מגישים:

דור כרמי – 205789662

עודד בסרבה- 315564641

עבודה 5 ב-PPL:

1.1.a:

שוויון בין רשימות עצלות מוגדר להיות:

תהי L_1, L_2 שתי רשימות עצלות.

נגדיר $L_1 = L_2$ אם מתקיים אחד מהתנאים הבאים:

1. או ש- $L_1 = L_2 = empty\ list$
2. או ש-
   a. נסמן ב- x_i את האיבר במקום ה-*i* ברשימה L_1
   b. נסמן ב- y_i את האיבר במקום ה-*i* ברשימה L_2
   c. לכן מתקיים לכל $1 \leq i \leq n$ (אם הרשימה אינסופית אז $\forall n \in \mathbb{N}$):
   d. $x_i = y_i$

1.1.b:

נוכיח שוויון בין הרשימות *even-square-1, even-square-2*.

נוכיח באינדוקציה על האיברים ברשימה לפי סדר הרשימה.

טענה:

הפרוצדורה *even-square-1* שקולה לפרוצדורה *even-square-2*.

בסיס:

בסיס האינדוקציה: *n=1*.

יהי x_1 האיבר הראשון ברשימה L_1, ויהי y_1 האיבר הראשון ברשימה L_2.

אז מתקיים עבור *even-square-1* ש:

האיבר הראשון ברשימה הוא האיבר הראשון מה-*lzl-lst* שהיא *integers-from 0*, אשר בוצע עליו *map* עם הפונקציה *x*x* ולאחר מכן הוא עבר סינון של *filter* עם ה-*predicat*:

x %2 =0

מאחר ועבור המספרים הטבעיים של הרשימה העצלה *integers-from 0*, הפעלה של *x*x* על 0 ייתן 0, וה-*predicat* על 0 יהיה *true* נקבל ש-0 הוא האיבר הראשון ברשימה *even-square-1*.

עבור *even-square-2* מתקיים:

האיבר הראשון ברשימה הוא האיבר הראשון מה-*lzl-lst* שהיא *integers-from 0*, אשר עבר את הסינון של *filter* עם ה-*predicat x % 2=0*, ולאחר מכן בוצע עליו *map* עם הפונקציה *x*x*.

מאחר והאיבר הראשון ברשימת הטבעיים שמקיים את התנאי *x%2=0* הוא המספר 0, והפעלה של הפונקציה על האיבר הזה מביאה לתוצאה של 0, נקבל ש-0 הוא האיבר הראשון ברשימה *even-square-2*.

קיבלנו ששני האיברים הראשונים בשתי הרשימות זהים, לכן תנאי הבסיס מתקיים.

הנחת האינדוקציה: נניח שעבור k מתקיים שלכל איבר $x_k = y_k$ (כלומר האיברים באידנקסים עד *k* שווים בשתי הרשימות).

צעד: נוכיח עבור *k+1*.

יהיה האיבר ה-*k+1* ברשימה *even-square-1*.

נסתכל על האיבר ה-*k* ברשימה. מתקיים:

$$(x_k = k * k) \cap ((k * k)\%2) = 0$$

נסתכל על האיבר הבא בסדר המספרים הטבעיים: $k + 1$.

מתקיים $(k + 1)(k + 1)\%2 = 1$, שכן כפל של מספר אי זוגי במספר אי זוגי נותן מספר אי זוגי. ולכן בהכרח מתקיים ש- $x_{k+1} = (k + 2)(k + 2)$ משום שקודם מתבצעת ההפעלה של הפונקציה ב-*map* של *x*x* ולאחר מכן מתבצע ה-*filter*, ולא קיים איבר קטן מ-*k+2* שעבורו זה מתקיים.

יהיה האיבר ה-*k+1* ברשימה *even-square-2*.

נסתכל על האיבר ה-*k* ברשימה. מתקיים:

$$(y_k = k * k) \cap (k\%2) = 0$$

נסתכל על האיבר הבא בסדר המספרים הטבעיים: $k + 1$.

מתקיים $(k + 1)\%2 = 1$, שכן העוקב של מספר זוגי *k* הוא אי זוגי. ולכן בהכרח מתקיים ש- $y_{k+1} = (k + 2)(k + 2)$ משום שלפי מה שהבחנו, *k+2* הוא מספר זוגי, ולכן יעבור את ה-*predicate* של ה-*filter*, ולאחר מכן תתבצע עליו ההפעלה של הפונקציה *x*x* של *map*.

קיבלנו ש:

$$x_{k+1} = (k + 2)(k + 2) = y_{k+2}$$

כלומר הוכחנו את הטענה לכל באינדוקציה לכל המספרים הטבעיים ולכן הטענה מתקיימת.

<u>:2.a</u>

נסתכל על הפונקציה *f1#* (סימון לפונקציה *f1* בגרסת ה-*Success,Fail continuation*) ועל טיפוסי ההחזרה האפשריים שלה:

$$G_{succ}, G_{fail}$$

פרוצדורה $f1$ שקולה לפרוצדורה $f1\#$ אם לכל ערכי קלט $x_1 \dots x_n$ שהטיפוסים שלהם הם: $T_1 \dots T_n$, ולכל שתי פונקציות:

$$success: T_i \rightarrow G_{succ}$$

$$Fail: T_i \rightarrow G_{fail}$$

מתקיים:

- במקרה של הצלחה:

$$(f_1\#\ x_1 \dots x_n\ succ\ fail) = (succ\ (f_1\ x_1 \dots x_n))$$

- במקרה של כישלון:

$$(f_1\#\ x_1 \dots x_n\ succ\ fail) = \left(fail\ (f_1\ x_1 \dots x_n)\right)$$

:2d

נוכיח באמצעות התנאי שכתבנו לעיל שמתקיים: **get-value** equivalent to **get-value$**

כיוון שהפרוצדורה *get-value* היא רקורסיבית, ההוכחה מתבצעת ע"י שימוש באינדוקציה על אורך הרשימה.

בסיס האינדוקציה: *n=0*. $list = ()'$, *key=anyKey*.

במקרה כזה תנאי הבסיס של הרקורסיה מתקיים, הרשימה ריקה, ולכן מוחזר *'fail* בפרוצדורה *get-value*.
בפרוצדורה *get-value$* מתקיים גם כן תנאי הבסיס של הרקורסיה, הרשימה ריקה, ולכן תופעל הפרוצדורה *fail* שלא מקבלת פרמטר ומחזירה *'fail*, ולכן הטיפוסי החזרה וערכי ההחזרה של הפעלת שתי הפרוצדורות על רשימה ריקה זהים.

הנחת האינדוקציה: עבור $|list| = n \le k$ הטענה נכונה, כלומר:

$$(f_1\#\ x_1 \dots x_k\ succ\ fail) = \left(fail\ (f_1\ x_1 \dots x_k)\right)$$

צעד: נוכיח עבור n=k+1.

נסתכל על הרשימה $x_1 \dots x_k, x_{k+1}$.

בשתי הפונקציות בכל שלב בוחנים את האיבר הראשון של הרשימה *assoc-list* אל מול ה-*key*.

לכן נחלק למקרים:

מקרה א':

אם $(equal\ ?\ \ caar\ assoc-list) = true$, כלומר האיבר הראשון של הזוג הראשון ברשימה שווה ל-*key*, יתקיים:

$get-value\ (x_1 \dots x_{k+1}, key) \overset{*}{\Rightarrow} (cdr\ (car\ assoc-list))$ כלומר מתקיים שהאיבר שחוזר הוא האיבר השני של הזוג הראשון ברשימה.

בפרוצדורה השנייה מתקיים:

$$get-value\$\ (x_1 \dots x_{k+1}, key\ succ\ fail) \overset{*}{\Rightarrow} (succ(cdr\ (car\ assoc-list)))$$

ועבור *success* שנגדירה להיות פונקצית הזהות, נקבל:

$$\left(succ\left(cdr\ (car\ assoc-list)\right)\right) = \left(cdr\ (car\ assoc-list)\right)$$

וקיבלנו את אותו ערך החזרה ואותו טיפוס החזרה.

מקרה ב':

מתקיים שהאיבר הראשון בזוג הראשון לא שווה ל-key. במקרה כזה:

$$get-value\ (x_1 \dots x_{k+1}, key) \overset{*}{\Rightarrow} get-value(cdr\ (car\ assoc-list), key) \overset{*}{\Rightarrow}$$
$$get-value\ (x_2 \dots x_{k+1}, key)$$

$$get-value\$\ (x_1 \dots x_{k+1}, key) \overset{*}{\Rightarrow} get-value\$(cdr\ (car\ assoc-list), key, succ, fail) \overset{*}{\Rightarrow}$$
$$get-value\$\ (x_2 \dots x_{k+1}, key, succ, fail)$$

ומהנחת האינדוקציה נקבל ש:

$$get-value\ (x_2 \dots x_{k+1}, key) = get-value\$\ (x_2 \dots x_{k+1}, key, succ, fail)$$

:3.1

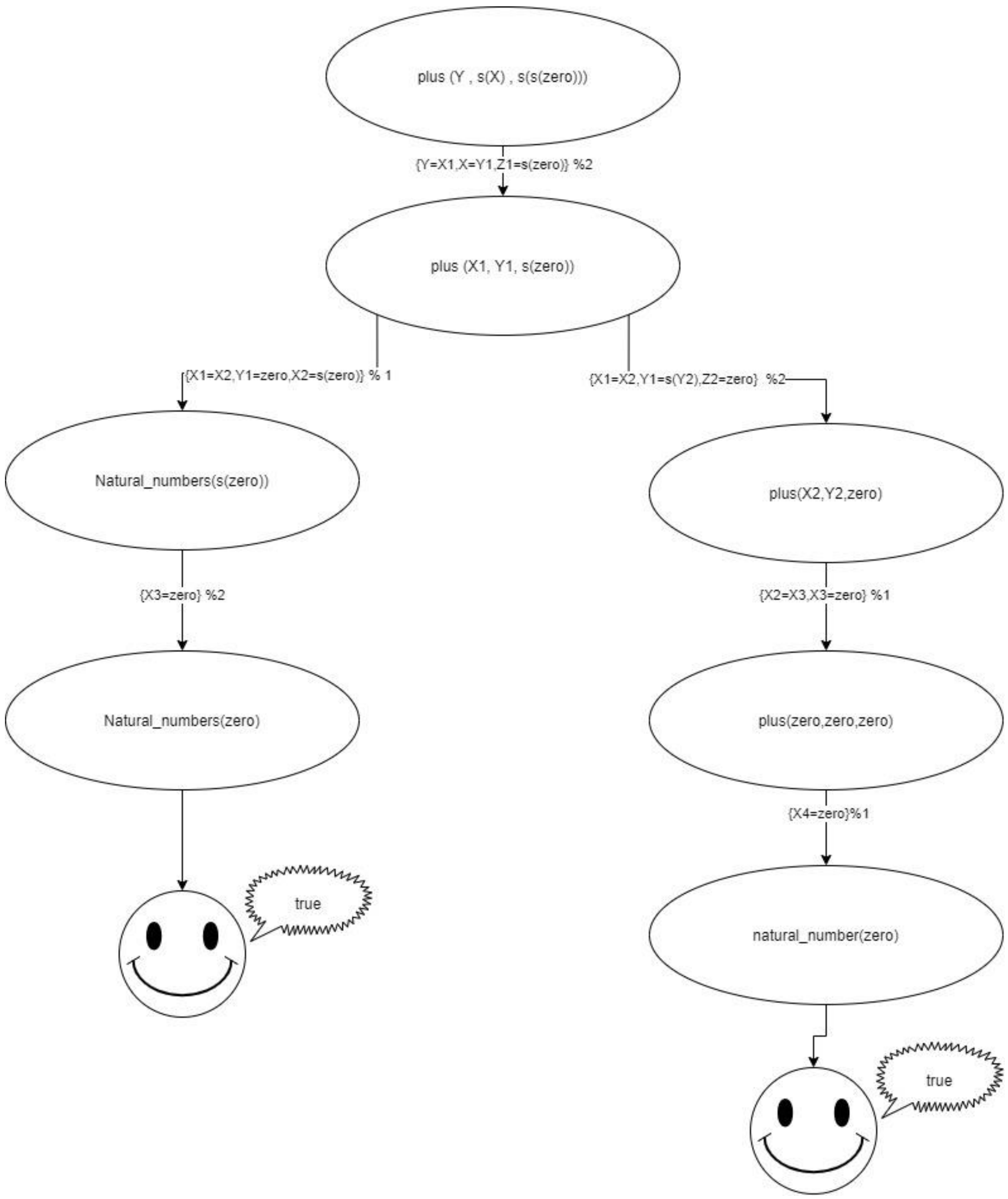
Task	Substitution	Equations	notes
a		t(s(s)),G,H,p,t(E),s)= t(s(H)),G,p,p,t(E),K)	
		t(s(s))=t(s(H)) G=G H=p p=p t(E)=t(E) K=s	
	H=p	s(s)=s(H) G=G p=p E=E K=s	
	H=p	H=s G=G p=p E=E K=s	Failure- H=p and H=s ➔not possible

Task	Substitution	Equations	notes
b		g(c,v(U),g,G,U,E,v(M))= g(c,M,g,v(M),v(G),g,v(M))	
		c=c g=g M=v(U) G=v(M) U=v(G)	

		E=g v(M)=v(M)	
		G=v(M)=v(v(U))=v(v(v(G)))…	Infinite loop

Task	Substitution	Equations	notes
c		s([v | [[v|V]|A]])= s([v| [v|A]])	
		[v|A]=[[v|V]|A]	
		v=[v|V]	Atomic cannot be compound expression

:3.3



סעיף b:

התוצאות הן:

$X=zero$	$Y=s(zero)$
$X=s(zero)$	$Y=zero$

סעיף c:

זהו עץ הצלחה משום שקיים מסלול משורש העץ לאחד העלים שמסתיים ב-*true*.

סעיף d:

העץ הספציפי שיצרנו הינו עץ סופי, משום שלכל מסלול אפשרי בעץ קיים עלה, כלומר מסתיים.