עבודת הגשה 1, חלק תיאורטי – PPL

מגישים: רועי אש, רון קורין

Q1.1 Type Intersection (6 points)

1.1.1. הType הנוצר כתוצאה מIntersection של ה-Types: T2,T1 במקרה זה הינו הT3 Type המוגדר בצורה הבאה: $T3 = \{a: number[\,], b: string\}$, כאשר ניתן לראות בתור דוגמה למשתנה מType זה את המשתנה הבא:

```
let T3 = {
    a: [0,1,2,3,4],
    b: "First 5 Natural Numbers"
};
```

1.1.2. הType הנוצר כתוצאה מIntersection של ה-Types: T2,T1 במקרה זה הינו הT3 Type המוגדר בצורה הבאה: $T3 = \{a: \{b: number, c: string\}\}$, כאשר ניתן לראות בתור דוגמה למשתנה מType זה את המשתנה הבא:

```
let T6 = {
    a: {
        b: 0,
        c: "Coca Cola - life is 0!"
    }
};
```

1.1.3. הType הנוצר כתוצאה מIntersection של ה-Types: T2,T1 במקרה זה הינו הT3 Type המוגדר בצורה הבאה: $T3 = empty\ set$ כאשר ניתן לראות בתור דוגמה למשתנה מType זה את המשתנה הבא: $let\ a: T1 = null$.

Q1.2 Type Inclusion (6 points)

1.2.1. נבחין כי במקרה זה מתקיים: $T1 < T2$ שכן למדנו בהרצאות שבהגדרת Type, מה שמצוין בהגדרה הוא חובה, אך על כך ניתן להוסיף דברים נוספים. כלומר, בעוד שבמשתנים מהטיפוס:T1 כל איבר במערך הוא map שמחויב להכיל שני פרמטרים: a שהוא מספר (number) ו-b שהוא map, במשתנים מהטיפוס: T2 כל איבר במערך רק מחויב להכיל את הפרמטר a שהוא מספר (number). אם כן, כל map שבתאי המערך של משתנה מטיפוס T1 יכול להופיע בתאי המערך של משתנה מטיפוס T2.

עבודת הגשה 1, חלק תיאורטי – PPL

מגישים: רועי אש, רון קורין

1.2.2. נבחין כי במקרה זה מתקיים: $T2 < T1$ שכן למדנו בהרצאות שפרמטר במפה מטיפוס any יכול לקבל כל ערך, ובפרט ערכים מהטיפוס number. מכאן שמפאת העובדה שכל משתנה מהטיפוס T2 הוא מפה שכוללת לפחות שני פרמטרים: a שהוא מסוג map שבתוכו פרמטר c מטיפוס number, ו-b מטיפוס number הוא בהכרח משתנה מטיפוס T1, שהוא מפה הכוללת לפחות שני שדות a,b ששניהם מטיפוס any.

1.2.3. נבחין כי במקרה זה מתקיים: $T1 < T2$ שכן כל משתנה מטיפוס T1 הוא מפה הכוללת בהכרח פרמטר a מטיפוס number, ופרמטר b שהוא undefined, והרי שכל משתנה מטיפוס T2 הוא גם כן מפה, הכוללת בהכרח פרמטר a מטיפוס number המתאים ל-a שב-T1 וכן פרמטר b מטיפוס any, שיכול לקבל גם undefined ולכן כל ערך של פרמטר b ב-T1 יכול להתקבל כערך b ב-T2. מכאן שעל כל משתנה מטיפוס T1 ניתן להסתכל גם כעל משתנה מטיפוס T2 כך שאכן $T1 < T2$.

Q1.3 Type Inference (8 points)

1.3.1. $T = \{name: string, age: number\}$

1.3.2. $T = \{\ children: (\{\ name: string\ \} | \{\ age: number\ \})[]\ \}$

1.3.3. $T = (x: number)\ \ =>\ \ number$

1.3.4. $T =< T, Y > (f: (x: T \mid Y)\ \ =>\ \ T \mid Y, l: (T \mid Y)[])$

Q1.4 Type Definitions (4 points)

שאלה ראשונה: האם אפשרי להגדיר Type ב-TypeScript עבור קבוצה של כל ה-Strings מאורך שגדול מ-2 באמצעות ה-Type Constructors שנלמדו בכיתה? התשובה לכך: לא. הבנאי בלבד לא יאפשר לנו לבדוק כי ערכים מסוימים שאינם תקינים (באורך 2 ומטה) לא ייוצרו בהמשך.

שאלה שניה: האם אפשרי להגדיר Type ב-TypeScript עבור קבוצה של כל המספרים הגדולים מ-0 באמצעות ה-Type Constructors שנלמדו בכיתה? התשובה לכך: לא. לא ניתן להציב תנאים מסוג זה כחלק מה-Type Constructors שנלמדו בכיתה, וכמו בשאלה הראשונה, לא נוכל לאכוף תנאי זה ולוודא שערכים מסוימים שאינם תקינים לא יוצרו בהמשך על סמך הבנאי בלבד.