Def$^{\underline{n}}$: (Pre-terms).

Assume given an infinite set of variables V.
Let A be the alphabet $V \cup \{ (,), \lambda, \cdot \}$
Let A^* be the set of all possible strings over A.
Let Λ be the smallest subset of A^* s.t:

- If $x \in V$, then $x \in \Lambda$ — Variables
- If $M, N \in V$, then $(MN) \in \Lambda$ — Applications
- If $x \in V$ and $M \in \Lambda$, then $(\lambda x . M) \in \Lambda$ — Abstractions.

The elements of Λ are called pre-terms.

Examples:

Pre-terms:	Not pre terms:
x	$()$
$((xx)(yy))$	λx
$(\lambda x . x)$	$(x_1(x_2(x_3(\dots$
$(\lambda x . (\lambda y . x))$	$\lambda M . N$, M not a variable.
$(\lambda x . (xx))$	

Convensional Shorthand.

- Omit outer brackets.
- Applications associate to the ~~right~~ left.
- Abstractions associate to the right.
- Multiple abstractions can be concatinated.

Write	Mean	Don't mean
MN	(MN)	
$\lambda x.M$	$(\lambda x.M)$	
MNP	$((MN)P)$	$(M(NP))$
$M_1M_2M_3M_4$	$(((M_1M_2)M_3)M_4)$	$(M_1M_2)(M_3M_4)$, $(M_1(M_2(M_3M_4)))$
$\lambda x.MN$	$(\lambda x.(MN))$	$((\lambda x.M)N)$
$\lambda xyz.M$	$(\lambda x.(\lambda y.(\lambda z.M)))$	$(\lambda(xyz).M)$

Free variables.

Def^n 2: (The set $FV(\cdot)$ of free variables).

let $x \in V$, $M, N \in \Lambda$:

- $FV(x) = \{x\}$ ①
- $FV(MN) = FV(M) \cup FV(N)$ ②
- $FV(\lambda x.M) = FV(M) - \{x\}$. ③

Eg) Find $FV(x\lambda x.xy)$

$= FV(x) \cup FV(\lambda x.xy)$ ②

$= \{x\} \cup (FV(xy) - \{x\})$ ①,③

$= \{x\} \cup ((FV(x) \cup FV(y)) - \{x\})$ ②

$= \{x\} \cup ((\{x\} \cup \{y\}) - \{x\})$ ①

$= \{x\} \cup (\{x,y\} - \{x\})$

$= \{x\} \cup \{y\}$

$= \{x,y\}$.

Def^n 3: (Free/bound occurrences).
An occurrence of x in M is said to be a bound occurrence iff the occurrence is in part of M of the form $\lambda x.L$, otherwise, we say it is a free occurrence.

Lemma: $x \in FV(M)$ iff there is a free occurrence of x in M.
(Proof by induction on $\ell(M)$, the length of M).

Def^n 4: (Substitution). Replacing x in M with N, written: $M[x:=N]$.
2 cases: x only has bound occurences, or no occurrences in M.
or
x has a free occurrence in M, but not in part of M with the shape $\lambda y.L$, $y \in FV(N)$ $(x \neq y)$.

In these two cases, substitution is defined, and has the following properties:

- $x[x:=N] = N$
- $y[x:=N] = y$, if $x \neq y$.
- $(PQ)[x:=N] = P[x:=N]Q[x:=N]$
- $(\lambda x.P)[x:=N] = \lambda x.P$
- $(\lambda y.P)[x:=N] = \lambda y.P[x:=N]$, if $x \neq y$.

If we ignore our requirements for substitution to be defined, and naively replaced x in M with N, we get the following:

$(\lambda y.xz)[x:=yz] = \lambda y.yzz$ (Naive)

$(\lambda y'.xz)[x:=yz] = \lambda y'.yzz$.

Later we will identify $\lambda y.xz$ and $\lambda y'.xz$, but not $\lambda y.yzz$ and $\lambda y'.yzz$.
We notice that in our naive substitution, we have captured our variable y, which is free in yz, but not in $\lambda y.yzz$.
We must avoid this.
This explains why we only define substitution in one of these two specific circumstances.

Another helpful lemma:

If $x \notin FV(L)$ or $y \notin FV(M)$ and all substitutions are defined, then

$$M[x:=N][y:=L] = M[y:=L][x:=N[y:=L]].$$

Proof is a great exercise!

Def^n 5: (α-equivalence). (α-conversion).

Smallest transitive, reflexive relation on Λ satisfying:

- If $y \notin FV(M)$ and $M[x:=y]$ is defined, then $\lambda x . M =_\alpha \lambda y . M[x:=y]$
- If $M =_\alpha N$, then $\lambda x . M =_\alpha \lambda x . N$, $\forall x \in V$.
- If $M =_\alpha N$, then $MZ =_\alpha NZ$
- If $M =_\alpha N$, then $ZM =_\alpha ZN$.

Examples:

$$\lambda xy . xy =_\alpha \lambda yx . yx$$
$$\lambda x . xy \neq_\alpha \lambda y . yx.$$

Helpful lemmas:

- If $M =_\alpha N$, then $FV(M) = FV(N)$
- If $M =_\alpha N$, then $\ell(M) = \ell(N)$
- If $M =_\alpha M'$ and $N =_\alpha N'$, then $M[x:=N] =_\alpha M'[x:=N']$.

Def^n 6: λ-terms.

α-Equivalence classes of pre-terms. ~~with respect to α-equivalence.~~

Define: $\Lambda_\alpha = \{ [M]_\alpha \mid M \in \Lambda \}$, $[M]_\alpha = \{ N \in \Lambda \mid M =_\alpha N \}$.

Example: $[\lambda x . x]_\alpha = \{ \lambda x . x, \lambda y . y, \dots \}$

Write: $\lambda x . x$.

Note: $M =_\alpha N \Rightarrow FV(M) = FV(N)$ (Proof on page 6).

$M[x:=N] =_\alpha M'[x:=N']$ (if $M =_\alpha M'$, $N =_\alpha N'$). (Proof on page 6)

Free variables and substitution for λ-terms:
We have already stated that these notions are well defined wrt α-equivalence.

~~So pick a pre-term representative, calculate the pre-term FV(·) set / perform the substitution, then~~

Free variables:
Pick a pre-term representative, calculate the free-variable set of this pre-term.
Since $FV(\cdot)$ is well defined wrt α-equivalence, it is OK to call this the free variable set of the λ-term.

Substitution:
Pick a pre-term representative of the equivalence class s.t the corresponding ~~defini~~ substitution is defined. Perform this substitution, then take the equivalence class of this λ-term.

Example: $(\lambda x . zy)[z := xy]$
Take the pre-term: $\lambda x' . zy$
$\Rightarrow (\lambda x' . zy)[z := xy]$
$= \lambda x' . xyy$
Take the equivalence class:
$\lambda x' . xyy$ is fine, might write: $\underline{\lambda z . xyy}$

Def^n 7: (β-reduction, β-redux).
The least compatible ~~funct~~ relation on Λ_α s.t:
i) $(\lambda x.P)Q \rightarrow_\beta P[x:=Q]$
ii) if $M \rightarrow_\beta N$, then $\lambda x\, M \rightarrow_\beta \lambda x.N$
iii) if $M \rightarrow_\beta N$, then $MZ \rightarrow_\beta NZ$, and $ZM \rightarrow_\beta ZN$.

A λ-term of the form $(\lambda x.P)Q$ is called a β-redex.
Note: There is an important difference between β-redexes and abstractions.
$(\lambda x.y)z$ is a β-redex
$(\lambda x.y)$ is not.

~~Examples~~:
Def^n 8: (multi-step β-reduction)
If we allow transitivity, then we get multi-step β-red.
Write: $P \twoheadrightarrow_\beta Q$.

Examples: $(\lambda a.ab)cd \rightarrow_\beta cbd$
$(\lambda a.ab)(cd) \rightarrow_\beta cdb$.
$(\lambda a.b)(\underbrace{\lambda zxy.xx(M\lambda a.aa)}_{\text{Is this a } \beta\text{-redex? No!}}) \rightarrow_\beta b$

$(\lambda x.xx)(\lambda x.xx) \rightarrow_\beta (\lambda x.xx)(\lambda x.xx)$
$(\lambda x.xxx)(\lambda x.xxx) \rightarrow_\beta (\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx)$
let $\omega = \lambda x.xx$, $I = \lambda x.x$. $\omega(II) \rightarrow_\beta \omega I$
$\omega(II) \rightarrow_\beta (II)(II)$
$(\lambda p.p(\lambda xy.y))(\lambda x.xMN) \rightarrow_\beta (\lambda x.xMN)(\lambda xy.y) \rightarrow_\beta (\lambda xy.y)MN \twoheadrightarrow_\beta N$.
$(\lambda xy.x)MN \twoheadrightarrow_\beta M$.

(Very) helpful lemmas.

i) If $x \notin FV(M)$, then $M[x:=N]$ is defined, and $M[x:=N] = M$.

ii) If $M[x:=N]$ is defined, then $y \in FV(M[x:=N])$ iff
- $y \in FV(M)$, and $x \neq y$, or
- $y \in FV(N)$, and $x \in FV(M)$.

iii) $M[x:=x]$ is defined, and $M[x:=x] = M$.

iv) If $M[x:=y]$ is defined, then $\ell(M) = \ell(M[x:=y])$.

Prove all by induction on $\ell(M)$.

eg) iv:

Base case, $\ell(M)=1$. $M = var$, result is obvious.

Inductive step, Assume $\ell(M)=k$, result is true for all N s.t. $\ell(N)<k$, ie for all N s.t. $\ell(N)<k$, $\ell(N) = \ell(N[x:=y])$.

3 cases, M a variable, application, or abstraction.

Case 1: Back to base case.

Case 2: $M = M_1 M_2$.

$$\begin{aligned}\ell(M[x:=y]) &= \ell(M_1[x:=y]\,M_2[x:=y]) \\ &= \ell(M_1[x:=y]) + \ell(M_2[x:=y]) + 2 \\ &= \ell(M_1) + \ell(M_2) + 2, \text{ induct hyp.} \\ &= \ell(M_1 M_2) \\ &= \ell(M)\end{aligned}$$

Case 3: $M = (\lambda z.Q)$, with either $z = x$, $z \neq x$.

Result is obvious for $z = x$.

If $z \neq x$, $\ell(M[x:=y])$

$$\begin{aligned}&= \ell(\lambda z.Q[x:=y]) \\ &= \ell(Q[x:=y]) + 5 \\ &= \ell(Q) + 5, \text{ induct hyp.} \\ &= \ell(\lambda z.Q) \\ &= \ell(M) \qquad \square\end{aligned}$$