

מבחן 05/11/2014 סעיף ק' מועד ב'

שאלה 1. BNF

$$\langle ME \rangle ::= \langle N \rangle \quad (1)$$

נתון:

$$1 \langle N \rangle + \langle ME \rangle \quad (2)$$

$$1 \langle N \rangle - \langle ME \rangle \quad (3)$$

$$1 \langle ME \rangle * \langle N \rangle \quad (4)$$

$$1 \langle ME \rangle / \langle N \rangle \quad (5)$$

$$\langle N \rangle ::= 0111x1y1z$$

א. אלו התחבורות נכונות?

ב. השפה שמסמלת הדקדוק יש תימם באורך שנים $n-1$ כל מספר טבעי n

ג. האם ניתן לייצג על גביה?

$$1) \{ x + y * z - z^2 \}$$

$$\langle ME \rangle \xRightarrow{(1)} \{ \langle N \rangle + \langle ME \rangle \}$$

לא ניתן לייצג

$$\xRightarrow{(4)} \{ \langle N \rangle + \{ \langle ME \rangle * \langle N \rangle \} \} \Rightarrow \text{נתקנו}$$

הביטוי $z - z^2$ לא נכנס ל $\langle N \rangle$

$$2) \{ x * 0 + z^2 \}$$

$$\langle ME \rangle \xRightarrow{(2)} \{ \langle ME \rangle * \langle N \rangle \}$$

לא ניתן לייצג

מכיוון ש z^2 הוא לא מבוטא של $\langle N \rangle$

$$3) \{ 0 + 1 + 0 + 1/1/0 \}$$

כן, ניתן לייצג

$$\langle ME \rangle \xRightarrow{(1)} \{ \langle N \rangle + \langle ME \rangle \}$$

$$\xRightarrow{(4)} \{ 0 + \langle N \rangle + \langle ME \rangle \}$$

$$\xRightarrow{(2)} \{ 0 + 1 + \langle N \rangle + \langle ME \rangle \}$$

$$\xRightarrow{(3)} \{ 0 + 1 + 0 + \langle N \rangle + \langle ME \rangle \}$$

$$\xRightarrow{(5)} \{ 0 + 1 + 0 + 1 + \langle ME \rangle / \langle N \rangle \}$$

$$\xRightarrow{(5)} \{ 0 + 1 + 0 + 1 + \langle ME \rangle / \langle N \rangle / 0 \}$$

$$\xRightarrow{(1)} \{ 0 + 1 + 0 + 1 + \langle N \rangle / 1 / 0 \}$$

$$\Rightarrow \{ 0 + 1 + 0 + 1 + 1 / 1 / 0 \}$$

שאלה ב: שאלות עליות

א. תהליך ה- `scoping`

א. תהליך ה- `scoping` משני צדדים. מנתק את תהליך הערכה של התניות. בפרט, אם התוצר של התהליך הראשון יכולה להיות מופעת פונקציות הערה שונות. למדרי המחריות זכרים. מטיפים שונים.

ד. במיון שלנו, במידה הסבירות בתום תהליך ה- `scoping` מתקבל `FLANG` הוא מופיע על תחבירי אנטיקט. בגלל קודקוד מיוחד. ע"י קב' `FLANG` או ארטיקט של איתו. בגלל, רק, לעצמי שונים להיות בתום קב'.

ג. אלו מדגשפטים האזן נכונים לעבר קוטאוי ולוקאטל `Scoping`

א. ההבדל המרכזי בין שני המודלים הוא בהתחייבות. לפונקציות בסט' פונקציה היא אופקט שלם וזה אם תופס יותר מאוחר בתכנת. תשתתט תהיה באותם זכרים לשמות מזהים באז' הפונקציה. בדינאמי זכרים אלו שונים להשתתט בין המודל.

ג. במודל הדינאמי עבור הקוד הבא יוחזר 1 ואחר כך 2

```
(define (foo) x)
let([x 1] (foo)) => 1
(define (bar x) (foo))
let([x 1] bar 2) => 2
```

א. ז. הסיד. שכתבנו בקודם מת"ים לפונקציות `first` ומאחר

לפונקציה. לקבל כפרימטל פונקציה שנוצרת בזמן רצה. איך אנה מאפשרת יקומה. ג. האופטימיזציה של הקט. לתכנית פנה נוצרה לעצמי את ניהול הזיכרון בשיוט התוכנית.

שאלה 3: שפת ההיסטוריה

(define-type BIT = (U 0 1))

ליבוטת תפטים

(define-type BIT-list = (List BIT))

א) פונקציה המקבלת שני ביטס ומחזירה 1 אם שניהם שווים 1 ו-0 אחרת

(:bit-and: BIT BIT \rightarrow BIT)

(define (bit-and a b)

(if (and (= a 1) (= b 1)) 1 0))

ב) פונקציה המקבלת שני ביטס ומחזירה 1 אם לפחות אחד מהם שווה 1 ו-0 אחרת

(:bit-or: BIT BIT \rightarrow BIT)

(define (bit-or a b)

(if (or (= a 1) (= b 1)) 1 0))

ג) פונקציה המקבלת רשימה של ביטס ומחזירה רשימה שנתקבלה מהרשימה

המקורות ע"י הלוגי כל אחד מהם שאלה והלוגי הטעם

ביותר עימין הרשימה, נענו להנחיה שהרשימה אינה ריקה

(:shift-left: Bit-list \rightarrow Bit-list)

(define (shift-left b)

(if (null? b) b

(append (rest b) (list (first b)))))

7. נאמר פקדון עבור הסדר RegE

$\langle \text{RegE} \rangle ::=$

$1 \mid \text{or } \langle \text{RegE} \rangle \langle \text{RegE} \rangle$

$1 \mid \text{shl } \langle \text{RegE} \rangle$

(define-type RegE

הוא/סל את הקוד הנדרש:

[And RegE RegE]

[Or RegE RegE]

[Shl RegE]

[With Symbol RegE RegE]

[Id Symbol]

7. הא'ס כל הקד הנדרש parse-sexpr

(\vdash parse-sexpr : Sexpr \rightarrow RegE)

:

[(list 'and l reg rreg) (And (parse-sexpr l reg) (parse-sexpr rreg))]
[(list 'or l reg rreg) (Or (parse-sexpr l reg) (parse-sexpr rreg))]
[(list 'shl reg) (Shl (parse-sexpr reg))]

7. הערות הסובסטי

(\vdash subst : RegE Symbol RegE \rightarrow RegE)

(define (subst expr from to)

(cases expr

[(Reg b) expr]

[(And l r) (And (subst l from to) (subst r from to))]

[(Or l r) (Or (subst l from to) (subst r from to))]

[(Shl reg) (Shl (subst reg from to))]

7. הוספת הערות הסובסטי eval שמקבל בולרס על (בולרס) 8 eval

(\vdash bit-arith-op : (BIT BIT \rightarrow BIT) Bit-list Bit-list \rightarrow Bit-list)

(define (bit-arith-op op reg1 reg2)

(if (null? reg1) reg1

(cons (First reg1) (first reg2)

(bit-arith-op op (rest reg1) (rest reg2))))

(\vdash eval : RegE \rightarrow Bit-list)

7. הוספת הערות eval

(define (eval expr)

:

[(And l r) (bit-arith-op bit-and (eval l) (eval r))]

[(Or l r) (bit-arith-op bit-or (eval l) (eval r))]

[(Shl reg) (shift-left (eval reg))]

[(with bound-id named-expr bound-body)

(eval (subst bound-body bound-id

(Reg (eval named-expr))

שאלה 4: הוכח כי FLANG

נתון הקוד הבא:

```

(run (with x (fun y z z)
      (with f (fun z (call x z z))
        (with x 3
          (call f x z z z))
        )
      )
  )
  
```

א. הוכח כי קוד זה מייצג את הוכח.

AST₁: (With ^{id} x (Fun y (Id y)) (With f (Fun z (Call (Id x) (Id z))) (With x (Num 3) (Call (Id f) (Id x) (Id x)))))

body

RET₁:

ENV₁: (EmptyEnv)

AST₂: (Fun y (Id y))

RET₂: (FunV y (Id y) (EmptyEnv))

ENV₂: (EmptyEnv)

AST₃: (With ^{id} f (Fun z (Call (Id x) (Id z))) (With x (Num 3) (Call (Id f) (Id x) (Id x)))))

body

RET₃:

ENV₃: (Extend x (FunV y (Id y), (EmptyEnv), EmptyEnv)

AST₄: (Fun z (Call (Id x) (Id z)))

RET₄: (FunV z (Call (Id x) (Id z)), ENV₃

ENV₄: ENV₃

AST₅: (With ^{id} x (Num 3) (Call (Id f) (Id x) (Id x)))

body

RET₅:

ENV₅: (Extend f (FunV z (Call (Id x) (Id z)), ENV₃), ENV₄)

AST₆: (Num 3)

RET₆: (NumV 3)

ENV₆: ENV₅

AST₇: (Call (Id f) (Id x) (Id x))

RET₇:

ENV₇: (Extend x (NumV 3), ENV₆,

←

$AST_8: Id\ f$
 $RET_8: (FunV\ \overset{id}{2}\ (\overset{body}{call\ (Id\ x)\ (Id\ z)}),\ Extend\ x\ (FunV\ y\ (Id\ y),\ (EmptyEnv)))$
 $ENV_8: ENV_4$
 $AST_9: Id\ x$
 $RET_9: NumV\ 3$
 $ENV_9: ENV_7$
 $AST_{10}: (Call\ (\overset{fun}{Id\ x})\ (\overset{arg}{Id\ z}))$
 $RET_{10}: Id\ x$
 $ENV_{10}: (Extend\ 2\ (NumV\ 3),\ (Extend\ x\ (FunV\ y\ (Id\ y),\ (EmptyEnv))\ (EmptyEnv)))$
 $AST_{11}: Id\ x$
 $RET_{11}: (FunV\ y\ (Id\ y)\ (EmptyEnv))$
 $ENV_{11}: ENV_{10}$
 $AST_{12}: Id\ z$
 $RET_{12}: (NumV\ 3)$
 $ENV_{12}: ENV_{10}$
 $AST_{13}: Id\ y$
 $RET_{13}: NumV\ 3$
 $ENV_{13}: (Extend\ y\ (NumV\ 3))$

בסוף יומך NumV 3 (שם לא נעשה צורך) כל הקוד נעשה רק למחשב.