**PPL – Assignment4 – Part1**

**By Yair Derry & Gal Noy**

(a) The typing statement (f (g a)) : T2 is **false** because function f expects an input of type T1, not T2 as provided by function g. Therefore, the resulting type of the expression cannot be T2.

(b) The typing statement (lambda (x) (f x 100)) : [T2 → T3] is **false** because the type T2 is not necessarily restricted to a number. The expression (f x 100) implies that the second argument must be of type T2, but it does not specify that T2 is specifically a numeric type.

(c) The typing statement ((lambda (x) (f x))) : [T1 → T2] is **true** because the lambda function takes an argument x, and when applied to an input of type T1, it returns a value of type T2. The type of the bound variable x can be inferred through type inference, and it does not require any assumptions in the type environment on the left side.

(d) The typing statement (lambda (x) (f x y)) : [T1 → T3] is **true** because the lambda function accepts an argument x of type T1 and returns a value of type T3, assuming the left-hand side typing context. The additional variable y with type T2 does not affect the overall type of the lambda function.

(a) The typing statement (f (g a)) : T2 is **false** because function f expects an input of type T1, not T2 as provided by function g. Therefore, the resulting type of the expression cannot be T2.

(b) The typing statement (lambda (x) (f x 100)) : [T2 → T3] is **false** because the type T2 is not necessarily restricted to a number. The expression (f x 100) implies that the second argument must be of type T2, but it does not specify that T2 is specifically a numeric type.