Programátorská dokumentace

Moduly

Použil jsem modul `Data.List`, abych nemusel vytvářet vlastní triviální funkce jako např. `intersect`,`union`,`subseqences`.

Typy

Nový typ v této implementaci je pouze jeden

```
type KakuroCell = (Int,Int,Int)
```

kde

1. Int má hodnoty -2 až 9, kde -2 určuje prázdný blok, -1 určuje pomocný blok tzn. určuje sumu v řádku a sloupci, 0 je pro nevyplněnou buňku a 1 až 9 pro vyplněnou buňku
2. Int má hodnoty 0 až 45 a určuje sumu pro řádek, 0 znamená, že neví vyplněno
3. Int je to samé jako 2. Int pouze pro sumu sloupce

Pokud tedy chceme zapsat nevyplněnou buňku bude to vypadat následovně

```
let nevyplnenaBunka = (0,0,0)
```

Funkce

```
kakuro::[KakuroCell]->IO()
```

Pokud dostane vyplněné kakuro, pak ho zkontroluje a vypíše zda je vyplněno správně nebo špatně. Pokud dostane nevyplněné kakuro, pak ho vyhodnotí a vrátí výsledek.

```
solve::[KakuroCell]->[KakuroCell]
```

Zavolá funkci `calculationIntersect` na vstup a zkontroluje výsledek

1. Pokud je výsledek hotové řešení vrátí ho
2. Pokud je výsledek stejný jako vstup, pak zavolá

   `calculationRandom`

3. Jinak zavolá sám sebe na výsledek

Tato funkce obstarává cyklus, kde v každé iteraci dojde k dalšímu vyplnění kakura nebo vrátí výýsledek

```
calculationIntersect'::[KakuroCell]->[Int]->[KakuroCell]
```

Tato funkce je volaná z pomocné funkce `calculationIntersect::[KakuroCell]->[KakuroCell]`, která pouze ke vstupu přidá seznam indexů buněk, které nejsou vyplněny.
`calculationIntersect'` si zjistí kombinace možné na řádku a v sloupci a udělá průnik. Pokud je průnik jednoznačný (tzn. obsahuje pouze jedno číslo), pak ho dosadím. A toto zopakuji pro každou nevyplněnou buňku.

`calculationRandom'::[KakuroCell]->[KakuroCell]`

Tato funkce je volána z pomocné funkce `calculationRandom::[KakuroCell]->[KakuroCell]`, která pouze ke vstupu přidá pozici první nevyplněné buňky a pole možných průniků kombinací na řádku a v sloupci.
`calculationRandom'` vytvoří nové kakuro s první možnou hodnotou na pozici a spustí `solve`.

1. Pokud je to hotové a validní řešení, pak vrátím výsledek
2. Jinak zavolá sama sebe na zbylé možné hodnoty na pozici

Tato funkce musí být implementováma, protože ne všechny situace jsou vyřešit pomocí průniků.

Další funkce jsou triviální a jejich popis se nachází jako komentář v kódu