**שאלה 5**

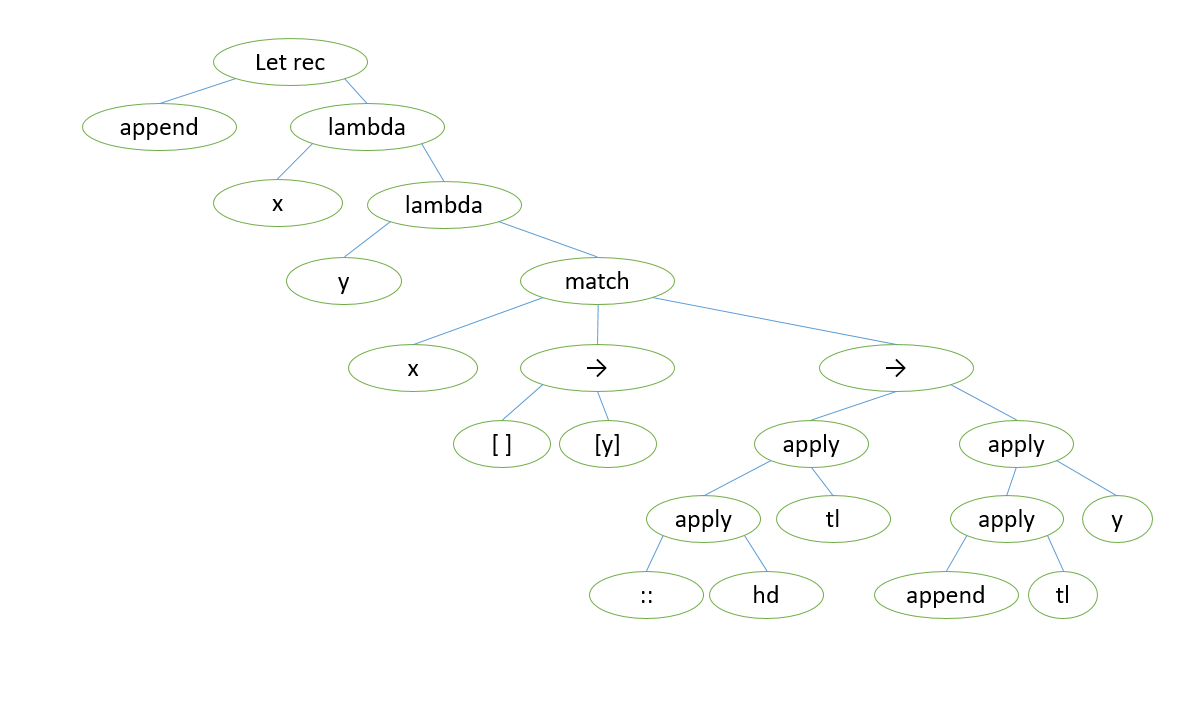
**סעיף a**

נבצע את תהליך הסקת הנתונים על הפונקציה הנתונה.

1. בניית parse tree

מבנה חולית let rec – כמו שראינו בתרגול, הבן השמאלי הוא שם הפונקציה והבן הימני הוא ביטוי הלמדה שמגדיר את הפונקציה.

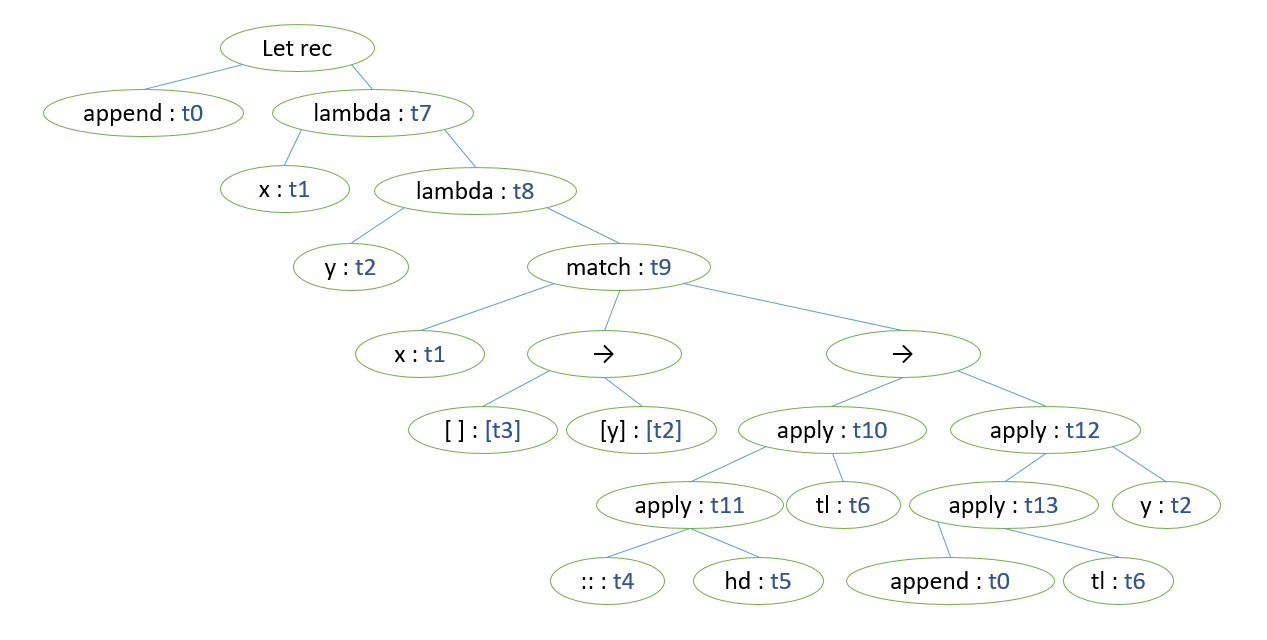
מבנה חולית match – לא ראינו בכיתה ולכן החלטנו להגדיר זאת באופן הבא: הבן השמאלי הוא המשתנה עליו מבצעים את הpattern matching, שאר הבנים הם אפשרויות להתאמה. כל אפשרות כזאת מיוצגת ע"י חולית "<-", שהבן השמאלי שלה הוא התאמה לpattern והבן הימני הוא הערך שה-match יחזיר באפשרות זו.



2. הוספת טיפוסים לעץ

נוסיף טיפוסים לחוליות.

(לחוליות ה-"<-" לא נוסיף טיפוס. בשלב הבא נדאג לקשר בין הטיפוסים של החוליות השונות בתת העץ match, על פי הסמנטיקה של match).



3. יצירת אילוצים

t0 = t7 // let rec

t7 = t1 -> t8 // lambda expression

t8 = t2 -> t9 // lambda expression

t9 = [t2]

match result

t9 = t12

t1 = [t3]

match pattern

t1 = t10

t11 = t6 -> t10 // application

t4 = t5 -> t11 // application

t4 = a -> a -> [a] // :: is a built-in operator

t13 = t2 -> t12 // application

t0 = t6 -> t13 // application

4. פתרון האילוצים באמצעות unification

נפתור את המשוואות ונקבל:

t0 = [a] -> t2 -> [t2]

t1 = [a]

t3 = a

t4 = a -> [a] -> [a]

t5 = a

t6 = [a]

t7 = [a] -> t2 -> [t2]

t8 = t2 -> [t2]

t9 = [t2]

t10 = [a]

t11 = [a] -> [a]

t12 = [t2]

t13 = t2 -> [t2]

5. מציאת הטיפוס של הפונקציה הנתונה

הטיפוס של append הוגדר להיות t0, ולכן:

append : [a] -> t2 -> [t2]

או אם נסמן t2 = b בשביל האחידות:

append: [a] -> b -> [b]

כאשר a, b הם טיפוסים כלליים.

**סעיף b**

כן, יש משהו שמצביע על טעות. אמנם הצלחנו להסיק את הטיפוס של append, אבל קיבלנו שהיא פונקציה שמקבלת רשימה עם איברים מטיפוס a ואיבר נוסף מטיפוס b. append אמורה לשרשר איבר לרשימה, ולכן היינו מצפים לקבל append: [a] -> a -> [a]. כמו כן, גם בלי לדעת מה מטרת הפונקציה append, זה נראה משונה שהטיפוס שהיא מחזירה כלל לא מתייחס לטיפוס של הארגומנט.

**סעיף c**

תיקון הפונקציה:

let rec append x y =

match x with

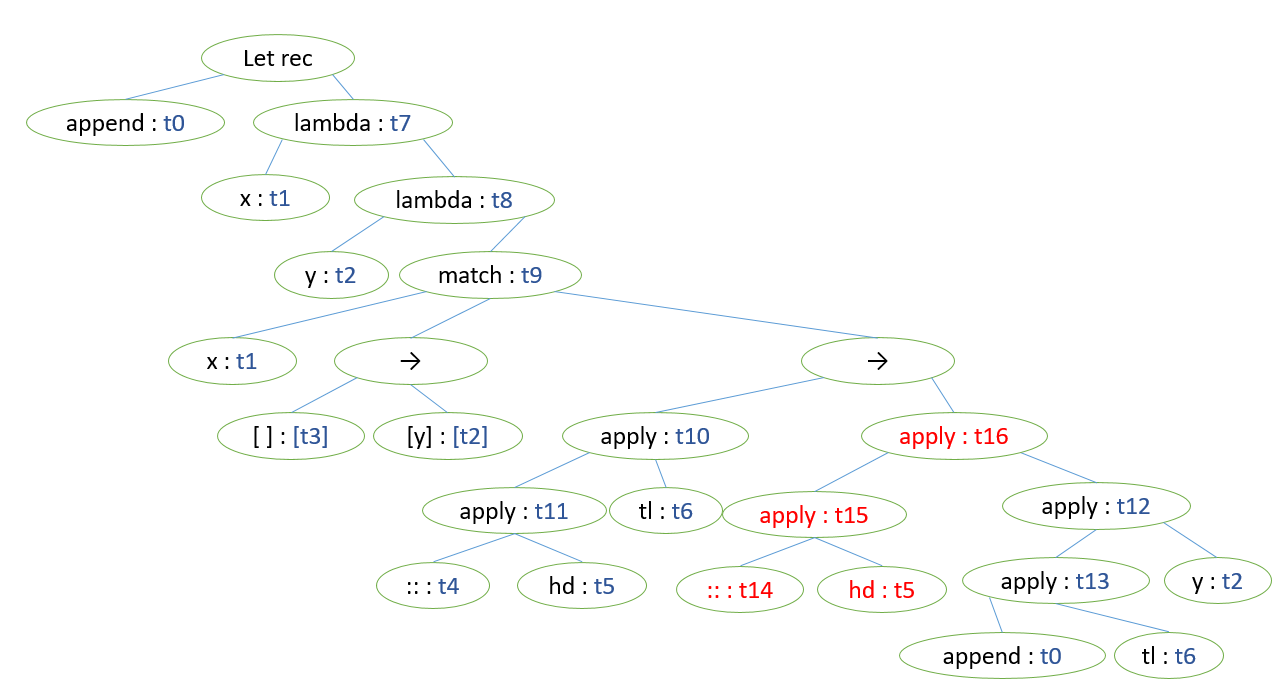
| [ ] -> [y]

| hd :: tl -> **hd ::** (append tl y)

**סעיף d**

לאחר התיקון נקבל:

append: [a] -> a -> [a]

ההבדל בהרצה של אלגוריתם הסקת הטיפוסים הוא שלאחר התיקון נקבל עץ גדול יותר:

נקבל את האילוץ t15 = t12 -> t16 (application), וכן t15 = [a] -> [a] (בגלל ה-:: והטיפוס של hd שנשאר כמו מקודם). בסך הכל נסיק ש-t12 = [a].

תת העץ של t12 נשאר כפי שהוא ולכן נקבל עדיין t12 = [t2]. מכאן ש-t2 = a.

לכן t0 = [a] -> t2 -> [t2] מסעיף a יהפוך לt0 = [a] -> a -> [a], כפי שרצינו.