04/04/18

Liav Rabi, id: 205688328
Benny Peresetsky, id: 310727771

PPL – Assignment 1

Q1.1 Type Intersection
1.1.1   Type T3 = T1&T2 → T3 = {a: number[], b:string}
      let example1 = {a: [1,2,3], b: 7} → is not of type T3.
      let example2 = {a: [1,2,3], b: "string"} → is of type T3.
1.1.2   Type T3 = T1&T2 → T3 = {a: {b: number, c: string}}
      let example3 = {a: {b: 87, c: 12}} → is not of type T3.
      let example4 = {a: {b: 87, c: "ABC" }}→ is of type T3.
1.1.3   Type T3 = T1&T2 → T3 = {a: undefined | null}
      let example5 = {a: 182} → is not of type T3.
      let example6 = {a: undefined, b:124}→ is of type T3.

Q1.2 Type Inclusion
1.2.1 T1<T2, T1 has another key "b" which is empty map that's why T2 is subset of T1.
1.2.2 T2<T1, T2 is a subset of T1 because all T1 keys are of type "any" and all T2 keys are "number" which is a specific type that's a subset type of "any". For example, T1 keys can be also a "String", "boolean" and etc.
1.2.3 T1<T2, first the key "a" in both types is similar. Also, "undefined" is included in the type "any" and therefore T1 is a subset of T2.

Q1.3 Type Inference
1.3.1 {name: string, age: number}
1.3.2 {children:[{name: string}, {age: number}]}
1.3.3 (x: any) $\Rightarrow$ any
1.3.4 ((x: T) $\Rightarrow$ T, T[]) $\Rightarrow$ T[]

Q1.4 Type Definitions
1.4.1 no, we can define that the variables would in type string but can't limit them to be of a specific length. Also, there are infinite number of strings which belong to this type and we can not apply operators of types like (x.length > 2), x is a String.
1.4.2 no, we can define a set of numbers but can't limit them to be greater than zero. Also, there are infinite number of numbers which belong to this type and we can not apply operators of types like (d > 0), d is a number.