

// For each unquoted identifier (q0, not q0'), state if it type checks in F#
 // If it does, state its type, and whether the compiler gives a warning on its definition.
 //

module tbi1q5

let q0 = fun (a,b) -> (a,a) $a \rightarrow b \rightarrow a \rightarrow a$

let q0' = q0 ("aa", 1) $string \rightarrow int$

let q1 x y = x, y $'a \rightarrow 'b \rightarrow 'a \times 'b$

let q2 x y z = (x,y),(x,z) $a \rightarrow b \rightarrow c \rightarrow (a \times b) \times (a \times c)$

let q3 x = [x] $a \rightarrow a \text{ list}$

let q3' x = ^{q3} "hello", ^{q3} 22 $string \times int \text{ list}$

let q4 f = [f 1], [f 2] $int \rightarrow int$
 compiler error ("a" should be int)
 $int \rightarrow (int \rightarrow a) \rightarrow a \text{ list} \times a \text{ list}$

let q5 x y z = [x; y; z] $a \rightarrow a \rightarrow a \rightarrow a \text{ list}$

let q6: (int -> int) = fun x -> x + 1 $int \rightarrow int$

let q7 = [] ~~int~~ $a \text{ list}$

let q7' = q7 @ q7 $a \text{ list} \rightarrow a \text{ list}$

let q8 = [] $a \text{ list}$

let q8' = q8 @ [1] $int \text{ list}$

let q9 = [[[[[[]]]]]] $a \text{ list list list list}$

let rec q10 a b = function ~~int~~ $int \rightarrow int \rightarrow int \rightarrow int$

| 0 -> 1

| n when n > 0 -> q10 b (a + b + 1) (n-1)

| n when n < 0 -> failwithf "Invalid"

q10 a b match

let rec q11 f = f (q11 f) ~~(a -> b)~~ $(a \rightarrow a) \rightarrow a$

let rec q12 f = f f ~~(a -> b)~~ $(a \rightarrow b) \rightarrow b$ ~~logical type~~

let q13 = Ok (Ok ()) $Result (Result (unit, a), b)$

let q13' = match q13 with | Ok x -> failwith "Ok" | Error x -> x

let q13'' = id q13