

Zadání online půlsestrálního testu

Jedná se o programy v Haskellu, příklady mohou sloužit k procvičení.

a)

V jazyku Haskell nadefinujte datový typ pro reprezentaci aritmetických výrazů s operacemi násobení a sčítání nad proměnnými – typ proměnné je dopředu neznámý. Definujte funkci `tr`, která má jako parametr výraz definovaný dle vašeho typu. Na základě vlastnosti asociativity a komutativity operace sčítání pro reálná čísla upraví datovou reprezentaci tak, aby při vyhodnocení post-order průchodem bylo sčítání jednotlivých operandů prováděno v maximální míře zleva doprava. Pokud pro pořadí použijeme v linearizovaném zápisu závorky, tak by např. takovýto výraz $(a + b) + (c + d)$ byl transformován na $((a + b) + c) + d$, nebo obdobný až na pořadí proměnných, nicméně význam výrazu se změnit nesmí. Volně využijte obsah Prelude.

b)

V jazyku Haskell nadefinujte datový typ pro reprezentaci aritmetických výrazů s operacemi násobení a sčítání nad celými čísly a proměnnými – typ proměnné je dopředu neznámý. Definujte funkci `eo`, která využije vlastnosti čísla jedna vzhledem k operaci násobení a co nejvíce zjednoduší výraz, který je dán jako parametr a ten zjednodušený vrátí jako výsledek. Dbejte na to "co nejvíce". Volně využijte obsah Prelude.

c)

V jazyku Haskell nadefinujte datový typ pro reprezentaci booleovských výrazů s operacemi logického součtu, součinu i negace nad booleovskými hodnotami a proměnnými – typ proměnné je dopředu neznámý. Definujte funkci `dm`, která má dva parametry, tabulku symbolů a booleovský výraz dle vašeho typu. Tabulka symbolů je seznam dvojic proměnná, hodnota. Funkce `dm` odstraní dvojí negaci, aplikuje de Morganova pravidla a pro proměnné obsažené v tabulce dosadí jejich hodnoty. Vše s maximální účinností. Volně využijte obsah Prelude.

d)

V jazyku Haskell nadefinujte datový typ pro reprezentaci aritmetických výrazů s operacemi násobení a sčítání nad celými čísly a proměnnými – typ proměnné je dopředu neznámý. Definujte funkci `ez`, která využije vlastnosti čísla nula vzhledem k operaci násobení a co nejvíce zjednoduší výraz, který je dán jako parametr a ten zjednodušený vrátí jako výsledek. Dbejte na to "co nejvíce". Volně využijte obsah Prelude.

e)

V jazyku Haskell definujte typ pro reprezentaci vyhledávacího binárního stromu nad klíčem a hodnotou dopředu neznámého typu. Definujte funkci `put`, která pro zadaný klíč a hodnotu provede vložení hodnoty a úpravu daného stromu dle vašeho typu. Pokud je více hodnot spojených se stejným klíčem, tak je ukládá do seznamu k danému klíči, ale neduplikuje je, tedy každá hodnota je v seznamu jen 1x!

f)

V jazyku Haskell definujte typ pro reprezentaci vyhledávacího binárního stromu nad klíčem a hodnotou dopředu neznámého typu. Definujte funkci `ins`, která pro zadaný klíč a hodnotu provede vložení hodnoty a úpravu daného stromu dle vašeho typu. Situaci, že pro daný klíč již hodnota existuje, vhodně ošetřete, úprava stromu se provádět nebude, vrátí se původní strom a informace o tom, že vložení selhalo. Volně využijte obsah Prelude.