restriction to finite domain is not. We need to allow partial functions so that the induction will get off the ground, i.e. $\mathbf{Cl} \neq \varnothing$. The restriction to finite domain makes the monotone map in the definition finitary, which ensures that the construction closes at ω. Without this restriction we would have to use transfinite induction.

More concretely, define

$$\begin{aligned} \mathbf{Cl}_0 &\triangleq \varnothing \\ \mathbf{Env}_n &\triangleq \{\text{partial functions } \rho : \mathbf{Var} \to \mathbf{Cl}_n \text{ with finite domain}\} \\ \mathbf{Cl}_{n+1} &\triangleq \{\langle \lambda x.\, e,\ \rho \rangle \mid \rho \in \mathbf{Env}_n,\ FV(\lambda x.\, e) \subseteq \operatorname{dom} \rho\} \\ \mathbf{Env} &\triangleq \bigcup_{n \geq 0} \mathbf{Env}_n \\ \mathbf{Cl} &\triangleq \bigcup_{n \geq 0} \mathbf{Cl}_n. \end{aligned}$$

Note that

$$\begin{aligned} \mathbf{Env}_0 &= \{\bot\} \\ \mathbf{Cl}_1 &= \{\langle \lambda x.\, e,\ \bot \rangle \mid \lambda x.\, e \text{ is closed}\}. \end{aligned}$$

Iterated Substitution

Let $\mathbf{C}$ denote the set of pairs $\langle e,\, \rho\rangle$, where e is any λ-term (not necessarily closed or CBV-irreducible) and ρ an environment such that $FV(e) \subseteq \operatorname{dom}\rho$. Every such pair gives rise to a closed λ-term obtained by "iterated substitution". This is given by a map $F : \mathbf{C} \to \mathbf{L}$ defined inductively as follows:

$$F(\langle e,\, \rho\rangle) \;\stackrel{\triangle}{=}\; e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho\}.$$

Again, this may seem like a circular definition, but it's not.

Lemma F is well-defined on $\mathbf{C}$ and takes values in $\mathbf{L}$. On inputs in $\mathbf{Cl}$, F takes values in $\mathbf{Val}$.

Proof. Induction on the stage of definition of $\rho \in \mathbf{Env}$. Consider first $\langle e,\, \rho\rangle \in \mathbf{C}$ with $\rho \in \mathbf{Env}_0$. Then $\rho = \bot$, e is closed, and

$$F(\langle e,\, \bot\rangle) \;\stackrel{\triangle}{=}\; e\{F(\bot(y))/y,\ y \in \operatorname{dom}\bot\} \;=\; e.$$

If $\rho \in \mathbf{Env}_{n+1}$, then ρ takes values in $\mathbf{Cl}_{n+1}$. Then for all $y \in \operatorname{dom}\rho$, $\rho(y) = \langle \lambda x.\, d,\, \sigma\rangle$ for some $\sigma \in \mathbf{Env}_n$, $FV(\lambda x.\, d) \subseteq \operatorname{dom}\sigma$. By the induction hypothesis, $F(\rho(y)) \in \mathbf{L}$. Then

$$F(\langle e,\, \rho\rangle) \;=\; e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho\} \;\in\; \mathbf{L}.$$

F takes values in $\mathbf{Val}$ on inputs in $\mathbf{Cl}$, because

$$F(\langle \lambda x.\, e,\, \rho\rangle) \;=\; (\lambda x.\, e)\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho\},$$

which is closed and CBV-irreducible, thus a value of λ-CBV. □

λ-Cl

The terms of λ-Cl consist of the elements of $\mathbf{C}$. The values of λ-Cl are elements of $\mathbf{Cl}$. We now give a set of evaluation rules defining a big-step SOS for a binary relation $\Downarrow \subseteq \mathbf{C} \times \mathbf{Cl}$. The statement $\langle e,\, \rho\rangle \Downarrow v$ should be interpreted as: When e is evaluated in the environment ρ, the result is v. Given this informal meaning, these rules below reflect the usual evaluation rules for functional expressions in Scheme or ML.

$$\langle x,\, \sigma\rangle \Downarrow \sigma(x) \qquad \langle \lambda x.\, e,\, \rho\rangle \Downarrow \langle \lambda x.\, e,\, \rho\rangle \qquad \frac{\langle e_1,\, \sigma\rangle \Downarrow \langle \lambda x.\, e,\, \tau\rangle, \quad \langle e_2,\, \sigma\rangle \Downarrow u, \quad \langle e,\, \tau[u/x]\rangle \Downarrow v}{\langle e_1\, e_2,\, \sigma\rangle \Downarrow v}.$$

Note that the rule for λ-abstractions is just the identity relation! This rule says that evaluating $\lambda x.\, e$ in the environment ρ results in the closure $\langle \lambda x.\, e,\, \rho\rangle$.

The third rule is the usual rule for evaluation of a function application: to evaluate $e_1\, e_2$ in the environment σ, first evaluate the function e_1 in environment σ to get a closure $\langle \lambda x.\, e,\, \tau\rangle$, then evaluate the argument e_2 in environment σ, then bind the value of the argument to the formal parameter x in the environment of the closure τ and evaluate the body of the function in that environment.

By the lemma above, the "iterated substitution" map F translates λ-Cl expressions to λ-CBV expressions and λ-Cl values to λ-CBV values. The following theorem asserts adequacy of this translation.

Theorem $F(\langle e,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} v \;\Leftrightarrow\; \exists w\ \langle e,\, \sigma\rangle \Downarrow w \wedge v = F(w)$.

Proof. ($\Leftarrow$) We wish to show that if $\langle e,\, \sigma\rangle \Downarrow w$, then $F(\langle e,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} F(w)$. The proof is by induction on the derivation $\langle e,\, \sigma\rangle \Downarrow w$.

For the case $e = x$, we have $\langle x,\, \sigma\rangle \Downarrow \sigma(x)$. In this case $F(\langle x,\, \sigma\rangle) = x\{F(\sigma(y))/y,\ y \in \operatorname{dom}\sigma\} = F(\sigma(x))$, so $F(\langle x,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} F(\sigma(x))$.

For the case $\lambda x.\, e$, we have $\langle \lambda x.\, e,\, \sigma\rangle \Downarrow \langle \lambda x.\, e,\, \sigma\rangle$. In this case $F(\langle x,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} F(\langle x,\, \sigma\rangle)$ trivially.

Finally, for the case $e_1\, e_2$, if $\langle e_1\, e_2,\, \sigma\rangle \Downarrow w$, for some $\lambda x.\, d$, τ, and u, we must have

- $\langle e_1,\, \sigma\rangle \Downarrow \langle \lambda x.\, d,\, \tau\rangle$;
- $\langle e_2,\, \sigma\rangle \Downarrow u$;
- $\langle d,\, \tau[u/x]\rangle \Downarrow w$.

By the induction hypothesis,

- $F(\langle e_1,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} F(\langle \lambda x.\, d,\, \tau\rangle)$;
- $F(\langle e_2,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} F(u)$;
- $F(\langle d,\, \tau[u/x]\rangle) \xrightarrow[\text{cbv}]{*} F(w)$.

Under CBV semantics, we have

$$\begin{array}{rcl} F(\langle e_1\, e_2,\, \sigma\rangle) & = & F(\langle e_1,\, \sigma\rangle)\ F(\langle e_2,\, \sigma\rangle) \\ & \xrightarrow[\text{cbv}]{*} & F(\langle \lambda x.\, d,\, \tau\rangle)\ F(u) \\ & = & (\lambda x.\, d)\{F(\tau(y))/y,\ y \in \operatorname{dom}\tau\}\ F(u) \\ & = & (\lambda x.\, d\{F(\tau(y))/y,\ y \in \operatorname{dom}\tau,\ y \neq x\})\ F(u) \\ & \xrightarrow[\beta]{} & d\{F(\tau[u/x](y))/y,\ y \in \operatorname{dom}\tau,\ y \neq x\}\{F(\tau[u/x](x))/x\} \\ & = & d\{F(\tau[u/x](y))/y,\ y \in \operatorname{dom}\tau\} \\ & = & F(\langle d,\, \tau[u/x]\rangle) \\ & \xrightarrow[\text{cbv}]{*} & F(w). \end{array}$$

($\Rightarrow$) Suppose $F(\langle e,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} v$. We proceed by induction on the length of the derivation.

For the case $e = x$, we have $\langle x,\, \sigma\rangle \Downarrow \sigma(x)$ and

$$F(\sigma(x)) \quad = \quad x\{F(\sigma(y))/y,\ y \in \operatorname{dom}\sigma\} \quad = \quad F(\langle x,\, \sigma\rangle) \quad \xrightarrow[\text{cbv}]{*} \quad v.$$

By the lemma, $F(\sigma(x)) \in \mathbf{Val}$, therefore $F(\sigma(x)) = v$, so we can take $w = \sigma(x)$.

For the case $\lambda x.\, e$, we have $\langle \lambda x.\, e,\, \sigma\rangle \Downarrow \langle \lambda x.\, e,\, \sigma\rangle$ and

$$F(\langle \lambda x.\, e,\, \sigma\rangle) \quad = \quad (\lambda x.\, e)\{F(\sigma(y))/y,\ y \in \operatorname{dom}\sigma\} \quad \in \quad \mathbf{Val},$$

thus $F(\langle \lambda x.\, e,\, \sigma\rangle) = v$, so we can take $w = \langle \lambda x.\, e,\, \sigma\rangle$.

Finally, for the case $e_1\, e_2$, we have

$$F(\langle e_1\, e_2,\, \sigma\rangle) \quad = \quad F(\langle e_1,\, \sigma\rangle)\ F(\langle e_2,\, \sigma\rangle) \quad \xrightarrow[\text{cbv}]{*} \quad v.$$

Under the CBV reduction strategy, the only way this can happen is if

- $F(\langle e_1,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} \lambda x.\, d$;
- $F(\langle e_2,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} u$;
- $d\{u/x\} \xrightarrow[\text{cbv}]{*} v$

for some $\lambda x.\, d$ and $u \in \mathbf{Val}$. These are all shorter derivations than the original derivation $F(\langle e,\, \sigma\rangle) \xrightarrow[\text{cbv}]{*} v$, so by the induction hypothesis there exist $\lambda x.\, e$, ρ, and $t \in \mathbf{Cl}$ such that

- $\langle e_1,\, \sigma\rangle \Downarrow \langle \lambda x.\, e,\, \rho\rangle$ and $F(\langle \lambda x.\, e,\, \rho\rangle) = \lambda x.\, d$;

- $\langle e_2,\, \sigma\rangle \Downarrow t$ and $F(t) = u$.

Then

$$\begin{aligned} \lambda x.\, d &= (\lambda x.\, e)\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho\} \\ &= \lambda x.\, (e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho,\ y \neq x\}), \end{aligned}$$

therefore $d = e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho,\ y \neq x\}$, and

$$\begin{aligned} d\{u/x\} &= e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho,\ y \neq x\}\{u/x\} \\ &= e\{F(\rho(y))/y,\ y \in \operatorname{dom}\rho,\ y \neq x\}\{F(t)/x\} \\ &= e\{F(\rho[t/x](y))/y,\ y \in \operatorname{dom}\rho,\ y \neq x\}\{F(\rho[t/x](x))/x\} \\ &= e\{F(\rho[t/x](y))/y,\ y \in \operatorname{dom}\rho\} \\ &= F(\langle e,\, \rho[t/x]\rangle). \end{aligned}$$

By the induction hypothesis, $\langle e,\, \rho[t/x]\rangle \Downarrow w$ and $F(w) = v$ for some w, therefore $\langle e_1\, e_2,\, \rho[t/x]\rangle \Downarrow w$. □