Pamphlet 5, INF222, Spring 2021

5.1 Calculator with more operations

We now have a fairly advanced calculator with variable declarations and assignment. But the set of operations are limited: addition, multiplication and subtraction/negation.

Here we add two more operations: integer division `Idiv` and remainder `Rem` to the AST. With normal semantics, both these operations will crash if their second argument is 0.

```
-- | AST for variable based integer calculator with integer division and remainder.
--
-- Author Magne Haveraaen
-- Since 2020-03-23

module CalculatorVariableIdivremAST where


--------------------


-- | Expressions for a calculator with variables.
-- The calculator supports literals and operations
-- Addition, multiplication, subtraction/negation,
-- integer division and remainder.
data CalcExprAST
  = Lit Integer
  | Add CalcExprAST CalcExprAST
  | Mult CalcExprAST CalcExprAST
  | Sub CalcExprAST CalcExprAST
  | Neg CalcExprAST
  | Idiv CalcExprAST CalcExprAST
  | Rem CalcExprAST CalcExprAST
  | Var String
  deriving (Eq, Read, Show)

-- | Statement for setting and changing a variable
data CalcStmtAST
  = SetVar String CalcExprAST
  | AssVar String CalcExprAST
  deriving (Eq, Read, Show)


--------------------


-- | A few ASTs for variable based CalcExprAST.
calculatorVariableAST1
  = Lit 4
calculatorVariableAST2
  = Neg (Mult (Add (Lit 3) (Sub (Lit 7) (Lit 13))) (Lit 19))
calculatorVariableAST3
  = Add (Var "Reg1") (Var "Reg4")
calculatorVariableAST4
  = Var "Reg2"
```

```
-- | A few CalcStmtASTs for setting and assigning variables.
calculatorSetVariableAST1
  = SetVar "Reg4" calculatorVariableAST1
calculatorSetVariableAST2
  = SetVar "Reg1" calculatorVariableAST2
calculatorSetVariableAST3
  = AssVar "Reg2" calculatorVariableAST3
calculatorSetVariableAST4
  = AssVar "Reg1" calculatorVariableAST4

--------------------
```

5.2 Task

The tasks are to:

- Extend the variable based calculator with the two new operations.
- Upgrade the unit tests to check the new operations (add some examples that uses `Idiv` and `Rem`). You may want to add an example that makes it easy to check for any pair of integers x, y, where $y \neq 0$, that $x = y * (x \operatorname{idiv} y) + x \operatorname{rem} y$.
- Upgrade the list of examples for the interactive calculator to make testing this property easy.