Wednesday, October 13, 2021 10:56 AM

Last time
- Parsing expressions / terms
- Reducible expressions (Redexes)

Today
- Reducible expressions
- β-reductions
- Booleans

Reminder. A reducible expression is a term (you can put parentheses around it) of the form

$$\ldots\ldots((\lambda x.(t))(t'))\ldots\ldots$$

where t and t' are also terms (we can put parentheses around them)

Examples

1. $(\lambda x.\ x\ x)$ No redex

2. $x\ ((\lambda x.\ x)\ x)$

$((x\ (\lambda x.\ x))\ x)$ No redex

3. $(x\ (\lambda x.\ x\ (\lambda x.\ x\ y\ z)))$ No redex

3. $(x\ (\lambda x.\ x\ (\lambda x.\ x\ y\ z)))$ No redex

4. $(\lambda x.\ (\lambda y.\ x\ y\ z))$ No redex

No parentheses → No redex

parentheses →  maybe

5. $((\lambda x.\ (y))\ (z))$

6. $\lambda x.\ y\ z$ No redex

7. $(\lambda x.\ y)\ (\lambda x.\ y\ z\ (\lambda x.\ x))$ One redex

$(\lambda x.\ t)$ t'

8. $((((\lambda x.\ x)\ (\lambda x.\ x))\ (\lambda x.\ x))\ (\lambda x.\ x))$

I II III IV

not a term

Missing λx One redex.

8'. $((\lambda x.\ (\lambda x.\ x)\ (\lambda x.\ x))\ (\lambda x.\ x))\ (\lambda x.\ x)$

2 redexes

9. $((\lambda x.\ x)\ (\lambda x.\ x))\ ((\lambda x.\ x)\ (\lambda x.\ x))$

2 redexes

Definition. A lambda expression is in normal form if it has no redexes

---

β-reductions. After we identify redexes, we can apply β-reductions

$$(\lambda x.\ t)\ t' \xrightarrow{\beta} [x \to (t')]\ t$$

- x: formal parameter
- t': argument
- $[x \to (t')]\ t$: replace x with (t') in t

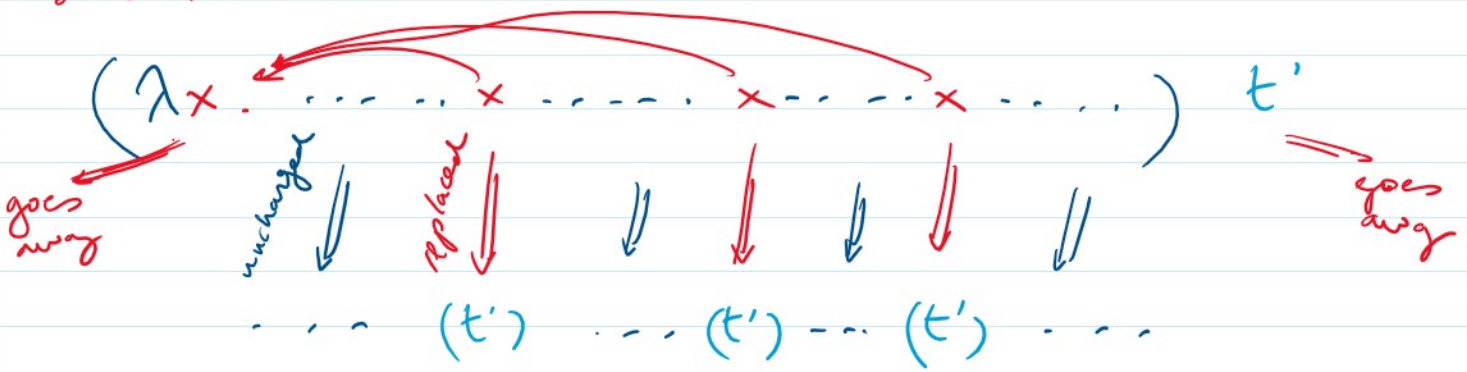
the x's that are replaced are those that are bound to $\lambda x.$

Evaluation: replace all instances of the formal parameter with the argument

There is sometimes a need for renaming which I will address later

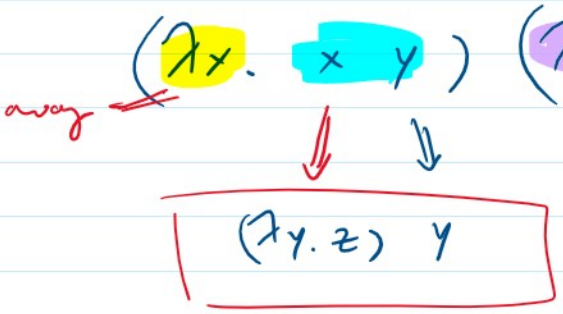
Here is sometimes a need for renaming which I will address later

Illustration

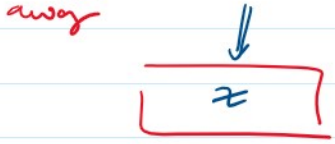


Examples.

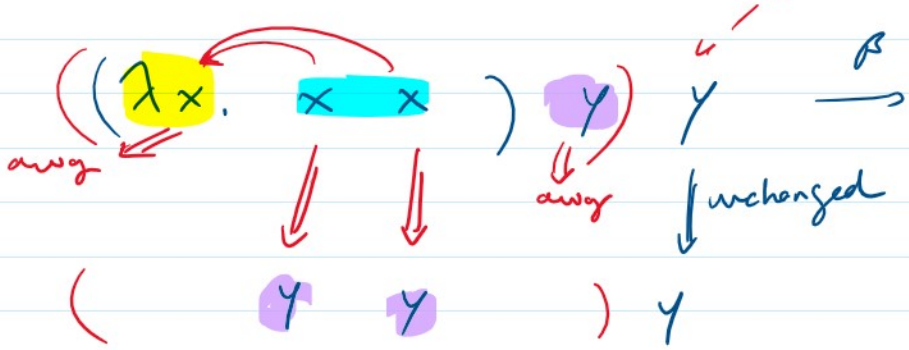
1. $(\lambda x.\ x\ y)\ (\lambda y.\ z) \longrightarrow$



$(\lambda y.\ z)\ y \longrightarrow$



1'. $((\lambda x.\ x\ x)\ y)\ y \longrightarrow$



2'. $(\lambda x.\ x\ x)\ (\lambda x.\ x\ x) \longrightarrow$

2'. $(\lambda x.\ x\ x)(\lambda x.\ x\ x) \longrightarrow$

$(\lambda x.\ x\ x)(\lambda x.\ x\ x) \longrightarrow$

$(\lambda x.\ x\ x)(\lambda x.\ x\ x) \longrightarrow \cdots$

This expression is called Omega

It has no normal form.

You can evaluate it any way you want and the evaluation never terminates.

3'. $(\lambda x.\ a)\ ((\lambda x\ x\ x)(\lambda x.\ x\ x)) \xrightarrow{\beta}$

I

$\downarrow$

a

$\lambda x.\ a$ is a constant function.

---

if $\underbrace{(x\ !=\ 0)}_{a}\ \underbrace{\&\&}_{\text{and}}\ \underbrace{(\frac{1}{x} > 10)}_{b}$

Short circuit evaluation

and (a, b) = if a then b else false

eve

else false

4'. $(\lambda x.\ x\ (\lambda x.x)\ x)\ (\lambda x.a)$ $\rightarrow$

$((\lambda x.a)\ (\lambda x.x))(\lambda x.a)$ $\rightarrow$

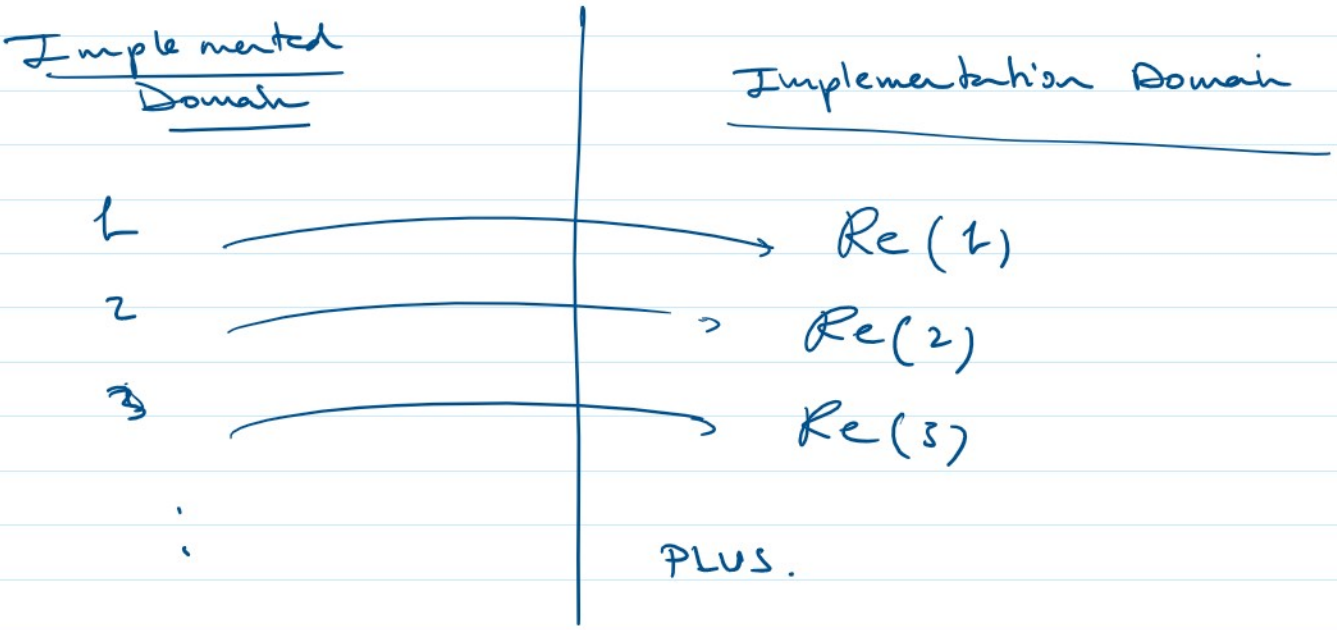
$a\ (\lambda x.a)$

Booleans.

$tru = \lambda a.\ \lambda b.\ a$

$fls = \lambda a.\ \lambda b.\ b$

$AND = \lambda a.\ \lambda b.\ a\ b\ fls$



$n \longrightarrow Re(n)$

$m \longrightarrow Re(m)$

$Re(n)$: representation of n PLUS $Re(n)\ Re(m) = Re(n+m)$

$tru = \lambda a.\ \lambda b.\ a$

$((tru\ x)\ y) \xrightarrow{\beta} ((\lambda a.\ \lambda b.\ a)\ x)\ y \xrightarrow{\beta}$

$(\lambda b\ x)\ y \xrightarrow{\beta}$

x

$tru\ t_1\ t_2 = t_1$

$tru\ (t_1\ t_2) \neq t_1$

$fls = \lambda a.\ \lambda b.\ b$

$fls\ x\ y = ((\lambda a\ \lambda b.\ b)\ x)\ y \xrightarrow{\beta}$

$(\lambda b.\ b)\ y \xrightarrow{\beta}$

y

$fls\ t\ t \ldots t$

$$fls\ t_1\ t_2 = t_2$$

$$AND = \lambda a.\ \lambda b.\ a\ b\ fls$$

$$AND\ fls\ x = ((\lambda a.\ \lambda b.\ a\ b\ fls)\ fls)\ x \xrightarrow{\beta}$$

$$(\lambda b.\ fls\ b\ fls)\ x \xrightarrow{\beta}$$

$$fls\ \underset{1}{x}\ \underset{2}{fls} = fls$$

$$AND\ tm\ x = ((\lambda a.\ \lambda b.\ a\ b\ fls)\ tm)\ x$$

$$\xrightarrow{\beta} (\lambda b.\ tm\ b\ fls)\ x$$

$$\xrightarrow{\beta} (tm\ \underset{1}{x}\ \underset{2}{fls}) = x$$

$$NOT = \lambda a.\ a\ fls\ tm$$

$$NOT\ tm = (\lambda a.\ a\ fls\ tm)\ tm =$$

$$tm\ \underset{1}{fls}\ \underset{2}{tm} = \underset{1}{fls}$$

$$OR = \lambda a\ \lambda b.\ a\ tm\ b)$$

$OR = \lambda a.\, \lambda b.\, a \text{ tru } b$

$XOR = \lambda a.\, \lambda b.\, a \ (NOT\ b)\ b$