CS 557 Homework 1 Problem 1.2

Soumya Banerjee

october 6th, 2008

Contents

1 Introduction

module Main where main = if (elem' 12 [12,23]) then putStr "Yes" else putStr "no"

2 Elem function (user written)

```
elem' :: (Eq a) =¿ a -¿ [a] -¿ Bool
    elem' m [] = False
    elem' m (n:ns) = if m == n then True else elem' m ns
    ...
```

3 Set Equal

```
setEqual :: [Integer] -¿ [Integer] -¿ Bool
    setEqual [] [] = True
    setEqual [] x = False
    setEqual x [] = False
```

```
    setEqual (m:ms) (n:ns) = (m == n) && (setEqual ms ns)
    ...
```

4 Set Difference

```
setDiff [] x = []
    setDiff x [] = x
    setDiff (m:ms) (n:ns) = if elem' m (n:ns) then setDiff ms (n:ns) else m:(setDiff ms (n:ns))
```

5 Set Intersection

```
setIntersection :: [Integer] -¿ [Integer] -¿ [Integer]
    setIntersection [] x = []
    setIntersection (m:ms) (n:ns) = if elem' m (n:ns) then m:(setIntersection ms (n:ns)) else setIntersection ms (n:ns)
```

6 Set Union

```
setUnion :: [Integer] -¿ [Integer] -¿ [Integer]
    setUnion xs ys = (setDiff xs ys) ++ ys
```

7 Power Set

```
compl :: Integer -¿ [[Integer]] -¿ [[Integer]]
    compl y yss = [[y] ++ ys — ys ¡- yss]
    powerSet :: [Integer] -¿ [[Integer]]
    powerSet [] = [[]]
    powerSet (x:xs) = (powerSet xs) ++ (compl x (powerSet xs))
```