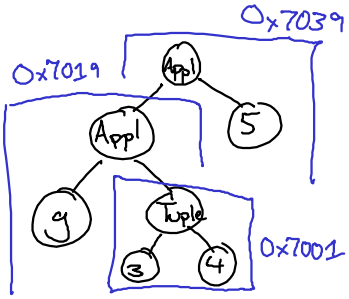
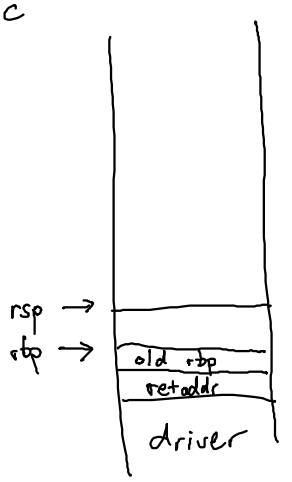
Falcon example

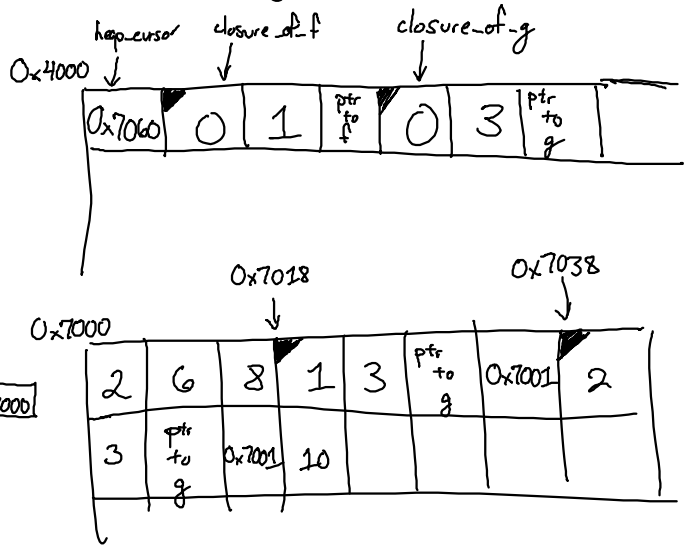
```
def f h =
  h 4
end
def g a b c =
  a[0] + a[1] + b + c
end
f (g (3,4) 5)
```

0x7019




```
section .data
align 8
heap_cursor:
  dq 0
closure_of_f:
  dq 0x8000000000000000, 1, fn_f
closure_of_g:
  dq 0x8000000000000000, 8, fn_g
```



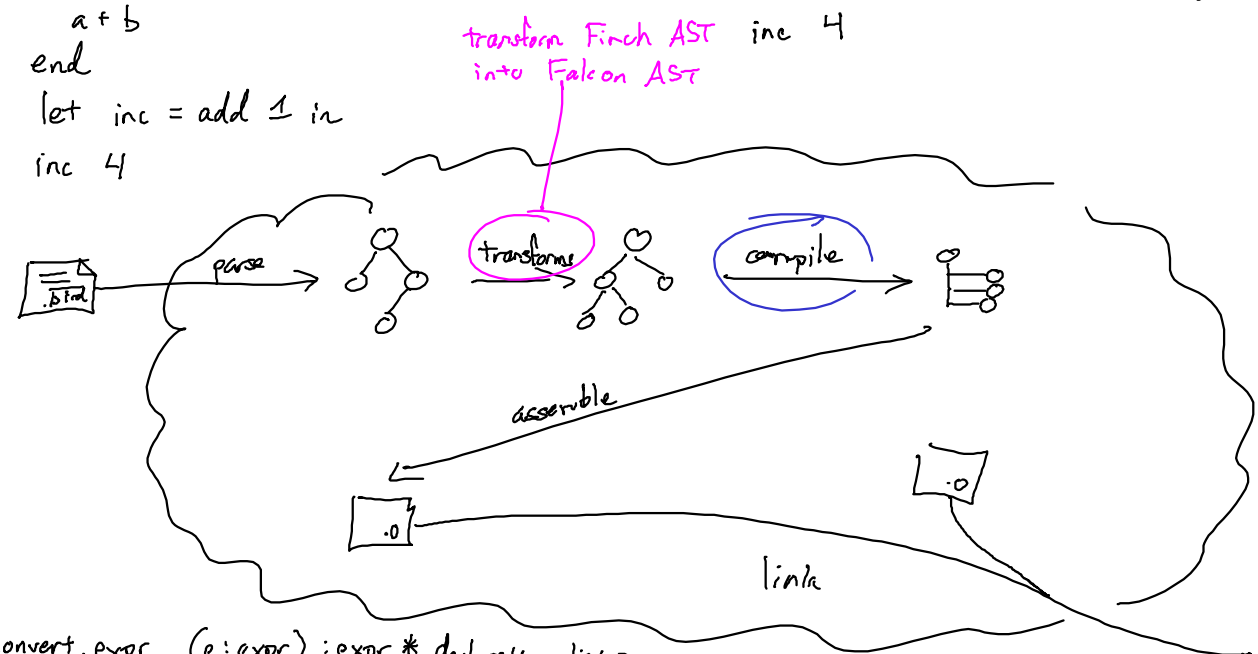


Finch

<expr> ::= ... | fun <paran_list> → <expr>

```
def add a b =
  a + b
end
let inc = add 1 in
inc 4
```

```
let inc = (fun a b → a + b) 1 in
inc 4
```




```
let rec closure_convert_expr (e : expr) : expr * declaration list =
  ...
```

```
let inc = (fun a b → a + b) 1 in
inc 4
```

```
def $0 a b =
  a+b
end
let inc = $0 1 in
inc 4
```

a is free

```
def f a =
  (fun b → a+b)
end
let q = f 2 in
q 5
```

```
def $0 a b =
  a + b
end
def f a
  $0 a
end
let q = f 2 in
q 5
```