a.1.1.   נאמר ששתי רשימות עצלות, lzl1, lzl2 שקולות אמ"מ לכל n טבעי מתקיים:

(take lzl1 n) = (take lzl2 n)

b.1.1.   נוכיח באינדוקציה על n כי

(take even-square-1 n) = (take even-square-2 n)

מקרה בסיס: n = 0

נשים לב שעבור 0, תנאי הבסיס של take, הוא יחזיר רשימה ריקה בשני המקרים ולכן נקבל שוויון.

הנחה: נניח שעבור n כלשהו הטענה נכונה.

צעד:

נוכיח ל n +1 כאשר n > 0.

עבור (take even-square-1 n+1) נקבל

(cons (head even-square-1) (take (tail even-square-1) (- n+1 1))))

כלומר

(cons (head even-square-1) (take (tail even-square-1) (n))))

עבור (take even-square-2 n+1) נקבל

(cons (head even-square-2) (take (tail even-square-2) (- n+1 1))))

כלומר

(cons (head even-square-2) (take (tail even-square-2) (n))))

מהנחת האינדוקציה ידוע כי

(take (tail even-square-1) (n) = (take (tail even-square-2) (n)

נזכר בפונקציה take:

```
(define take
        (lambda (lz-lst n)
                (if (or (= n 0) (empty-lzl? lzl-lst))
                        empty-lzl
                        (cons (head lz-lst)  (take (tail lz-lst) (- n 1)) )
                )
        )
)
```

נשים לב שמהנחת האינדוקציה אנחנו מקבלים כי הרשימות שוות עבור ההתחלה באורך n שלהן, כלומר עבור אינדקסים 1 עד n (בהנחה ואנו מתחילים למנות אינדקים מ 1 ולא מ 0).

נותר כעת להראות שהאיבר ה 1 + n בהן זהה.

על מנת לעשות זאת נתבונן בפונקציה take שמתוארת לעיל בקריאה ה 1 + n שלה לעצמה.

הפונקציה take יוצאת זוג שבנוי מ ראש הרשימה: (head), ושאר הרשימה: tail lz-lst 0

(הקריאה הבאה מתבצעת עם 0 כי הפעלנו את take מספר פעמים ששווה ל 1 + n ואנחנו בקריאה ה n+1)

לכן נקבל בשתי הרשימות את הזוג הסדור: האיבר ה n +1 במקום הראשון בזוג, עם רשימה ריקה במקום השני בזוג.

נותר להראות שהאיבר הראשון בזוג שווה בשתי הרשימות.

יהי X האיבר ה 1 + n ברשימה integers.

נתבונן בפונקציה even-square-2 על X:

אם X זוגי נרשום את X בתור 2x כך ש X = 2x.

לאחר בפעולת map נקבל 4*(x^2) = X^2

נשים לב שזהו מספר זוגי ולכן נקבל ש filter יחזיר אותו ב head.

לכן take יחזיר את הזוג הסדור עם: 4*(x^2) במקום הראשון ואילו empty-lzl במקום השני.

כלומר האיבר ה n+1 ברשימה 1 הוא 4*(x^2).

אם X אי זוגי לאחר הפעלת map נקבל X^2 ובגלל ש X אי זוגי גם X בחזקת 2 אי זוגי (לא קיים גורם ראשוני זוגי (2) ב X ולכן גם לא קיים גורם כזה במכפלת X עם עצמו).

לכן filter יחזיר את המשך הרשימה ונקבל שעבור המקום ההפעלה ה 1 + n ב take לא יוחזר מספר.

נתבונן כעת ב even-square-2 על X:
אם X זוגי.
לאחר הפעלת filter נקבל את X גם כן כי הוא מתחלק ב 2.
נפעיל עליו את הפונקציה של map ונקבל את 2^(X) כלומר האיבר ה 1 + n שווה בשתי הרשימות המתקבלות מ take במצב זה.
אם X אי זוגי לאחר הפעלת filter נקבל פשוט את המשך הרשימה וכיוון שאנו בפעולה ה 1 + n נקבל שהפעלה זו לא מחזירה איבר, כלומר האיבר, ה 1 + n שמוחזר מ take שווה בשתי הרשימות המתקבלות מ take במצב זה.

בשני המקרים קיבלנו כי האיבר ה 1 + n המתקבל ברשימה שנוצרת על ידי take של שתי הרשימות שווה ולכן מעיקרון האינדוקציה נקבל ש take עבור כל n מחזיר את אותה רשימה משתי הרשימות הללו. לכן לפי הגדרתנו בסעיף הקודם הרשימות שקולות.

.a.1.3

Unify [t (s (s) , G , H , p, t (E) , s) , t (s (H) , G , p , p, t (E) , K)]

ניצור משוואה:

t (s (s) , G , H , p, t (E) , s) = t (s (H) , G , p , p, t (E) , K)

ונריץ את אלגוריתם פתרון המשוואות:

אתחול:

- אוסף משוואות:

[t (s (s) , G , H , p, t (E) , s) = t (s (H) , G , p , p, t (E) , K)]

הצבה:

{}

בחירת משוואה מהאוסף:

t (s (s) , G , H , p, t (E) , s) = t (s (H) , G , p , p, t (E) , K)

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[s (s) = s (H), G = G, H = p, p = p, t (E) = t (E), s = K]

הצבה:

{}

בחירת משוואה מהאוסף:

s = K

אחד הצדדים הוא משתנה לוגי, הוספת המשוואה להצבה

{} * {K = s} = {K = s}

- אוסף משוואות:

[s (s) = s (H), G = G, H = p, p = p, t (E) = t (E)]

הצבה:

{s = K}

בחירת משוואה מהאוסף:

p = p

הערכים האטומיים שווים ולכן נמשיך

- אוסף משוואות:

[s (s) = s (H), G = G, H = p, t (E) = t (E)]

הצבה:

{s = K}

בחירת משוואה מהאוסף:

G = G

הערכים הלוגיים שווים ולכן נמשיך

- אוסף משוואות:

[s (s) = s (H), H = p, t (E) = t (E)]

הצבה:

{s = K}

בחירת משוואה מהאוסף:

H = p

צד אחד משתנה לוגי והשני אטומי, נוסיף לסביבת ההצבה

- אוסף משוואות:

[s (s) = s (p), t (E) = t (E)]

הצבה:

{K = s, H = p}

בחירת משוואה מהאוסף:

t (E) = t (E)

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[s (s) = s (H), E = E]

הצבה:

{K = s, H = p}

בחירת משוואה מהאוסף:

E = E

הערכים הלוגיים שווים ולכן נמשיך

- אוסף משוואות:

[s (s) = s (H)]

הצבה:

{K = s, H = p}

בחירת משוואה מהאוסף:

s (s) = s (H)

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[s = H]

הצבה:

{K = s, H = p}

בחירת משוואה מהאוסף:

s = H

נבצע הרכבה מהסביבה:

s = p

שני הצדדים הם ערכים אטומיים , אך אינם שווים ולכן נחזיר Fail

בחירת משוואה מהאוסף:

s (s) = s (H)

אוסף המשוואות:

[s = H]

b.1.3.

Unify [g (c , v (U) , g , G , U , E , v (M)) , g (c , M , g ,v (M) , v (G) , g , v (M)]

ניצור משוואה:

g (c , v (U) , g , G , U , E , v (M)) = g (c , M , g ,v (M) , v (G) , g , v (M)

ונריץ את אלגוריתם פתרון המשוואות:

אתחול:

- אוסף משוואות:

[g (c , v (U) , g , G , U , E , v (M)) = g (c , M , g ,v (M) , v (G) , g , v (M)]

הצבה:

{}

בחירת משוואה מהאוסף:

g (c , v (U) , g , G , U , E , v (M)) = g (c , M , g ,v (M) , v (G) , g , v (M)

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[c = c, v (U) = M, g = g, G = v (M), U = v (G), E = g, v (M) = v (M)]

הצבה:

{}

בחירת משוואה מהאוסף:

g = g

הערכים האטומיים שווים ולכן נמשיך

- אוסף משוואות:

[c = c, v (U) = M, G = v (M), U = v (G), E = g, v (M) = v (M)]

הצבה:

{}

בחירת משוואה מהאוסף:

c = c

הערכים האטומיים שווים ולכן נמשיך

- אוסף משוואות:

[v (U) = M, G = v (M), U = v (G), E = g, v (M) = v (M)]

הצבה:

{}

בחירת משוואה מהאוסף:

E = g

צד אחד משתנה לוגי והשני אטומי, נוסיף לסביבת ההצבה

- אוסף משוואות:

[v (U) = M, G = v (M), U = v (G), v (M) = v (M)]

הצבה:

{E = g}

בחירת משוואה מהאוסף:

v (M) = v (M)

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[v (U) = M, G = v (M), U = v (G), M = M]

הצבה:

{E = g}

בחירת משוואה מהאוסף:

M = M

הערכים הלוגיים שווים ולכן נמשיך

- אוסף משוואות:

[v (U) = M, G = v (M), U = v (G)]

הצבה:

{E = g}

בחירת משוואה מהאוסף:

v (U) = M

צד אחד משתנה לוגי, נוסיף לסביבת ההצבה

- אוסף משוואות:

[G = v (v (U)), U = v (G)]

הצבה:

{E = g, M = v (U)}

בחירת משוואה מהאוסף:

U = v (G)

צד אחד משתנה לוגי, נוסיף לסביבת ההצבה

- אוסף משוואות:

[G = v (M)]

הצבה:

{E = g, M = v (v (G)), U = v (G)}

בחירת משוואה מהאוסף:

G = v (v (v (G)))

לא אפשרי מצב בו משתנה S, שווה ל v(S), לכן נחזיר Fail

c.1.3.

Unify [s ([v | [[v | V] | A]]) , s ([v | [v | A]])]

ניצור משוואה:

s ([v | [[v | V] | A]]) = s ([v | [v | A]])

ונריץ את אלגוריתם פתרון המשוואות:
אתחול:

- אוסף משוואות:

[s ([v | [[v | V] | A]]) = s ([v | [v | A]])]

הצבה:

{}

בחירת משוואה מהאוסף:

s ([v | [[v | V] | A]]) = s ([v | [v | A]])

שני האגפים בעלי אותו מבנה, נפרק למשוואות קטנות יותר

- אוסף משוואות:

[[v | [[v | V] | A]] = [v | [v | A]]]

הצבה:

{}

בחירת משוואה מהאוסף:

[v | [[v | V] | A]] = [v | [v | A]]

נשווה בין האיברים בזוג

- אוסף משוואות:

[v = v, [[v | V] | A] = [v | A]]

הצבה:

{}

בחירת משוואה מהאוסף:

v = v

הערכים האטומיים שווים ולכן נמשיך

- אוסף משוואות:

[[[v | V] | A] = [v | A]]

הצבה:

{}

בחירת משוואה מהאוסף:

[[v | V] | A] = [v | A]

נשווה בין האיברים בזוג

- אוסף משוואות:

[[v | V] = v, A = A]

הצבה:

{}

בחירת משוואה מהאוסף:

[v | V] = v

נקבל שגיאה כי האיברים אינם זהים, לכן נחזיר Fail.

2. א.

נאמר שפרוצדורה f,

$f\left(x_1, \ldots, x_n\right)$ שקולה (equivalence) לגרסת ה Success-Fail Continuations שלה אם מתקיים שלכל ערכי קלט $\left(x_1, \ldots, x_n\right)$ ולכל success וfail מתקיים ש:

$$f\ \$\left(x_1,\ldots,x_n,\ succ,\ fail\right)\ = \begin{cases} succ\ \ f\left(x_1,\ldots,x_n\right) & if\ f\ was\ succesful \\ fail\ \ f\left(x_1,\ldots,x_n\right) & else \end{cases}$$

2. ד.

נתבונן בשתי הפונקציות:

$$get-value(assoc-list,\ key)\ :\ \left[List<Pair(Symbol,\ T)>\ *\ Symbol\ \rightarrow\ T\ \mid\ fail\right]$$

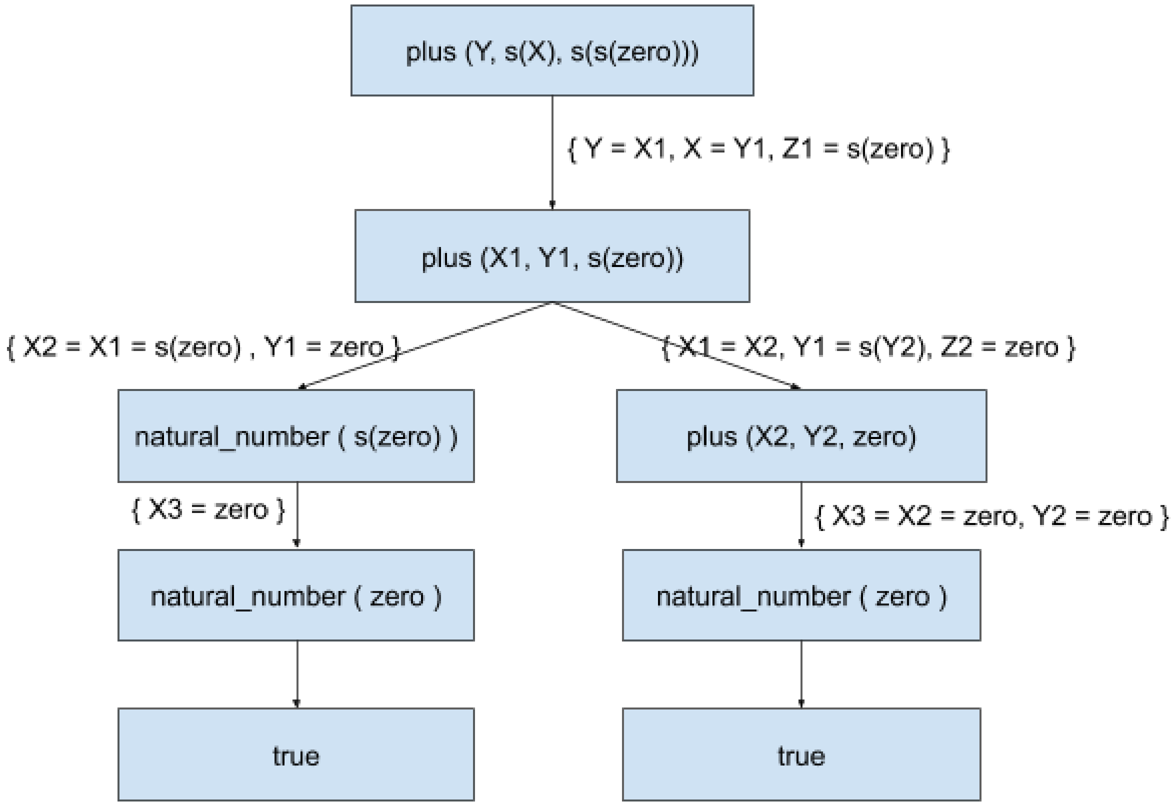
$$\begin{aligned} &\mathrm{get-value\$\ (assoc-list,\ key)\ :} \\ &\left[\mathrm{List<Pair(Symbol,\ T)>\ *\ Symbol\ *}\left[\mathrm{T\rightarrow T_1}\right]*\left[\mathrm{Empty\ \rightarrow T_2}\right]\rightarrow\ \mathrm{T_1\ \mid\ T_2}\right] \end{aligned}$$

נראה שהן שקולות לפי ההגדרה ב 2 א.

תהי L רשימה ויהי k מפתח שמתקבלים כקלט בf ו $f. נחלק למקרים:

1. לא קיים ב L ערך עם מפתח k. אז f מחזירה ערך failure וגם $f מחזירה ערך failure. והפעלת failure על תוצאת f תתן failure, כלומר השיווין מתקיים במקרה זה.
2. קיים מפתח k ברשימה L. אז f מחזירה את הערך המתאים ל k, ו $f מחזירה את הערך עם המפתח k. הפעלת succ על הערך של המפתח k תחזיר את אותו ערך ולכן גם במקרה זה השיוויון לעיל מתקיים.

3.3.a.



3.3.b.

התשובה של האלגוריתם תהיה {[Y = s(zero) , X = zero], [Y = zero, X = s(zero)]}.

3.3.c.

זהו עץ הצלחה, משום שיש בו לפחות קודקוד true אחד.

3.3.b.

זהו עץ סופי כפי שניתן לראות באיור, וכמובן שאין בו קריאות רקורסיביות של משתנה לעצמו.