CS 558 HW2

Soumya Banerjee

October 12, 2010

1 HW 2.3 (Graphs)

1.1 1. Are all directed graphs representable in this fashion?

No, all directed graphs cannot be represented in this fashion. The edges are represented in the form [(Int,Int)] and one would run out of numbers to enumerate the vertices if the number of vertices exceeded $2^{31} - 1$.

2 HW 2.6 (Programs)

foldr f e xs = foldl (flip f) e (reverse xs)

axioms:

foldr f v [] = v (Eq. 1)

foldr f v (x:xs) = f x (foldr f v xs) (Eq. 2)

foldl f v [] = v (Eq. 3)

foldl f v (x:xs) = foldl f (f v x) xs (Eq. 4)

flip f x y = f y x (Eq. 5)

reverse [] = [] (Eq. 6)

reverse (x:xs) = (reverse xs) ++ [x] (Eq. 7)

[] ++ ys = ys (Eq. 8)

(x:xs) ++ ys = x:(xs ++ ys) (Eq. 9)

Proof by calculation

Let xs = [x_0, x_1, x_2, ..., x_n] (Eq. 10)

LHS = foldr f e xs

= { by Eq. 10 }

foldr f e [x_0, x_1, x_2, ..., x_n]

= { by Eq. 2 definition of foldr }

f x_0 (foldr f v [x_1, x_2, ..., x_n])

= { by Eq. 2 definition of foldr }

f x_0 (f x_1 (foldr f v [x_2, ..., x_n]))

= { by Eq. 2 definition of foldr }

f x_0 (f x_1 (f x_2 (foldr f v [x_3, ..., x_n])))

= { after applying Eq. 2 $n-2$ times}

f x_0 (f x_1 (f x_2 (.... (f x_n (foldr f v []))...)))

= { by Eq. 1 base case definition of foldr}

f x_0 (f x_1 (f x_2 (.... (f x_n e)...)))

RHS = foldl (flip f) e (reverse xs)

= { by Eq. 10 }

foldl (flip f) e (reverse [x_0, x_1, x_2, ..., x_n])

= { by Eq. 7 definition of reverse }

foldl (flip f) e ((reverse [x_1, x_2, ..., x_n]) ++ [x_0])

= { by Eq. 7 definition of reverse }

foldl (flip f) e (((reverse [x_2, ..., x_n]) ++ [x_1]) ++ [x_0])

= { by Eq. 7 definition of reverse }

foldl (flip f) e ((((reverse [x_3, ..., x_n]) ++ [x_2]) ++ [x_1]) ++ [x_0])

= { after applying Eq. 7 $n-2$ times }

foldl (flip f) e ((((...... ((reverse [ ]) ++ [x_n]) ++ ....) ++ [x_2]) ++ [x_1]) ++ [x_0])

= { after applying Eq. 6 base case of reverse }

foldl (flip f) e ((((...... ([ ] ++ [x_n]) ++ ....) ++ [x_2]) ++ [x_1]) ++ [x_0])

= { after applying Eq. 8 base case of ++ }

foldl (flip f) e ((((...... ([x_n] ++ [x_{n-1}]) ....) ++ [x_2]) ++ [x_1]) ++ [x_0])

= { after applying Eq. 9 definition of ++ }

foldl (flip f) e ((((...... ([x_n, x_{n-1}]) ....) ++ [x_2]) ++ [x_1]) ++ [x_0])

= { after applying Eq. 9 $n-1$ times }

foldl (flip f) e ([x_n, x_{n-1}, ..., x_2, x_1, x_0])

= { after applying Eq. 4 definition of foldl }

foldl (flip f) (flip f e x_n) ([x_{n-1}, ..., x_2, x_1, x_0])

= { after applying Eq. 5 definition of flip }

foldl (flip f) (f x_n e) ([x_{n-1}, ..., x_2, x_1, x_0])

= { after applying Eq. 4 definition of foldl }

foldl (flip f) (flip f (f x_n e) x_{n-1}) ([x_{n-2}, ..., x_2, x_1, x_0])

= { after applying Eq. 5 definition of flip } foldl (flip f) (f x_{n-1} (f x_n e)) ([x_{n-2}, ..., x_2, x_1, x_0])

= { after applying Eq. 5 definition of flip and Eq. 4 definition of foldl $n-4$ times }

foldl (flip f) (f x_3 (.... (f x_{n-1} (f x_n e)) ....)) ([x_2, x_1, x_0])

= { after applying Eq. 4 definition of foldl }

foldl (flip f) (flip f (f x_3 (.... (f x_{n-1} (f x_n e)) ....)) x_2) ([x_1, x_0])

= { after applying Eq. 5 definition of flip }

foldl (flip f) (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....))) ([x_1, x_0])

= { after applying Eq. 4 definition of foldl }

foldl (flip f) (flip f (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....))) x_1) ([x_1, x_0])

= { after applying Eq. 5 definition of flip }

foldl (flip f) (f x_1 (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....)))) ([x_0])

= { after applying Eq. 4 definition of foldl }

foldl (flip f) (flip f (f x_1 (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....)))) x_0) ([])

= { after applying Eq. 5 definition of flip }

foldl (flip f) (f x_0 (f x_1 (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....))))) ([])

= { after applying Eq. 3 base case of foldl }

f x_0 (f x_1 (f x_2 (f x_3 (.... (f x_{n-1} (f x_n e)) ....))))

Hence LHS = RHS by the calculational proof method and foldr f e xs = foldl (flip f) e (reverse xs).