Olivia Pannell

CIS 425

Homework 5

Part 1: Type Checking

1.
    a. INT
    b. ARROW(INT, BOOL)
    c. ARROW(ARROW(VAR “a”, VAR “b”), ARROW(VAR “a”, VAR “b”))
2.
    a. AST_APP(AST_SUCC, (AST_APP(AST_PRED, AST_NUM 5)))
    b. AST_IF(AST_NUM 7, AST_BOOL true, AST_NUM 5)
    c. AST_FUN(“a”, AST_APP(AST_APP(AST_ID “f”, AST_ID “a”), AST_ID “a”))

Part 2: Type Inference

1. fun f (g,h) = g (h 0)
    a. f: 'a1 * 'a2 -> 'a3

        g: 'a1 : 'a4 -> 'a3

        h: 'a2: int -> 'a4

        f: ('a -> 'b) * (int -> 'a) -> 'b
2. fun apply(f, x) = f x
    a. apply: 'a1 * 'a2 -> 'a1
       f: 'a1: 'a2 -> 'a1
       apply: ('a -> 'b) * 'a -> 'b
3. fun reverse nil = nil | reverse (x::xs) = reverse xs
    a. reverse: 'a1 list -> 'a2 list
       nil: 'a3 list -> 'a4
       'a1 list = 'a3 list
       'a2 list = 'a4
       ‘a list -> ‘b list
4. fun f(g,h) = g h + 2
    a. f: 'a1 * a2 -> int
       g: 'a1: 'a2 -> int
       f: ('a -> int) * 'a -> int
5. This is an error because g acts both as term type and a function type. So g: ‘a1: ((a2 -> int) ->int) -> int would continuously loop forever.

6. fun ff f x y = if (f x y) then (f 3 y) else (f x "zero")
    a. ff: 'a1 -> a2 -> a3 -> a4
       f: a1: a2 -> a3 -> bool
       f: a1: int -> a3 -> a4
       f: a1: a2 -> string -> a4
       ff: (int -> string -> bool) -> int -> string -> bool
7. fun gg f x y = if (f x y) then (f 3 y) else (f y "zero")
    a. gg: ‘a1 -> ‘a2 -> ‘a3 -> ‘a4
       f: ‘a1:’ a2 -> ‘a3 -> bool
       f: ‘a1: int -> ‘a3 -> ‘a4
       f: ‘a1: ‘a3 -> string -> ‘a4
       This would be an error because the first constraint is already an int and a ‘a2 and a3 is a string, which do not match because an int is not a string.
8. fun hh f x y = if (f x y) then (f x y) else (f x "zero")
    a. ff: 'a1 -> a2 -> a3 -> a4
       f: a1: a2 -> a3 -> bool
       f: a1: a2 -> a3 -> a4
       f: a1: a2 -> string -> a4
       ff: ('a -> string -> bool) -> 'a -> string -> bool