Programming languages – Haskell

Homework exercise (2024/25)

T. Goluch

1st exercise (1 pts.)

B-smooth numbers[1] are those whose prime divisors are less than or equal to B. Write a program calculating for a given B and n, how many B-smooth numbers are there that do not exceed n? For example, for B = 5 and n = 30, the initial 5-smooth numbers, also known as Hamming numbers, are: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30… so the answer is: 18.

2nd exercise (6 pts.)

Write a program to solve propositional calculus[2] with: negation ¬ (N), conjunction ∧ (C), alternative ∨ (A) and implication → (I) functors.
Sentence type: data Sentence = S Char | … - i.e. the variable can be any variable S
Example sentence: `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`

Program should:
1) contains a function: `print sentence` – (1 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
the output should be `"(¬p => ((p ∧ q) ∨ r))"`
2) contains a function: `write_variables sentence` – (1 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
the output should be `[p, q, r]` (not necessarily sorted, but unique)
3) contains a function: `check sentence values_map` – (1 pts.)
for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
and values map `fromList [('p', False), ('q', True), ('r', False)]`
the output should be `False`
4) contains a function: `isTautology` – (1 pts.)
checking whether the sentence is true for any valuation of variables in the formula. Check if the following sentences are tautologies?:
a) (¬p∧p)→q)
b) ((p→q)→p)→p
c) (p→(p→q))→(p→q)

Extend the program to support Łukasiewicz's three-valued logic[3] `True`/`False`/`Nothing.` Program should:

[1] https://en.wikipedia.org/wiki/Smooth_number
[2] https://en.wikipedia.org/wiki/Propositional_calculus
[3] https://en.wikipedia.org/wiki/Three-valued_logic

5) contains a function: `check3val sentence values_map` – (1 pts.)

for a given example `(I (N (S 'p')) (A (C (S 'p') (S 'q')) (S 'r')))`
and values map `fromList [('p', LFalse), ('q', LTrue), ('r', LNothing)]`
the output should be `LFalse`

6) contains a function: `isTautology3val` – (1 pts.)

Check if the following sentences are tautologies?:
a) p→(q→q)
b) (p→q)→((p→r)→(p→(q∧r)))
c) (p→r)→((q→r)→((p∧q) →r))

Comments: all functions should have an appropriate header with the type of function,