

## ”דף נוסחאות” שפות תכנות

### 1 הגדרות

#### כללי

1. רקורסיית זנב: חישוב ושליחה חוזרת עד לתנאי עצירה בדרך כלל רקורסיית זנב דורשת פונקציית עזר
2. ambiguity - קיימת מילה בשפה שניתן לגזור אותה לפי שני עצי גזירה שונים
3. syntactic sugar: מקרה בו יש לי בשפה פתרון לפעולה מסוימת אבל אני רוצה לפשט אותה (=”להמתיק”) ולכן ממציא תחביר חלופי שעושה את אותו דבר.

#### 1.0.1 with

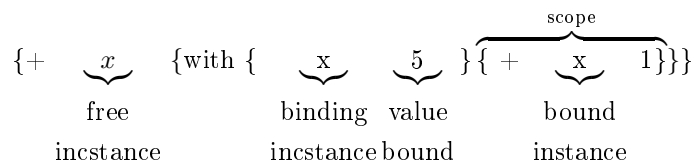
**הגדרה פורמלית Compositionality**:  $eval$  על קודקוד  $T$  עם בנים  $T_1, ..T_k$  יבצע :

$$1. v_1 = eval(T_1), v_2 = eval(T_2) \dots v_k = eval(T_k)$$

2. יבצע פעולה מקומית על  $v_1, \dots, v_k$

#### החלפת משתנים הגדרות

- Binding instance - מופע של  $x$ , שבו אני מקשר את  $x$  לביטוי כלשהו (הצהרה על שם מזהה  $x$ )
- Scope - עבור Binding instance אותו חלק בקוד שבו כל מופע של  $x$  מקושר למופע ההצהרתי הזה
- Bound instance - כל מופע של  $x$  שאינו Binding instance וגם נמצא בתוך  $scope$  של Binding instance
- Free instance - כל מופע של  $x$  שאינו Binding instance וגם אינו Bound instance



#### סימונים

1. החלף את המופעים המתאימים של  $name(=x)$  בערך  $v$  בתוך  $body(=e)$

$$[x/v]e \iff [name/v]body$$

(א) הסינטקס:

#### 1.0.2 פונקציות

1. פונקציית (first order) - אובייקט שיועד לבצע חישוב, לא יכול לשמור  $data$
2. פונקציית (high order) - אובייקט יותר מורכב שניתן להביע איתו חישובים ותהליכים ברמה גבוהה וקריאה, בעל אפשרות של שליחה וקבלת פונקציה כמשתנה.

3. פונקצית (First class) - שפות בהן פונקציה היא טיפוס כמו כל טיפוס אחר  $\Leftarrow$  נותן לנו את האפשרות לפונקציה ללא שם (אנונימית)

(א) מאפשר להגדיר פונקציה בזמן ריצה

(ב) יכולות להשמר במבנה נתונים, ועוד.

- Static Scope (also called Lexical Scope):

– In a language with static scope, each identifier gets its value from the scope in which it was defined (not the one in which it is used).

– הערכים מוגדרים בשלב ההגדרה.

- Dynamic Scope:

– In a language with dynamic scope, each identifier gets its value from the scope of its use (not its definition).

– הערכים מוגדרים בשלב השימוש.

closure:

```
param-name : 'x
body: (Mul (Id 'x) (id 'y))
env: (list ('y (Num 4)))
```

## 2 הערות parse

1. *With*

*name* (א)

*val\named* (ב)

*body* (ג)

2. *Fun* :

*name\arg* (א)

*body* (ב)

3. *Call* :

*fun name* (א)

*fun arg\param* (ב)

## 3 אלגוריתמים ב eval

$r$  : היא הקריאה ה  $r$

הערה חשובה: המספור  $r, r+1, r+2$  הוא של פעולות נדרשות ביחס לבנאי מסוים, אבל בעצים מורכבים יתכנו הפרשים גדולים יותר במחסנית הקריאות

## מודל ההחלפה:

1. עבור  $AST_r = (Num\ n)$  :

• נחזיר את  $(Num\ n) \iff Res_r = (Num\ n)$

2.  $AST_r = (Id\ name)$

• בשלב זה אסור לקבל  $Id \iff$  נחזיר שגיאה

3.  $AST_r = (OP\ l\ r)$

• גורם לשתי קריאות נוספות ל  $eval$

– נקרא ל  $eval(l) \iff AST_{r+1}(l)$

– נקרא ל  $eval(r) \iff AST_{r+2}(r)$

•  $Res_r$  מחכה לתוצאה

4.  $AST_r = (Fun\ b - id\ b - body)$

• מחזיר את הפונקציה  $RES_r = (Fun....) \iff$

5.  $AST_r = (With\ b - id\ named\ b - body)$

• גורם לשתי קריאות נוספות ל  $eval$

• גורם לקריאת  $Subst$  על  $b - body$

– נקרא ל  $eval(named) \iff AST_{r+1}(named)$

– נקרא ל  $subst$  עם  $\left\{ \begin{array}{l} b - body \rightarrow expr \\ b - id \rightarrow from \\ eval(named) \rightarrow to \end{array} \right\}$ , החלפה תתבצע ל:

$named *$

\*  $b - body$  רק אם  $b - id \neq from$   $\iff$  כן להחליף + לא לדרוס  $Scope$  עם שם משתנה דומה  $\iff$  רק את ה

$bound$  שנמצא בתוך ה  $scope$  (

– קרא ל  $eval(b - body)$  על ה  $body$  המעודכן  $AST_{r+2}(b - body) \iff$

•  $Res$  מחכה לתוצאה

6.  $AST_r = (Call\ f - expr\ arg)$

• גורם לשלוש קריאות נוספות ל  $eval$

• קריאה ל  $subst$  על ה  $b - body$

– וידוא ש  $f - expr$  הוא פונקציה  $eval(f - expr\ env_r) \iff AST_{r+1}(f - expr\ env_r)$

\* אם לא - מחזירים שגיאה

\* אם כן נשתמש בפרמטים  $b - id, b - body, \text{ ו:}$

– נקרא ל  $eval(arg)$  עם  $env_r \iff AST_{r+2}(arg\ env_r)$

– נקרא ל  $subst$  עם  $\left\{ \begin{array}{l} b - body \rightarrow expr \\ b - id \rightarrow from \\ eval(arg) \rightarrow to \end{array} \right\}$ , החלפה תתבצע ל:

$arg *$

\*  $b - body$  רק אם  $b - id \neq from$   $\iff$  כן להחליף + לא לדרוס  $Scope$  עם שם משתנה דומה  $\iff$  רק את ה

$bound$  שנמצא בתוך ה  $scope$  (

– נקרא ל  $eval(b - body)$  על ה  $body$  המעודכן  $AST_{r+3}(b - body) \iff$

## מודל ה-SC:

1. עבור  $AST_r = (Num\ n)$  :

• נחזיר את  $n$   $Res_r = n \iff$

•  $SC$  לא משתנה

2.  $AST_r = (Id\ name)$  :

• נחזיר את  $lookup(name)$  עבור ה- $SC_r$  המתאים (מושפע מעץ הקריאות)  $Res_r = lookup(name\ SC_r) \iff$

•  $SC$  לא משתנה

3.  $AST_r = (OP\ l\ r)$  :

• גורם לשתי קריאות נוספות ל- $eval$

– נקרא ל- $eval(l)$  עם  $SC_r$   $AST_{r+1}(l) \iff$

– נקרא ל- $eval(r)$  עם  $SC_r$   $AST_{r+2}(r) \iff$

•  $Res_r$  מחכה לתוצאה

•  $SC$  לא משתנה

4.  $AST_r = (Fun\ b - id\ b - body)$  :

• מחזיר את הפונקציה  $RES_r = (Fun....b - body) \iff$

•  $SC_r$  לא משתנה

5.  $AST_r = (With\ b - id\ named\ b - body)$  עם  $SC_r$  :

• גורם לשתי קריאות נוספות ל- $eval$

– נקרא ל- $eval(named\ env_r)$   $AST_{r+1}(named) \iff$

– נרחיב את את  $SC_r$  במעבר ל- $AST_{r+2}$  עם הזוג:  $(b - id)$ , התוצאה על  $(named)$

– נקרא ל- $eval(b - body\ SC_{r+2})$   $AST_{r+2}(b - body) \iff$

•  $Res_r$  מחכה לתוצאה

6.  $AST_r = (Call\ f - expr\ arg)$  עם  $SC_r$  :

• גורם לשלוש קריאות נוספות ל- $eval$

– וידוא ש  $f - expr$  הוא פונקציה  $AST_{r+1}(f - expr\ sc_r) \iff eval(f - expr\ sc_r)$

\* אם לא - מחזירים שגיאה

\* אם כן נשתמש בפרמטים  $b - id, b - body$  ו:

– נקרא ל- $eval(arg)$  עם  $sc_r$   $AST_{r+2}(arg\ sc_r) \iff$

– נרחיב את  $sc_r$  במעבר ל- $ENV_{r+3}$  עם הזוג:  $(b - id)$ , התוצאה על  $(arg)$

– נקרא ל- $eval(b - body\ Extend(SC))$   $AST_{r+3}(b - body) \iff$

•  $Res$  מחכה לתוצאה

## מודל הסביבות:

1. עבור  $AST_r = (Num\ n)$  :

- נחזיר את  $(NumV\ n)$ , כלומר  $Res_r = (NumV\ n)$
- $ENV$  לא משתנה

2.  $AST_r = (Id\ name)$

- נחזיר את  $lookup(name)$  עבור ה  $ENV_r$  המתאים (מושפע מעץ הקריאות)  $Res_r = lookup(name\ env_r) \iff$
- $ENV$  לא משתנה

3.  $AST_r = (OP\ l\ r)$

- גורם לשתי קריאות נוספות ל  $eval$
- נקרא ל  $eval(l)$  עם  $ENV_r$   $AST_{r+1}(l) \iff$
- נקרא ל  $eval(r)$  עם  $ENV_r$   $AST_{r+2}(r) \iff$

- $Res_r$  מחכה לתוצאה
- $ENV$  לא משתנה

4.  $AST_r = (Fun\ b - id\ b - body)$

- מחזיר את הפונקציה עם ה  $env_r$   $RES_r = (FunV...env_r) \iff$
- $ENV_r$  לא משתנה

5.  $AST_r = (With\ b - id\ named\ b - body)$ , עם  $env_r$

- גורם לשתי קריאות נוספות ל  $eval$
- נקרא ל  $eval(named\ env_r)$   $AST_{r+1}(named) \iff$
- נרחיב את את  $ENV$  במעבר ל  $ENV_{r+2}$  עם התוצאה על  $named$
- נקרא ל  $eval(b - body\ env - r + 1)$   $AST_{r+2}(b - body) \iff$
- $Res$  מחכה לתוצאה

6.  $AST_r = (Call\ f - expr\ arg)$

- גורם לשלוש קריאות נוספות ל  $eval$
- וידוא ש  $f - expr$  הוא פונקציה  $AST_{r+1}(f - expr\ env_r) \iff eval(f - expr\ env_r)$
- \* אם לא - מחזירים שגיאה
- \* אם כן נשתמש בפרמטים  $f - env, b - id, b - body$  ו:
- נקרא ל  $eval(arg)$  עם  $env_r$   $AST_{r+2}(arg\ env_r) \iff$
- נרחיב את  $f - env$  במעבר ל  $ENV_{r+3}$  עם התוצאה על  $arg$
- נקרא ל  $eval(b - body\ Extend(f - env))$   $AST_{r+3}(b - body) \iff$
- $Res$  מחכה לתוצאה