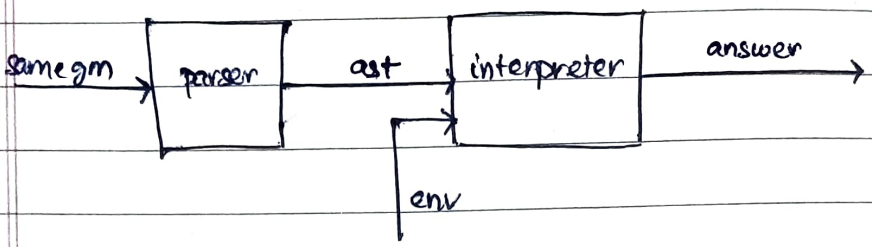
classmate
Date 14. 9. 2019
Page 1

LEXICAL LANGUAGE



- semantic domains
- abstract syntax datatype
-

e ::= n̄ | b̄ | x | (assume ([x ê]...) e) | (op e...)

op ::= + | - | * | / | > | =

(define-datatype ast ast?
  [num (n number?)]
  [bool (b boolean?)]
  [id (x id?)]
  [prim-app (op op?) (operands (list-of ast?))]
  [assume (binds binds?) (body ast?)]

id? a valid identifier.

ENVIRONMENTS

lookup: env? id? → denotable value? (error)

extend-env: env? (list-of id?) (list-of? value?) → env?

empty-env : env?

#t #f boolean?

classmate
Date 16. 9. 2019
Page 2

- [(x y z) (3 4 #t)] [(x w) (8 5)]

(define eval-ast
  (λ (a e)
    (cases ast a
      [num (n) n]
      [bool (b) b]
      ~~[op (operands)~~
      [prim-app (op operands) ...]
      [assume (binds body) identifiers on exp.
        (let ([mini-env (elaborate binds e)])
          (eval-ast body (extend-end
                          e mini-env)))]

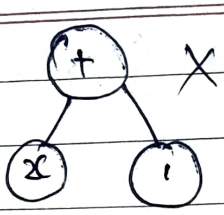
borrow idioms from each other.

FUNCTIONS AS VALUES

expressible values are functions : functional prog. lang.

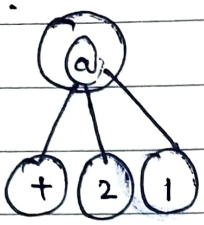
expressible-value? = number? | boolean? | proc?

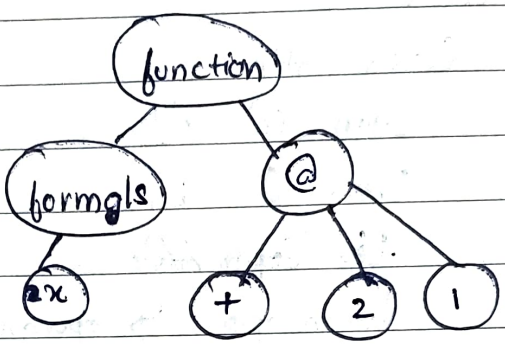
e ::= n̄ | b̄ | x | assume ([x e] ...) e |
(function (x ...) e) | (f e e e) | (e e ...)



base case.

(function (x) (+ x 1))





(@ (function (x) (+ x 1))
5)
⇒ 6

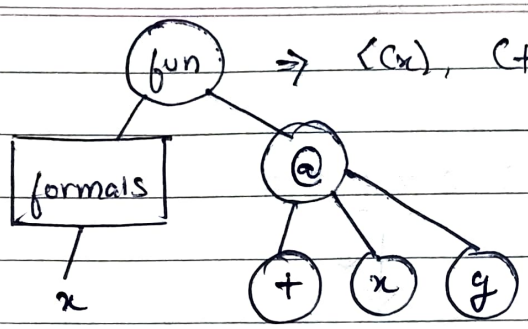
(let ([x e] ...) body) ⇒
((λ (x...) body) e...)

(let ([x 3])
(let ([y 5])
(+ x y))) ⇒ 8

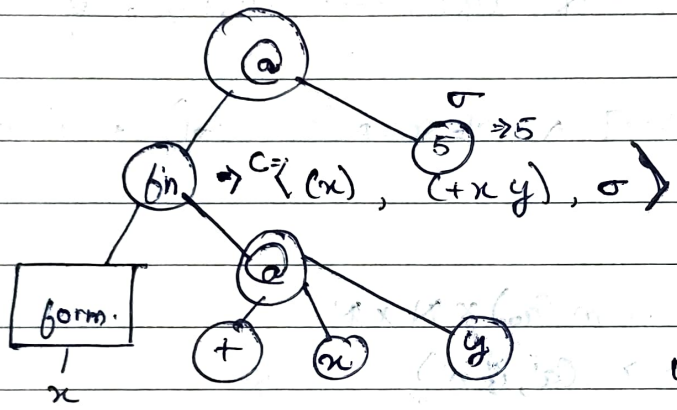
((λ(x) ((λ (y) (+ x y)) 5)) 3)

(λ (y) (+ x y))

y (+ x y) env



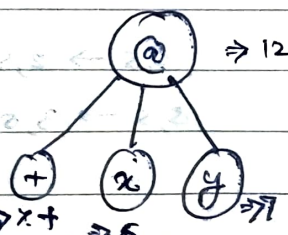
$\sigma = \{y \mapsto 7, + \mapsto \%.+\}$



1. map 5 with formal parameter.

1. build a small env by binding formals with args
2. extend the env in closure c with the mini env. α
3. evaluate the body of the closure c in (extend-env α σ)

$\alpha = \{x \mapsto 5\}$ $\alpha . \sigma$



BUBBLE SORT

7 3 1 8 2 → 3 7 1 8 2

↓ *

1 2 3 7 8

$Perm(A) \times \mathbb{N} \quad (a, i) \longrightarrow (a', i') =df=$

$a[i] > a[i+1] \wedge a' = swap(a, i, i+1) \wedge i' = i+1$

state space $perm(a_0) \times \mathbb{N} \times \mathbb{N}$

$(a, i, n) \longrightarrow (a', i', n') \quad =df=$

$(a, i, 1) \not\rightarrow$

start state

$(a_0, 0, n_0)$

$n_0 > 0$

Lexicographic ordering

• df =

① $n > 1$

② $i = n-1 \rightarrow a' = a$

$i' = 0$

$n' = n-1$

0,5 → 2,5, → 4,5, → 0,4

5,5 → 3,5 → 1,5 →

deterministic system?

and confluent.

Proof that the ARS terminates.

(termination).

★

2-index redex

a_0

7 3 2 9 4 1

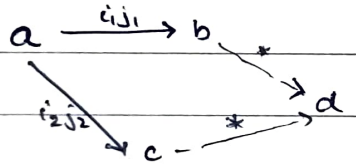
$\begin{cases} lo > hi \\ a[lo] > a[hi] \end{cases}$

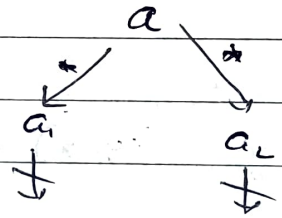
A = perm (a_0)

redex? (lo, hi) :df: $a_0 \xrightarrow{25}$ 7 3 1 9 7 2

$a \to a'$ Lexicographic order changes.

↳ termination.





confluent proof.

① a_1 is sorted, a_2 is sorted — conf.

② a_2 " a_2 not sorted — contradiction

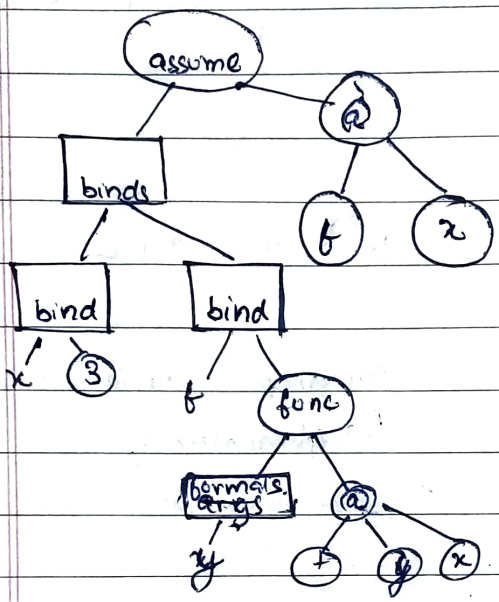
③ a_1 is not sorted, a_2 sorted — "

④ a_1 not sorted, a_2 not sorted — "

quick sort ARS?

turing machine.

e ::= $\bar{n}$ | $\bar{b}$ | x | @ e e ... | function (x ...) e |
if e e e | assume ((x e) ...) e.



$\alpha = \{x : 3, f : c\}$

$c = \langle (y), (+\ y\ x), \sigma \rangle$

$\beta = \{y : 3\}$