

Изпит по Функционално програмиране

спец. Информатика, 8.02.2016 г.

Вариант А

Задача 1. (10 т.) Да се напише функция `findColumns`, която по дадена матрица от числа намира броя на колоните, за които е вярно, че всичките им елементи се срещат в някой от редовете на матрицата.

Пример: `findColumns [[1,4,3],[4,5,6],[7,4,9]]` → 1

Задача 2. (15 т.) Ако **f** и **g** са унарни числови функции, а **h** е бинарна числова функция, то дефинираме функцията **f{h}g** по следния начин: **(f{h}g) (x) := h(f(x),g(x))**. Дадени са две цели числа **a < b** и два списъка **uns** и **bins**, съответно с унарни и бинарни функции. Да се напише функция `check`, която проверява дали има такива функции **f** и **g** от списъка **uns** и **h** от списъка **bins** така, че **f{h}g** да съвпада с някоя от функциите в **uns** върху всички цели числа в интервала **[a;b]**.

Упътване: използвайте помощна функция от по-висок ред `combine`, която по дадени **f**, **g** и **h** връща така дефинираната **f{h}g**.

Пример: `check 1 9 [(+1),(-1),(-1).(^2)] [(*)]` → True

Задача 3. (15 т.) Растение се описва с наредена тройка от име (низ), минимална и максимална температура, в която вирее (цели числа градуси). Да се напише функция `garden`, която по списък от растения намира интервал от стойности на температура, в който максимален брой растения могат да виреят, заедно с имената на тези растения.

Пример:

`garden[(“peas”, 5, 25), (“beans”, 3, 15), (“cocoa”, 20, 30)]` → `((20,25),[“peas”,“cocoa”])`

Задача 4. (15 т.) Даден е граф **g**, представен със списък от наследници и връх **u** в графа. Да се напише функция `maxPath`, която намира най-дългия ацикличен път в **g**, започващ от върха **u**.

Пример: `maxPath [[1,2,4],[2,3],[3,2],[4]] 1` → `[1,2,3]`

Изпит по Функционално програмиране

спец. Информатика, 8.02.2016 г.

Вариант Б

Задача 1. (10 т.) Да се напише функцията `hasColumn`, която по дадена матрица от числа проверява има ли в нея колона, в която всяко число се среща във всеки един от редовете на матрицата.

Пример: `hasColumn [[1,2,3],[2,3,4],[5,3,2]]` → `True`

Задача 2. (15 т.) Ако **f** и **g** са бинарни числови функции, а **h** е унарна числова функция, то дефинираме функцията **f—g→h** по следния начин: **(f—g→h) (x) := f(x,g(x,h(x)))**. Дадени са две цели числа **a < b** и два списъка **uns** и **bins**, съответно с унарни и бинарни функции. Да се напише функцията `check`, която проверява дали има такива функции **f** и **g** от списъка **bins** и **h** от списъка **uns** така, че **f—g→h** да съвпада с някоя от функциите в **uns** върху всички цели числа в интервала **[a;b]**.

Упътване: използвайте помощна функция от по-висок ред `combine`, която по дадени **f**, **g** и **h** връща така дефинираната **f—g→h**.

Пример: `check 1 9 [(+2),(-2)] [(+),(-)]` → `True`

Задача 3. (15 т.) Спектакъл се описва с наредена тройка от име (низ), начален кръгъл час (от 0 до 23) и брой минути, които продължава (от 1 до 300). Да се напише функцията `showtime` която по списък от спектакли в един и същ ден намира в кой едночасов интервал протичат максимален брой спектакли, колко минути продължава засичането и кои са имената засичащите се спектакли.

Пример: `showtime[("Cats",21,130),("Rent",19,100),("Hair",22,90)]` → `((22,70),["Cats","Hair"])`

Задача 4. (15 т.) Даден е граф **g**, представен със списък от наследници. Да се напише функцията `maxCycle` намери най-дългият прост цикъл в **g**, включващ даден връх **u**. Упътване: прост цикъл наричаме път в който само първият и последният връх съвпадат.

Пример: `maxCycle [[1,2],[2,3],[3,1,4],[4,2]] 1` → `[1,2,3,1]`