TD: Operational Semantics

Full β-reduction is the basis for the symbolic manipulation of λ-terms, *e.g.*, in proof assistants, in program transformations, and in higher-order unification.
However, when the λ-calculus is regarded as the core of a programming language it is sensible to consider *weaker* reduction strategies

A *weak* reduction is a strategy to reduce λ-terms that does *not* reduce under abstractions (ie in the body of a function)

Memo:
A one-step β-reduction is the closure of the β-rule under context

C:: = [] | Cu | uC | λx.C

Said otherwise

A one-step β-reduction is given inductively by

$$\frac{}{(\lambda x.t)\ u \to_\beta t\{x\backslash u\}} \qquad \frac{t \to_\beta t'}{\lambda x.t \to_\beta \lambda x.t'}$$

$$\frac{t \to_\beta t'}{t\ u \to_\beta t'\ u} \qquad \frac{u \to_\beta u'}{t\ u \to_\beta t\ u'}$$

☐ **Ex. 0.** Change the rules above, so that we do not reduce under λ.
Evaluate K(II)

Def. *A* **value** *V is a* ***closed*** *λ-term of the shape* ***λx.t*** *(a λ-abstraction).*

$$E ::= [\,] \mid Et \qquad (\textit{call-by-name evaluation context})$$
$$E ::= [\,] \mid Et \mid VE \qquad (\textit{call-by-value evaluation context}).$$

Call-by-value (weak)		**Call-by-name (weak)**
$\frac{}{(\lambda x.t)\ V \rightsquigarrow_v t\{V/x\}}$ $\frac{t \rightsquigarrow_v t'}{t\ u \rightsquigarrow_v t'\ u}$ $\qquad$ $\frac{u \rightsquigarrow_v u'}{V\ u \rightsquigarrow_v V\ u'}$		$\frac{}{(\lambda x.t)\ u \rightsquigarrow_n t\{u/x\}}$ $\frac{t \rightsquigarrow_n t'}{t\ u \rightsquigarrow_n t'\ u}$
What are the normal forms?		What are the normal forms?

☐★ **Ex 1. Progression.**
Assume t is a closed term. Prove that (in both systems) *values are exactly the normal forms.*
(Hence if a term is not a value, it has a reduction step)

☐★ **EX 2. Determinism**

1. **Proposition (decomposition)** *Let t be a closed λ-term. Then either t is a value or there is a unique call-by-name (call-by-value) evaluation context E such that:*

$$t \equiv E[(\lambda x.u_1)u_2] \qquad t \equiv E[(\lambda x.u_1)V].$$

2.
- If $t \rightsquigarrow_v u$ and $t \rightsquigarrow_v u'$, then $u = u'$.
- If $t \rightsquigarrow_n u$ and $t \rightsquigarrow_n u'$, then $u = u'$.

3.
- If $t \Downarrow_v V$ and $t \Downarrow_v V'$, then $V = V'$.
- If $t \Downarrow_n V$ and $t \Downarrow_n V'$, then $V = V'$.

Call-by-value (small-steps)		Call-by-value (BIG-STEPS)
$\dfrac{}{(\lambda x.t)\, V \rightsquigarrow_v t\{V/x\}}$ $\dfrac{t \rightsquigarrow_v t'}{t\, u \rightsquigarrow_v t'\, u}$ $\quad \dfrac{u \rightsquigarrow_v u'}{V\, u \rightsquigarrow_v V\, u'}$		$\dfrac{}{V \Downarrow_v V}$ $\quad \dfrac{t \Downarrow_v \lambda x.r \quad u \Downarrow_v W \quad r\{W/x\} \Downarrow_v V}{t\, u \Downarrow_v V}$

Call-by-name (small-steps)		Call-by-name (BIG-STEPS)
$\dfrac{}{(\lambda x.t)\, u \rightsquigarrow_n t\{u/x\}}$ $\dfrac{t \rightsquigarrow_n t'}{t\, u \rightsquigarrow_n t'\, u}$		$\dfrac{}{V \Downarrow_n V}$ $\quad \dfrac{t \Downarrow_n \lambda x.r \quad r\{u/x\} \Downarrow_n V}{t\, u \Downarrow_n V}$

☐★ **EX 3. Relating big steps and small steps**

1.
   - If $t \Downarrow_v V$, then $t \rightsquigarrow_v^* V$.
   - If $t \Downarrow_n V$, then $t \rightsquigarrow_n^* V$.
2.
   - If $t \rightsquigarrow_v^* u$ and u is a CBV normal-form, then $t \Downarrow_v u$.
   - If $t \rightsquigarrow_n^* u$ and u is a CBN normal-form, then $t \Downarrow_n u$.

☐ **EX 4. Remember Church numerals?**

Consider the definition of *Church Numerals* :

$$\overline{n} \equiv \lambda f.\lambda x.\ (\underbrace{f \ldots (f}_{n \text{ times}}\, x) \ldots)$$

and the following encodings

$$S \quad \equiv \quad \lambda n.\lambda f.\lambda x.f(n f x) \qquad \text{(successor)}$$

1. Evaluate: S **1**
2. Does the following statement still make sense in CbV ?
   Think of the Church numeral $\underline{n}$ as the procedure that takes a function-input and an argument-input, and applies the function n-times to the argument.

☐ **EX 5. Scott Numerals**

$$\ulcorner 0 \urcorner \overset{\text{def}}{=} \mathbf{k}$$
$$\ulcorner n+1 \urcorner \overset{\text{def}}{=} \lambda xy.y\ulcorner n \urcorner$$
$$\text{succ} \overset{\text{def}}{=} \lambda nxy.yn$$
$$\text{pred} \overset{\text{def}}{=} \lambda p.p\theta\mathbf{i}$$
$$\text{case} \overset{\text{def}}{=} \lambda xyz.xyz$$

where θ is any closed term and $\mathbf{i}$ the identity.

Evaluate:

1. succ [2]
2. pred [2]
3. case [n] fg (for f and g values)