

Question 1: Theoretical Questions

:Q1.a

True. לפי הסביבה, g מחזירה $T2$, ואז f מפעילה על ערך ההחזרה שלה, שהוא אכן מסוג $T2$ ומחזירה $T3$, לכן הטיפוס של כל הביטוי הזה הוא $T3$ כנדרש.

:Q1.b

False. לפי הסביבה, f היא פונקציה שאמורה לקבל משתנה אחד מסוג $T1$. אך f בביטוי הנתון מקבלת שני פרמטרים, ולכן נקבל false.

:Q1.c

True. לפי הסביבה, f היא פונקציה שמקבלת שני פרמטרים מסוג $T1$ ו $T2$, ומחזירה $T3$. בנוסף, l א לא נקבע טיפוס אז בהצבת x כפרמטר של λ , טיפוסו יקבע ל $T1$, נקבל ש f תפעיל את הפונקציה על x ו y שהם $T1$ ו $T2$ בהתאמה ותחזיר $T3$. כלומר הלמבדה לוקחת x שהוא $T1$, ומחזירה את תוצאת f שהיא $T3$ כנדרש.

:Q1.d

True. לפי הסביבה, f לוקחת משתנה מסוג $T2$ כלשהו ומחזירה $T1$. מכיוון ש $T2$ לא מוגדר פה, בהשמת x כפרמטר של f , נקבל שהטיפוס של x הוא גם $T1$ ו $T2$ כך ש $T1=T2$. אז ההשמה של הפרמטר של f חוקית, והיא מחזירה $T1$ כנדרש.

Question 2: L52 - Extending Types to Include Set Operations and Type Predicates

:Q2.1

- a. never
- b. string
- c. any
- d. number
- e. never
- f. boolean. הסבר:

`union boolean (diff string never)` זה `union boolean string`

אז החיתוך הוא בין `union boolean number` לבין `union boolean string`, שתוצאתו `boolean`.

:Q2.2

- a. `is? boolean`
- b. `is? boolean`
- c. `(isBoolean(z`

Q2.3

הפונקציה, לפי ה `type checker`, אמורה להחזיר `union number boolean (union string)`. הסבר: מבחינת ה `type checker`, בביטוי מסוג f הוא בודק האם ה `pred` הוא `boolean`, וערך ההחזרה הוא איחוד של `alt` ו `then`. לכן ה `type checker` ידרוש שערך ההחזרה יהיה `union string (union number boolean)`.

הערות חשובות:

- שימו לב שעשינו את 3.3 ו 3.4. הטסט אמנם לא עובר, אך זה כן מגיע לזה שהשם של ה var זה number (תראו את מה שהטסט האחרון ב-L5-typecheck.test.ts מחזיר, זה ממש קרוב לתשובה).
- לגבי שני הטסטים שלפינהם, הביטוי `(inter any number)` משום מה מתפרש כ never, ניסינו להסיר את כל ה any אך עדיין למרות אינספור ניסיונות זה לא עובד. אם מריצים `(inter number number)` לדוגמא זה כן עובד, אך גם כאן הבעיה היא באג קטן שאנחנו פשוט לא מוצאים!