

Call-by-Value λ -calculus: rules at a glance

reference sheet

1. Syntax

Terms $t, u, s ::= x \mid \lambda x. t \mid t u$
Values $v, w ::= x \mid \lambda x. t$

Application associates to the left; λ extends as far right as possible.
 $t\{u/x\}$: capture-avoiding substitution of u for x in t .
Notation: $I = \lambda x. x$, $\Delta = \lambda x. x x$, $\Omega = \Delta \Delta$.

2. The root rule

$$(\lambda x. t) v \mapsto_{\beta_v} t\{v/x\} \quad (\text{the argument must be a value})$$

Compare with call-by-name: $(\lambda x. t) u \mapsto_{\beta} t\{u/x\}$ for *any* u .

3. Full CbV reduction (Plotkin's λ_v)

$\rightarrow_{\beta_v}$ is the closure of $\mapsto_{\beta_v}$ under *all* contexts $C ::= \langle \cdot \rangle \mid \lambda x. C \mid C t \mid t C$, i.e. $C\langle t \rangle \rightarrow_{\beta_v} C\langle t' \rangle$ whenever $t \mapsto_{\beta_v} t'$.
Equivalently:

$$\frac{}{(\lambda x. t) v \rightarrow_{\beta_v} t\{v/x\}} \beta_v \quad \frac{t \rightarrow_{\beta_v} t'}{\lambda x. t \rightarrow_{\beta_v} \lambda x. t'} \lambda \quad \frac{t \rightarrow_{\beta_v} t'}{t u \rightarrow_{\beta_v} t' u} @L \quad \frac{u \rightarrow_{\beta_v} u'}{t u \rightarrow_{\beta_v} t u'} @R$$

Fact (Plotkin 1975). $\rightarrow_{\beta_v}$ is confluent; it satisfies a standardization theorem.

4. Weak CbV reduction (no reduction under λ)

$\rightarrow_w$ is the closure of $\mapsto_{\beta_v}$ under *weak* contexts $W ::= \langle \cdot \rangle \mid W t \mid t W$ — i.e. the rules of §3 *without* rule (λ) :

$$\frac{}{(\lambda x. t) v \rightarrow_w t\{v/x\}} \beta_v \quad \frac{t \rightarrow_w t'}{t u \rightarrow_w t' u} @L \quad \frac{u \rightarrow_w u'}{t u \rightarrow_w t u'} @R$$

Abstractions are *frozen*: $\lambda x. t$ is $\rightarrow_w$ -normal for every t . Weak reduction is non-deterministic (left/right of an application).

Properties.

-
-

Deterministic weak strategies (evaluation contexts)

$$\begin{array}{ll} \text{Left-to-right: } E ::= \langle \cdot \rangle \mid E t \mid v E & \frac{u \rightarrow_{\ell} u'}{v u \rightarrow_{\ell} v u'} @R_{\ell} \\ \text{Right-to-left: } E ::= \langle \cdot \rangle \mid t E \mid E v & \end{array} \quad (\text{left-to-right: replaces @R; @L unchanged})$$

5. Examples

- $(\lambda x. y) \Omega$: CbN gives y in one step; in CbV it only reduces to itself ($\Omega \rightarrow_{\beta_v} \Omega$), never to y .
- $(\lambda x. x x) (I I) \rightarrow_w (\lambda x. x x) I \rightarrow_w I I \rightarrow_w I$ (the argument is evaluated *before* being duplicated).
- $\lambda y. (I I)$: $\rightarrow_{\beta_v}$ -redex inside the body, $\lambda y. (I I) \rightarrow_{\beta_v} \lambda y. I$; but it is $\rightarrow_w$ -normal.
- $x (\lambda y. \Omega)$ and $(\lambda x. \Delta)(y z) \Delta$: open, $\rightarrow_w$ -normal, *not* values (stuck terms).