

# 1 Общие задания

1. Даны формулы:

(a)  $a \wedge b \vee \neg a \wedge \neg b \rightarrow (a \rightarrow b) \wedge (b \rightarrow a)$ ; (0.2 б.)

(b)  $((a \rightarrow b) \rightarrow a) \rightarrow a$ ; (0.2 б.)

(c)  $a \vee b \rightarrow (b \rightarrow c) \rightarrow (a \rightarrow d) \rightarrow (d \rightarrow c) \rightarrow c$ ; (0.2 б.)

(d)  $a \vee b \rightarrow (b \rightarrow c) \rightarrow (a \rightarrow d) \rightarrow (d \rightarrow c) \rightarrow d$ ; (0.2 б.)

(e)  $(a \wedge \neg b \vee \neg a \wedge b) \wedge a \wedge b$ . (0.2 б.)

Для каждой из формул расставьте все скобки, которые тут опущены, и укажите, является она общезначимой, выполнимой или невыполнимой.

2. Уберите все лишние скобки из формулы  $((\neg(a \rightarrow (\neg(b \wedge d)))) \wedge (d \wedge (b \vee c)))$ . (0.5 б.)

3. Напишите функцию, которая принимает две формулы логики высказываний и проверяет их на эквивалентность. (2.5 б.)

```
{-# LANGUAGE DeriveFunctor #-}
module Logic2021Khalansky01 where
import qualified Data.Map as M

infixr 3 :->
infixl 4 :|
infixl 5 :&

data Preterm a = Var a
               | Preterm a :& Preterm a
               | Preterm a :| Preterm a
               | Preterm a :-> Preterm a
               | Not (Preterm a)
               deriving (Show, Eq, Functor)

{- |Вернуть либо оценку данной формулы на данных значениях переменных, либо,
если переменных на входе не хватает, сообщить хотя бы одну переменную, которой
нет.

> tm = Var "a" :& Var "b" :| Var "c"
> evalTerm tm (M.fromList [("a", True), ("b", False), ("c", True)])
Right True
> evalTerm tm (M.fromList [("a", True), ("b", False)])
Left "c"
-}
evalTerm :: Ord a => Preterm a -> M.Map a Bool -> Either a Bool
evalTerm = undefined

type Term = Preterm Int

propsEquiv :: Term -> Term -> Bool
propsEquiv = undefined
```

Согласно теореме про условия эквивалентности, есть два возможных варианта, как решать это задание: либо убедиться, что эквивалентность этих формул является тавтологией, либо проверить, что на всех входах данные формулы имеют равную оценку.