

DAY 8

Parser->lexer->type checking->???->knormal (from bottom to top)

```
let lexbuf outchan 1 =
  .....
  (.....
    (.....
      (.....
        (.....
          (iter !limit
            (.....
              (.....
                (.....
                  (..... .))))))))))
```

Reference: point to a box containing content

Dereference: value pop out of the box???

```
let rec iter n e = (* Iterative optimization *)
  if n = 0 then e else
    let e' = Elim.f (ConstFold.f (Inline.f (Assoc.f (Beta.f e)))) in
    if e = e' then e else
      iter (n - 1) e'

let lexbuf outchan 1 =
  .....
  (.....
    (.....
      (.....
        (.....
          (iter !limit
            (.....
              (.....
```

Beta Reduction:

β reduction: ε is a mapping that takes a variable name and gives its β -converted name.
We consider $\varepsilon(x) = x$ when ε is not defined for x .

```
# beta_p
let x = 1 in
let y = x in
x + y
```

```
# alpha beta_p
Let (Ti5_6,
  Let (x_7, 1,
    Let (y_8, x_7, Add(x_7, y_8))), ...)
```

```
# beta beta_p
Let (Ti1_2,
  Let (x_3, 1, Add(x_3, x_3)), ...)
```

In min-calm check the examples above
cd min-caml —>./min-caml.top
control+D

```
(* assoc_let_p*)
let x =
  let y = 1 in
  let z = 2 in
  y + z in
x
```

Let Reduction:

```
(* assoc_let_p*)
let x =
  let y = 1 in
  let z = 2 in
  y + z in
x
```

move let y&z outside the let x