

מבחן 09/15 שאלה 3:

BNF : 1 step

$\langle \text{TRUE} \rangle := \langle \text{ATOM} \rangle$ 
(1)      הפקוד

 $1 \prec OP \prec TREE \prec TREE$  (2)
$$\langle \text{ATOM} \rangle ::= \langle \text{num} \rangle \quad (3)$$
$$\langle \text{Op} \rangle = \pm 1/1 \quad (4)$$

א. מה' השפה שלמספר היקדוק

רביט'ו הוא קטעו fix, השפה שמעבר רביט'ו הוא  $\lambda$  חטוב לטאר

7. היטאן געזינגען זייערע אפמאכונגן 7 עלל' ציור

$$\angle \text{Tree} \stackrel{6}{=} (\angle \text{Op} \angle \text{TREE} \angle \text{TREE})$$
$$\underline{G} \rightarrow (\langle OP \rangle (\langle OP \rangle \langle TREE \rangle \langle TREE \rangle) \langle TREE \rangle)$$
$$\frac{Q}{\wedge} \rightarrow (\angle OP \rightarrow (\angle OP \rightarrow \angle TREE \rightarrow (\angle OP \rightarrow \angle TREE \rightarrow \angle TREE))) \rightarrow \angle TREE$$
$$\underline{\underline{G}} > (\angle OP > (\angle OP > \angle TREE > (\angle OP > \angle TREE > \angle TREE)))$$

$\angle OP > \angle TFE > \angle TFE E$ )

④ > (

X. האם הקב"ה ר'נו רב משמ' 7

פא, מיליון שה"ש, הוא postfix אל (יתן) לעזור אותו בזירה סחת פלד

7 compositionality נהר הנוש

שטערקחים תתיבה מנהל ואפסר עהערך אותה ברנח, טון זה מתקיה

## שאלה ב:

א. תורס'ה אין סיפת

$(: S \text{ Any} \rightarrow \text{Any}$

$(\text{define } (S \ x)$

$(S \ x)$

$(S \ S)$

ז. מה זה passing by value ל- string עזל' עזל' הדקדוק הסר הסר

תהליך זה שחשבה חלל ~~הוא~~ עשוי להיות מנותק מתהליך

ההזרה של תכנות, בפרט אל התצר של התהליך הראשון יכולות להיות מופיעות פונקציות הזכר שונות שאינן המחוברת זכרם ממופיע שונות

\* passing by value זה

יש 3 מתודות איך שפה מתמודדת עם שפות פונקציה:

(1) first class = אי אפשר להחזיר פונקציה

(2) higher order = פונקציות יכולות לקבל פונקציות ולהחזיר פונקציות

(3) first class = אפשר לקבל ולהחזיר פונקציות, ניתן לשינוי אותם

בצד, ופונקציות חדשות יכולות להיות מיוצגות בצד

## first class פונקציות:

בשפות אלו פונקציות יכולות מופיעות בצדן רוצה כך שיש פונקציות

המופיעת חלל בקט של התצר

בשפה כזו כל פונקציה יכולה להחזיר מספר מסוים (סופי של ערכים)

7. Scope & Bindings = קושר בין אובייקט לערך [הפרמטר מופיע במקום מסוים של התוכנית]

(1) האפשרות זאת שניתן לזהות ערכים האובייקטים לקראתם של התניות

כמו גם לזהות ההפסד של התניות

(2) מה משמרה binding

(3) ניתן לבדוק האם קיימת מופע חופשי של משמרה כלשהו בתניות אוד קוד

הנחתה החתירה

## שאלה ב: שאלות עליות

א. parsing - במקרה ההתפלגות והרשימות זהה לתחילת ה-parsing של שורה

אחרי התחילת אפשר לזהות שמה נזהים תופעות

תחילת ה-parsing מתויה להסתיים לפני תחילת הסכנה של הקוד.

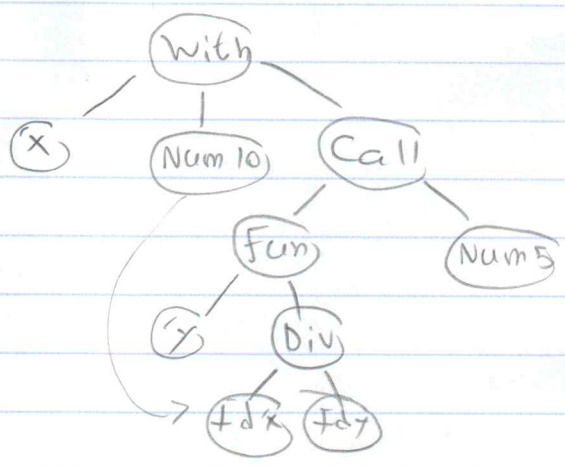
1. רשימת התפלגות קוראת בתור במהם לפונקציה  $\text{self}$  מ- $\text{self}$

כל הפונקציות בשפה  $\text{firstclass}$

שני המיפלים לא יחזירו אותו סק

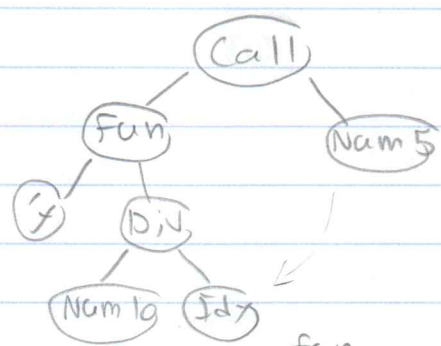






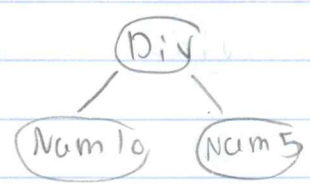
AST<sub>5</sub>: (With x (Num 10) (Call (Fun x (Div (Idx) (Idx))) Num 5))

RET<sub>5</sub>: Num 2  
 AST<sub>6</sub>: Num 10  
 RET<sub>6</sub>: Num 10



AST<sub>7</sub>: (Call (Fun x (Div (Num 10) (Idx))) <sup>fun</sup> (Num 5) <sup>arg</sup>)

RET<sub>7</sub>: Num 2  
 AST<sub>8</sub>: Fun x (Div (Num 10) (Idx))  
 RET<sub>8</sub>: Fun <sup>body</sup> (Div (Num 10) (Idx))  
 AST<sub>9</sub>: Num 5  
 RET<sub>9</sub>: Num 5



AST<sub>10</sub>: (Div (Num 10) (Num 5))

RET<sub>10</sub>: Num 2  
 AST<sub>11</sub>: Num 10  
 RET<sub>11</sub>: Num 10  
 AST<sub>12</sub>: Num 5  
 RET<sub>13</sub>: Num 5

id name body

AST<sub>1</sub>: (with <sup>id</sup> z (Num 5)) (with w (Fun y (Div (Id x) (Id y))) <sup>body</sup> (with x (Num 10) (Call (Id w) (Id z)))))

SC<sub>1</sub>: ()

RET<sub>1</sub>: Num 2

AST<sub>2</sub>: Num 5

SC<sub>2</sub>: ()

RET<sub>2</sub>: Num 5

AST<sub>3</sub>: (with <sup>id</sup> w (Fun y (Div (Id x) (Id y))) <sup>body</sup> (with x (Num 10) (Call (Id w) (Id z)))))

SC<sub>3</sub>: ((z (Num 5)))

RET<sub>3</sub>: Num 2

AST<sub>4</sub>: (Fun y (Div (Id x) (Id y)))

SC<sub>4</sub>: ((z (Num 5)))

RET<sub>4</sub>: (Fun y (Div (Id x) (Id y)))

AST<sub>5</sub>: (with <sup>id</sup> x (Num 10) (Call (Id w) (Id z)))

SC<sub>5</sub>: ((w (Fun y (Div (Id x) (Id y)))) (z (Num 5)))

RET<sub>5</sub>: Num 2

AST<sub>6</sub>: Num 10

SC<sub>6</sub>: ((w (Fun y (Div (Id x) (Id y)))) (z (Num 5)))

RET<sub>6</sub>: Num 10

AST<sub>7</sub>: (Call (<sup>fun</sup> (Id w) (<sup>arg</sup> (Id z))))

SC<sub>7</sub>: ((x (Num 10) (w (Fun y (Div (Id x) (Id y))))) (z (Num 5)))

RET<sub>7</sub>: Num 2

AST<sub>8</sub>: Id w

SC<sub>8</sub>: SC<sub>7</sub>

RET<sub>8</sub>: (Fun <sup>id</sup> y (Div (Id x) (Id y))) <sup>body</sup>

AST<sub>9</sub>: Id z

SC<sub>9</sub>: SC<sub>8</sub>

RET<sub>9</sub>: Num 5

←

$AST_{10} = (Div (Id_x) (Id_y))$

$SC_{10} = 7((\gamma Num5)(x Num10)(w(Fun\gamma(Div(Id_x)(Id_y))))(z Num5))$

$RET_{10} = Num\ 2$  כסוף

$AST_{11} = Id_x$

$SC_{11} = SC_{10}$

$RET_{11} = Num\ 10$

$AST_{12} = Id_y$

$SC_{12} = SC_{11}$

$RET_{12} = Num\ 5$

ג. מה היה קורה אם היינו מבצעים את הערה במודל הסביבתי?

מה שהיה קורה היינו מקבלים שגיאה מכיוון שמודל הסביבתי הוא מודל סטטי

אז כאשר בנקו העדכון הפונקציה 'ט' למעשה חופשי 'שם' סופי מידית

בסוף א' אומנם מודל סטטי אבל כיוון שיש כפי ביטוי א' בהמשך ש-א מוגדר

אז אין שגיאה וברשימות התלפיות הוא ביטוי א' וגם זה שיש הערה ט'ט

הערה ט'ט איך בהמשך זה נעשה

7/20/2021

AST<sub>1</sub>: (with z (Num 5) (with w (Fun y (Div (Id x) (Id y)))  
(with x (Num 10) (call (Id w) (Id z))))

ENV<sub>1</sub>: empty Env

RET<sub>1</sub>:

AST<sub>2</sub>: Num 5

ENV<sub>2</sub>: empty Env

RET<sub>2</sub>: Num 5

AST<sub>3</sub>: (with <sup>id</sup> w (Fun y (Div (Id x) (Id y)))  
(with x (Num 10) (call (Id w) (Id z))))

ENV<sub>3</sub>: (Extend 'z (Num 5) (empty Env))

RET<sub>3</sub>:

AST<sub>4</sub>: (Fun y (Div (Id x) (Id y)))

ENV<sub>4</sub>: (Extend 'z (Num 5) (empty Env))

RET<sub>4</sub>: (Fun y (Div (Id x) (Id y))) (Extend 'z (Num 5) (empty Env))

AST<sub>5</sub>: (with x (Num 10) (call (Id w) (Id z)))

ENV<sub>5</sub>: (Extend w (Fun y (Div (Id x) (Id y))) (Extend 'z (Num 5) (empty Env)))  
(Extend 'z (Num 5) (empty Env))

RET<sub>5</sub>:

AST<sub>6</sub>: Num 10

ENV<sub>6</sub>: ENV<sub>5</sub>

RET<sub>6</sub>: Num 10

AST<sub>7</sub>: (call (<sup>Fun</sup> (Id w) (<sup>exp</sup> (Id z))))

ENV<sub>7</sub>: (Extend x (Num 10) ENV<sub>6</sub>)

AST<sub>8</sub>: Id w

ENV<sub>8</sub>: ENV<sub>7</sub>

RET<sub>8</sub>: (Fun y (Div (Id x) (Id y))) (Extend 'z (Num 5) (empty Env))

AST<sub>9</sub>: Id z

ENV<sub>9</sub>: ENV<sub>8</sub>

RET<sub>9</sub>: Num 5 (Extend 'z (Num 5) (empty Env))

AST<sub>10</sub>: (Div (Id x) (Id y))

ENV<sub>10</sub>: (Extend y (Num 5) (Extend 'z (Num 5) (empty Env)))

RET<sub>10</sub>:



$AST_M: Id \times$

$\uparrow ENV_{11}: ENV_{10}$

$\downarrow RST_{11}: \text{error-free identifier}$

כלי התחביר שיתחיל

שאלה 4: הרחבת סל

א. הוספת הפונקציות:

$\angle FLANG := :$

$\{ \angle FLANG \dots \}$   
 $\{ - \angle FLANG \angle FLANG \dots \}$   
 $\{ * \angle FLANG \dots \}$   
 $\{ / \angle FLANG \angle FLANG \dots \}$   
 $\{ \angle FLANG \angle FLANG \dots \}$   
 $\{ call \angle FLANG \angle FLANG \angle FLANG \dots \}$

(define-type FLANG

:

[Add (Listof FLANG)]

[Sub FLANG (Listof FLANG)]

[Mul (Listof FLANG)]

[Div FLANG (Listof FLANG)]

:

[Funssymbol (Listof Symbol) FLANG]

[Call FLANG FLANG (Listof FLANG)]

(: not-member: Symbol (Listof Symbol)  $\rightarrow$  Boolean

(define (not-member s lst)

(cond [(null? lst) #f]

[eq? (first lst) s) #f]

[else (not-member s (rest lst))])

(: all-unique: (Listof Symbol)  $\rightarrow$  Boolean)

(define (all-unique lst)

(cons [(null? lst) #f]

[(not-member (first lst) (rest lst)) (all-unique (rest lst))]

[else #f])

(: parse-sexpr: sexpr  $\rightarrow$  FLANG)

[cons 'fun mod)

(match sexpr

[list 'fun (list (symbol: names) ...) body)

(if (all-unique? names)

(Fun Names args) (rest names)

(parse-sexpr body)

(else (error 'Fun expected for unique arguments

[list '+ args ...) (Add (parse-sexpr\* args))]

[list '- fst args ...) (Sub (parse-sexpr fst)

(parse-sexpr\* args))]

[list '\* args ...) (Mul (parse-sexpr\* args))]

[list '/ fst args ...) (Div (parse-sexpr fst)

(parse-sexpr\* args))]

;

[list 'call fun args ...) (Call (parse-sexpr fun)

(parse-sexpr\* args))]

7. נכתב חלק הקריאה של המערכת

(: extend\* (list of symbol) (list of val) ENV  $\rightarrow$  ENV)

(define (extend\* names vals env)

(if (and (null? names) (null? vals))

env

(extend\* (rest names) (rest vals) env)

(extend (first name) (first vals) env)))

←

eval new

C: eval: FLANG ENV  $\rightarrow$  VAL

(defn (eval expr env  
map NumV  $\rightarrow$  NumVal env  
[Add args) (NumV ((foldl + 0 (eval \* args)<sup>env</sup>))  
[Sub fst args] (NumV (foldl - (eval fst)<sup>env</sup>  
(eval \* args)<sup>env</sup>)  
[Mul args] (NumV ((foldl \* 0 (eval \* args)<sup>env</sup>)  
[Div fst args] (NumV ((foldl / (eval fst)<sup>env</sup>  
(eval \* args)<sup>env</sup>))

[Id name) (lookup name env)

[Fun bound-ids bound-body)

(FunV bound-ids bound-body env)]

[Call fun expr arg-exprs)

(if (= (length bound-ids) (length arg-exprs))

(eval \* bound-body

(extend\* bound-ids (eval arg-exprs env) f-env)