חלק 1

**שאלה 1**

**(a) {f : [T1 → T2], g : [T1 → T2], a : T1} |- (f (g a)) : T2**

**הטענה לא נכונה.**

הפעלת (g a) מחזירה טיפוס מסוג T2 כאשר הפונקציה f צריכה לקבל משתנה מסוג T1.

נשים לב שהטענה נכונה עבור המקרה הפרטי בו

**(b) {x : T1, y : T2, f : [T2 → T1]} |- (f y) : T1**

**הטענה נכונה.**

הפונקציה f מקבלת כפרמטר משתנה מסוג T2, כלומר ט מקיים את הדרישה כיוון שהוא מאותו הטיפוס, ולכן f תחזיר משתנה מסוג T1 כנדרש.

**(c) {f : [T1 → T2]} |- (lambda (x) (f x)) : [T1 → T2]**

**הטענה נכונה.**

נשים לב שהפעלת f על x מחזירה ביטוי מטיפוס T2 והפונקציה f מקבלת פרמטר מטיפוס T1. לכן נוכל להסיק שהטיפוס של x הוא T1 ומתקיים כי טיפוס ה-clouser כולו הוא [T1 → T2].

**(d) {f : [T1 ∗ T2 → T3]} |- (lambda (x) (f x 100)) : [T1 → T3]**

**הטענה נכונה.**

נשים לב שהפעלת f על x ו-100 מחזירה ביטוי מטיפוס T3 והפונקציה f מקבלת פרמטרים מטיפוסים T1,T2 לכן נוכל להסיק שהטיפוס של x הוא T1, הטיפוס של T2 הוא Num ומתקיים כי טיפוס ה-clouser כולו הוא [T1 → T3].

**שאלה 2**

**(a) ((lambda (x1) (+ x1 1)) 4)**

שלב 1: Rename bound variables

נבצע ריניימינג (למרות שבמקרה הספציפי הנ"ל אין צורך).

((lambda (**x**) (+ **x** 1)) 4)

שלב 2: Assign type variables for every sub expression

|  |  |
| --- | --- |
| **Variable** | **Expression** |
|  | ((lambda (x) (+ x 1)) 4) |
|  | (lambda (x) (+ x 1)) |
|  | (+ x 1) |
|  | + |
|  | x |
|  | 4 |
|  | 1 |

שלב 3: Construct type equations

|  |  |
| --- | --- |
| **Equations** | **Expression** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**שלב 4: Solve the equations**

**step 1**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 2**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 3**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 4**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 5**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 6**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 7**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 8**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 9**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**(b) ((lambda (f1 x1) (f1 x1 1)) 4 +)**

שלב 1: Rename bound variables

נבצע ריניימינג (למרות שבמקרה הספציפי הנ"ל אין צורך).

((lambda (f x) (f x 1)) 4 +)

שלב 2: Assign type variables for every sub expression

|  |  |
| --- | --- |
| **Expression** | **Variable** |
| ((lambda (f x) (f x 1)) 4 +) |  |
| (lambda (f x) (f x 1)) |  |
| (f x 1) |  |
| f |  |
| x |  |
| 1 |  |
| 4 |  |
| + |  |

שלב 3: Construct type equations

|  |  |
| --- | --- |
| **Equations** | **Expression** |
|  | ((lambda (f x) (f x 1)) 4 +) |
|  | (lambda (f x) (f x 1)) |
|  | (f x 1) |
|  | 1 |
|  | 4 |
|  | + |

שלב 4: Solve the equations

**step 1**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 2**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 3**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 4**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 5**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |

**step 6**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**step 7**

|  |  |
| --- | --- |
| **Substitution** | **Equation** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

נשים לב בשלב זה יש סתירה בין המשוואות:

ולכן הביטוי לא מורכב נכון מבחינת טיפוסים

**חלק 2**

שאלה 2.2 b

הפונקציה "העטופה" מחזירה ערך Promise<R> ולא R, מהסיבה שאנחנו מצפים מהערך המוחזר לקבל את ההפעלה של הפונקציה רק בזמן הקריאה.

כלומר לבדוק בזמן הקריאה האם קיים לנו ערך בstore , אם הוא קיים להשתמש בו, ואם הוא לא קיים להוסיף אותו לstore ולהחזיר את הערך המוחזר.