1. Оценка позволяет трактовать формулу логики высказываний как функцию $\mathbb{B}^n \to \mathbb{B}$, где n — число различных переменных. i'ая компонента аргумента функции кодирует истинность высказывания p_i.

   Хотим ввести альтернативную интерпретацию формулы как функции $\mathbb{Q}^n \to \mathbb{Q}$ ($\mathbb{Q}$ — рациональные числа). Пусть, если в формуле n различных переменных, происходит симуляция n независимых случайных событий, причём вероятность успеха события i — это p_i.

   Тогда вся формула должна описывать некоторое составное событие. Например, если мы подкинем монету и бросим кубик, то событие "выпал орёл и единица или выпала решка и что-то кроме единицы" можно записать как $p \wedge q \vee (\neg p \wedge \neg q)$, где $p = 0.5$ — вероятность выпадения орла, $q = 1/6$ — вероятность выпадения единицы.

   $\wedge$ означает, что произошли оба события. $\vee$ — что хотя бы одно. $\neg$ — что событие не произошло. $x \to y$ выразим через $\neg x \vee y$.

   - Правда ли, что такая оценка формулы P равна оценке формулы P', полученной из P по алгоритму построения ДНФ? Почему?
   - Опишите, как можно осуществить такую интерпретацию. Можно словами, можно в виде кода на любом языке программирования.

2. На лекции было доказано, что критерий Поста задаёт достаточные условия: имея набор связок, который ему удовлетворяет, можно построить формулу, соответствующую любой логической функции.

   Докажите обратное: для каждого предкласса приведите такой пример логической функции, которая не может быть выражена ни в какой системе связок, все связки которой относятся к этому предклассу.

   Например, как было упомянуто на лекции, если все функции сохраняют единицу, то нельзя построить формулу $\neg a$.

3. Приведите к СДНФ и СКНФ:

   (a) $(a \to b \vee c \to b) \wedge (\neg c \vee \neg b)$

   (b) $a \oplus b \oplus (a \leftrightarrow b)$

   (c) $(a \uparrow b) \uparrow (\neg a \downarrow b)$

   (d) $a \leftrightarrow (b \leftrightarrow c)$

4. Постройте полином Жегалкина для формулы $(a \oplus b) \to (\neg c \oplus (a \vee d \wedge \neg a))$.

5. Проверьте полноту каждой системы связок и докажите своё суждение.

   (a) $(\oplus, \vee, 1)$

   (b) $(\to, 0)$

   (c) $(\to, 1)$

   (d) $(\oplus, \wedge)$

6. Напишите программу, которая принимала бы формулу в логике высказываний возвращала эквивалентную ей, но состоящую из применений штриха Шеффера. Уже есть заготовка на Haskell, но можно ей не пользоваться.