

T1\*T2\*T3 🡪 T4.

E(x): T5 🡪 T6 so T5 🡪 Boolean

t(x, y): T7\*T8 🡪 T9 = T1

T7 = T2

T8 = T3

T1 = T2\*T3 🡪 T1

T1 = T4

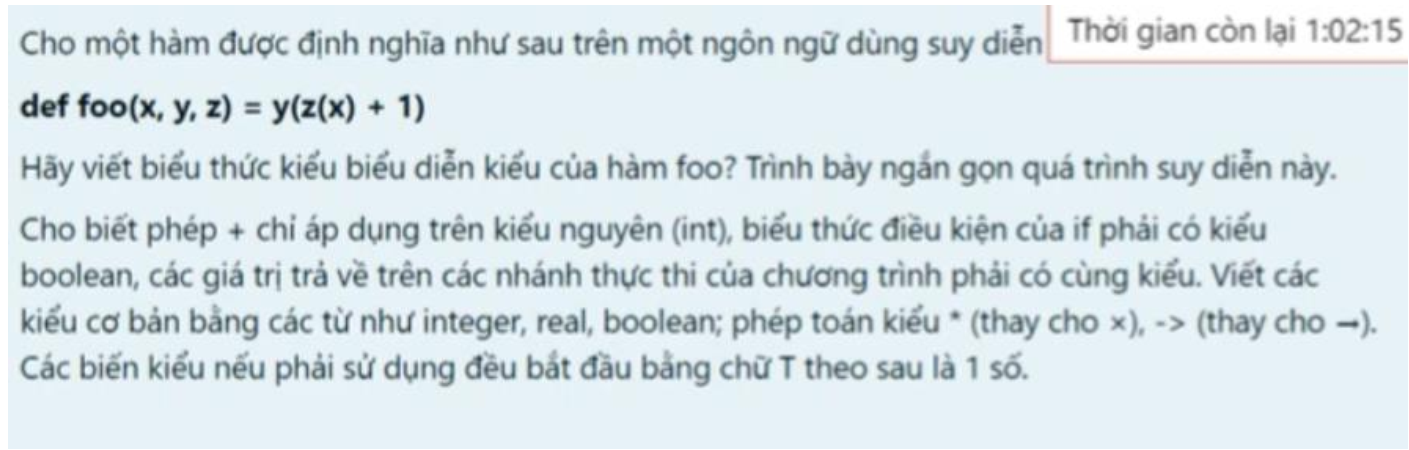
(T2\*T3 🡪 T1)\*T2\*T3 🡪 T1

F: (T2\*T3 🡪 Integer)\*T2\*T3 🡪 Integer

G(m, n) = n(m)

T1 🡪 T2

T3\*(T3 🡪 T4) 🡪 T4



Assume types of each function is as follow:

foo: T1 🡪 T2 (1), where T1 is the input type and T2 is the output type.

Function foo receives 3 parameters, therefore:

T1 = T3\*T4\*T5, where T3 is type of x, T4 is type of y, T5 is type of z.

Let function y, with the type T4 = T6 🡪 T7, where T6 is input type and T7 is output type

Let function z, with the type T5 = T8 🡪 T9, where T8 is input type and T9 is output type

Because foo returns y, so T2 = T4 = T7.

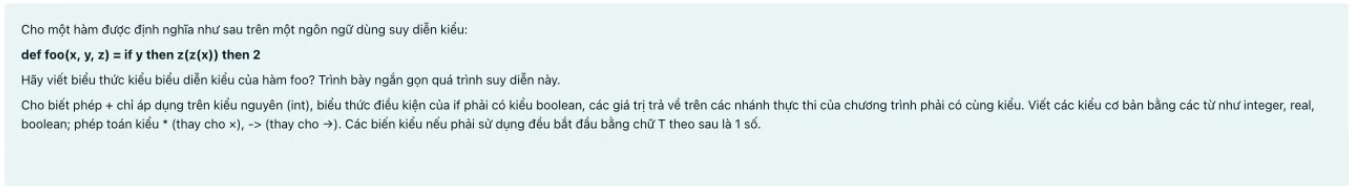
Because x has the type T3 so T8 = T3.

Because minus operator only works for integer, therefore z(x) has the type Integer 🡪 T5 = T6 = T9 = Integer.

Conclusively, x has the type T3, y has the type (Integer 🡪 T2), z has the type (T3 🡪 Integer).

🡪 T1 = T3\*(Integer 🡪 T2)\*(T3 🡪 Integer)

🡪 Type of foo is: T3\*(Integer 🡪 T2)\*(T3 🡪 Integer) 🡪 T2



Let the type of function foo is:

T1 🡪 T2, where T1 is input type and T2 is return type.

As foo receives 3 parameters:

T1 = T3\*T4\*T5, where T3 is type of x, T4 is type of y, T5 is type of z.

Let the type of z is:

T5 = T6 🡪 T7, where T6 is input type and T7 is output type.

Because y is the boolean expression, therefore type of y is T4 is Boolean type.

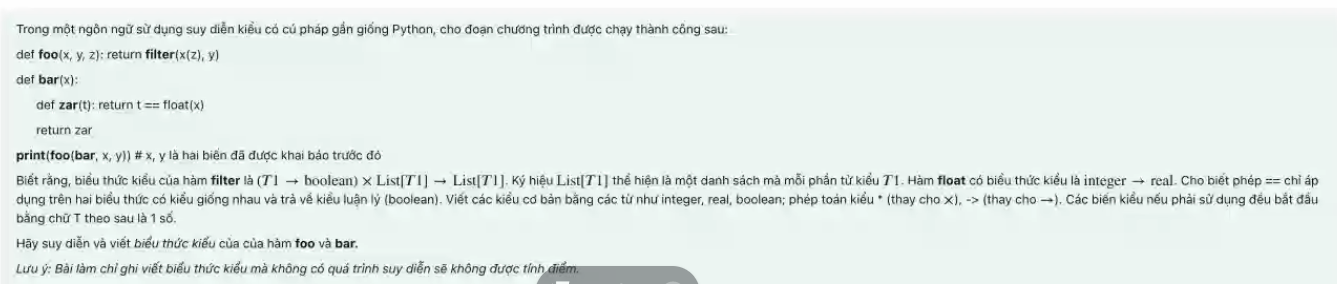
Because return types of all if-else branches are the same, so z(z(x)) must return the same type as 2 🡪 type of z(z(x)) is T7 is Integer.

With z(z(x)) should return the same as z(x), so T6 is also Integer.

As z(x) is Integer, therefore the input of z is also Integer 🡪 type of x is T3 is integer.

🡪 T1 = T3\*T4\*T5 = Integer\*Boolean\*(Integer 🡪 Integer)

🡪 Type of function foo is Integer\*Boolean\*(Integer 🡪 Integer) 🡪 Integer.



a. Type inference for zar:

Let type of zar is:

T1 🡪 T2, where T1 is input type and T2 is output type.

Float(x) receives Integer and return Real, therefore x has the type Integer.

Because ‘==’ operator only works on 2 same operand types and return Boolean type, therefore:

Type T1 has the same return type of float(x) is Real.

Type T2 is Boolean

🡪 Type of zar is: **Real 🡪 Boolean**

b. Type inference for bar:

Let type of bar is:

T1 🡪 T2, where T1 is input type and T2 is return type.

Return type of bar is return type of zar, and it receives x as input, where it was inferred to be Integer as above.

🡪 Type of bar is: **Integer 🡪 (Real 🡪 Boolean)**

c.



foo: T1 🡪 T2

x, y, z: T1 = T3\*T4\*T5

z or z – 2: T5 = Integer

y(x): T4 = T6 🡪 T7

y(x) must be same type as x: T3 = T4

y(y(x)) works as a boolean expression 🡪 T7 = T3 = T4 = Boolean

🡪 T1 = Boolean\*(Boolean 🡪 Boolean)\*Integer 🡪 Integer