Задачи и решения отборочного тура олимпиады ДМиТИ 2014-2015

Все задачи, манипуляторы и решения доступны участникам на сайте олимпиады. Все предложенные задачи оценивались одинаковым числом баллов.

Графы. Конструирование непланарного графа

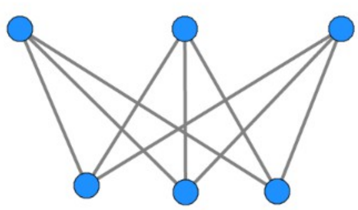
Сопровождающий задачу манипулятор позволяет перемещать вершины графа. При этом, если две вершины в исходном положении были соединены ребром, то это соединение сохранится и в новом положении, при этом новых рёбер не появится.

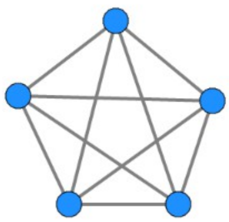
Планарным называется граф, вершины которого можно переместить так, что рёбра графа не будут пересекаться (совмещать вершины запрещается). Непланарным называется граф, в котором такими перемещениями вершин нельзя избавиться от пересекающихся рёбер.

<u>Задание 1</u>. Построить непланарный граф, удаление каждого ребра которого оставляет граф непланарным, а удаление любой пары рёбер делает граф планарным. Если второе условие не совместимо с первым, то лучшим считается решение, в котором число таких пар будет наибольшим (в процентном отношении ко всем парам рёбер).

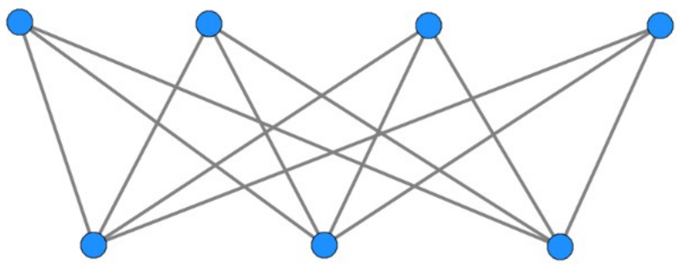
<u>Решение.</u>
<u>Замечание.</u> Автоматическая обработка решения порождает комментарии типа «После удаления одного ребра Ваш граф не является планарным в 8 из 18, после удаления двух - является планарным в 69 из 153».

Существуют два базовых непланарных графа, один из которых обязательно является частью любого непланарного графа: один из них называется полным пятивершинным графом, второй называется "три дома - три колодца" (см. рисунок).

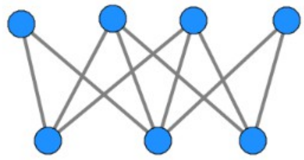


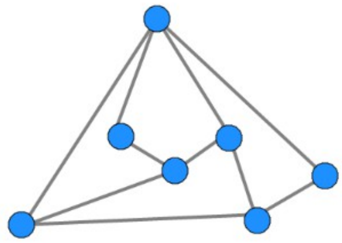


Таким образом, искомый граф должен получаться из одного из этих графов добавлением вершин и рёбер.
Лучшее решение можно получить рассмотрением нескольких таких графов и анализом того, какой из них лучше по указанным в задаче критериям.
Рассмотрим, например, граф, который можно назвать "четыре дома - три колодца" (см. рисунок).



Удаление одного ребра сохраняет непланарность, так как в качестве подграфа будет всегда иметься граф "три дома - три колодца". При удалении двух рёбер, к сожалению, непланарность сохранится, если рёбра ведут от одного "дома" (таких вариантов 12). В то же время выбор любых других пар рёбер приводит к планарному графу (см. рисунок). Всего пар имеется C(12;2)=66, значит "хороших" пар 66-12=54, они составляют 54/66=9/11 (82%).





Замечание. Поясним, что значит формулировка "является частью любого непланарного графа", использованная в начале объяснения. Это означает, что этот граф может быть получен с помощью отбрасывания части вершин и рёбер и заменой пары рёбер, смежных вершине, у которой других рёбер нет, одним ребром.

Графы. Перечисление неизоморфных графов

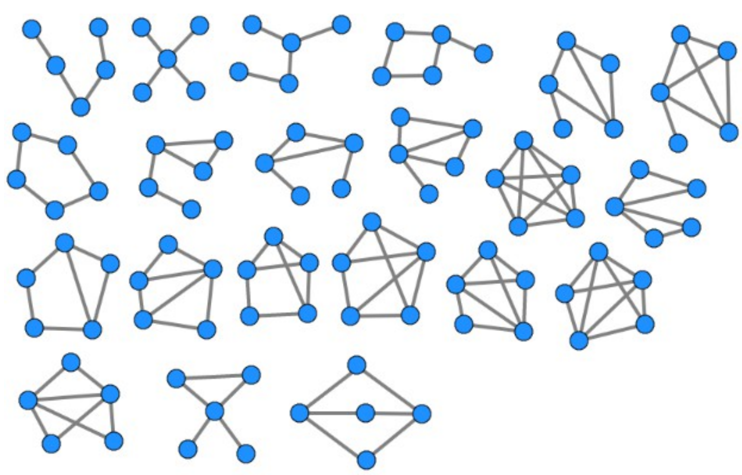
Сопровождающий задачу манипулятор позволяет перемещать вершины графа. При этом, если две вершины в исходном положении были соединены ребром, то это соединение сохранится и в новом положении, при этом новых рёбер не появится (совмещать вершины запрещается).

Говорят, что графы изоморфны, если описанными выше перемещениями вершин графы можно сделать одинаковыми (и их можно будет, например, совместить параллельным переносом).

Задание 2. Построить все неизоморфные между собой связные графы на 5 вершинах. Если будут построены не все графы, лучшим будет считаться решение с большим числом графов. Если в состав ответа войдут изоморфные между собой графы, то из числа правильно указанных графов будет вычтен штраф в количестве «лишних» графов.

Замечание. Автоматическая обработка решения порождает комментарии типа «Вы перечислили 17 из 21 возможных неизоморфных графов».

Ответ. Приводим картинку с полным перечислением всех 21 различных (неизоморфных) связных графов на 5 вершинах.



Машина Тьюринга. Двойной инкремент

<u>Манипулятор к задаче</u>:

<u>Задание 3</u>. Создайте машину Тьюринга, увеличивающую любое число в троичной системе счисления на 2. Используйте минимальное количество состояний.

Примечание. Для примера указана МТ, заменяющая число нулями.

<u>Замечание</u>. Автоматическая обработка решения порождает комментарии типа «Ваша МТ при обработке строки 2222222 не получила на ленте 10000001».

Решение.

Поскольку сложение чисел начинается с младших разрядов, каретку надо передвинуть вправо. Для этого нужно, не меняя состояния, пропускать все символы, пока не встретится *. После этого нужно сдвинуть каретку на 1 шаг влево - это и будет место младшего разряда. Далее перейдем к новому состоянию (q1), в котором начнём процесс сложения:

s0 [1] -> s0 [1] R
s0 [2] -> s0 [2] R
s0 [0] -> s0 [0] R
s0 [*] -> q1 [*] L

Если в последнем разряде стоит 0, то достаточно заменить его на 2 и завершить алгоритм. q1 [0] -> f [2] N

Если в последнем разряде стоит 1, то нужно запомнить 1 переноса в следующий разряд (перейдя в состояние q2) , а в последний разряд записать 0 и перейти к более старшему разряду (шаг влево).

q1 [1] -> q2 [0] L

Если в состоянии "единица в уме" встречается 0, то записываем в этот разряд 1 и останавливаемся, если 1, то записываем в этот разряд 1 и также останавливаемся. Если же в разряде встречаем 2, то записываем 0 и продолжаем движение влево в этом же состоянии.

q2 [0] -> f [1] N
q2 [1] -> f [2] N
q2 [2] -> q2 [0] L

Если в последнем разряде стоит 2, то аналогично предыдущему нужно запомнить 1 переноса в следующий разряд (перейдя в состояние q2) , а в последний разряд записать 1 и перейти к более старшему разряду (шаг влево). Остальное уже рассмотрено выше.

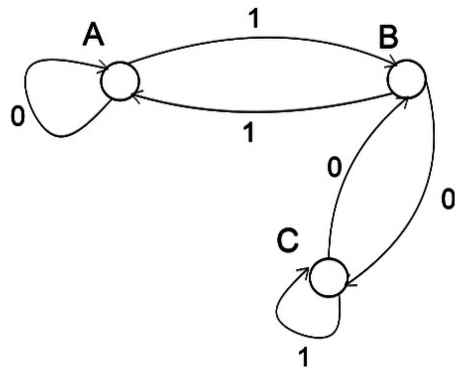
q1 [2] -> q2 [1] L

Если движением влево мы доходим до *, это означает, что число состояло из одних двоек (222...2) и при добавлении 2 длина числа увеличивается и нужно * заменить 1, после чего можно завершить работу.

q2 [*] -> f [1] N

Регулярные выражения и конечные автоматы

На рисунке изображен граф распознающего конечного автомата.

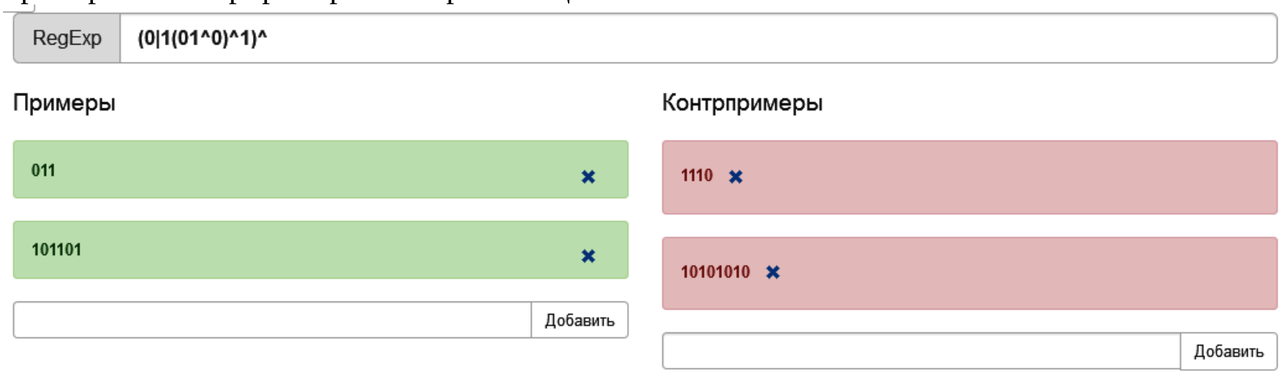


Говорят, что автомат распознает цепочку символов, если её можно получить следующим образом. Начать в вершине А и двигаться по рёбрам графа (включая, петли), записывая по мере их прохождения символы, приписанные этим рёбрам. Завершить обход надо в конечной вершине, которой в нашем примере является также вершина А.

Множество цепочек, распознаваемых автоматом, образуют автоматный язык, который определяется заданным автоматом. Любой автоматный язык может быть описан регулярным выражением (этот результат называется теоремой Клини).

<u>Задание 4.</u> Постройте регулярное выражение для языка, определенного заданным автоматом. Лучшим считается более короткое регулярное выражение.

<u>Замечание</u>. Автоматическая обработка решения (в процессе работы с задачей) позволяет участнику проводить эксперименты и испытывать построенное регулярное выражение на примерах и контрпримерах конкретных цепочек.



Решение.

Решение можно представить последовательным избавлением от петель.

Например, самая "дальняя" петля в вершине С соответствует множеству строк из одной цифры 1 (если будем "ходить" по петле многократно) и описывается выражением 1^.

Теперь можно описать все пути из вершины В в вершину С и обратно: сначала идём по стрелочке 0 (ВС), потом "крутимся" на петле с 1, а потом возвращаемся по стрелочке 1 (СВ). Это записывается как 01^0.

Однако такой путь мы можем повторить любое число раз, это можно записать выражением (01^0)^.

Аналогично путь из А в В (с промежуточным переходом в С) можно описать так: 1(01^0)^1.

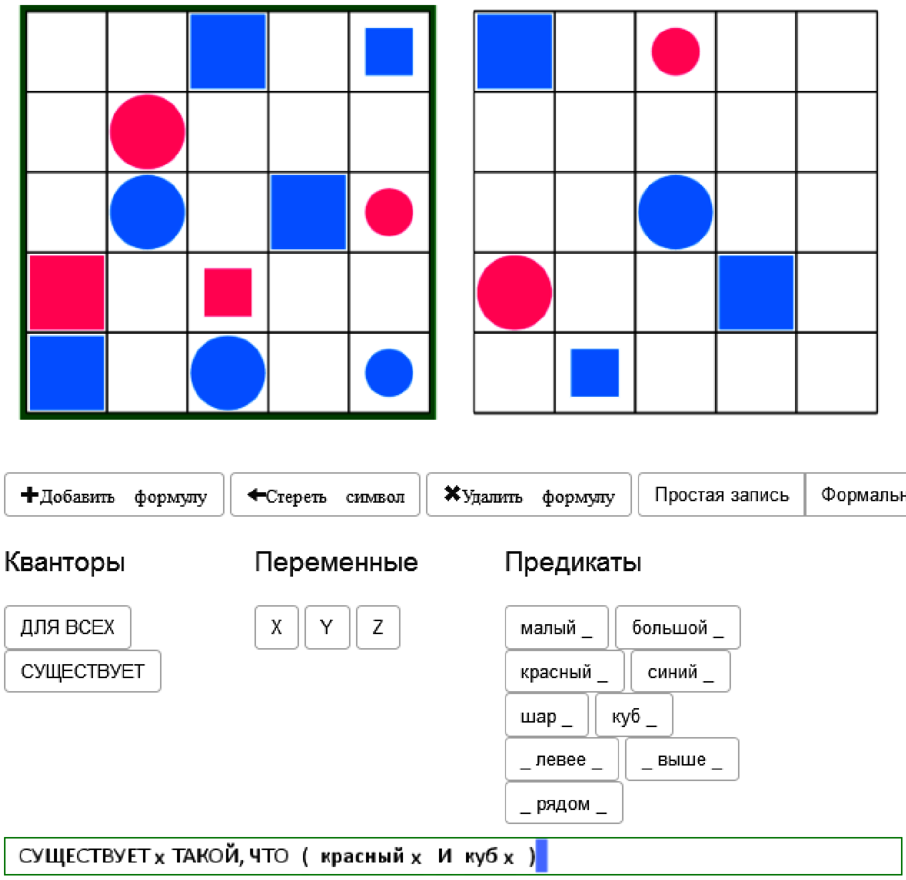
Но такие пути можно чередовать петлями из А в А, что записывается альтернативой 0|1(01^0)^1.
Наконец, такие проходы могут повторяться, чередуясь, любое количество раз. Получаем ответ: (0|1(01^0)^1)^.

Обращаем внимание, что могут быть и другие регулярные выражения, правильно описывающие этот автоматный язык (скорее всего, они будут более длинными).

Мир Тарского

Задание 5. Напишите формулу, которая будет верна только для левой картинки. Лучшей считается более короткая формула.

Манипулятор к задаче (на нём демонстрируется одно из правильных решений):

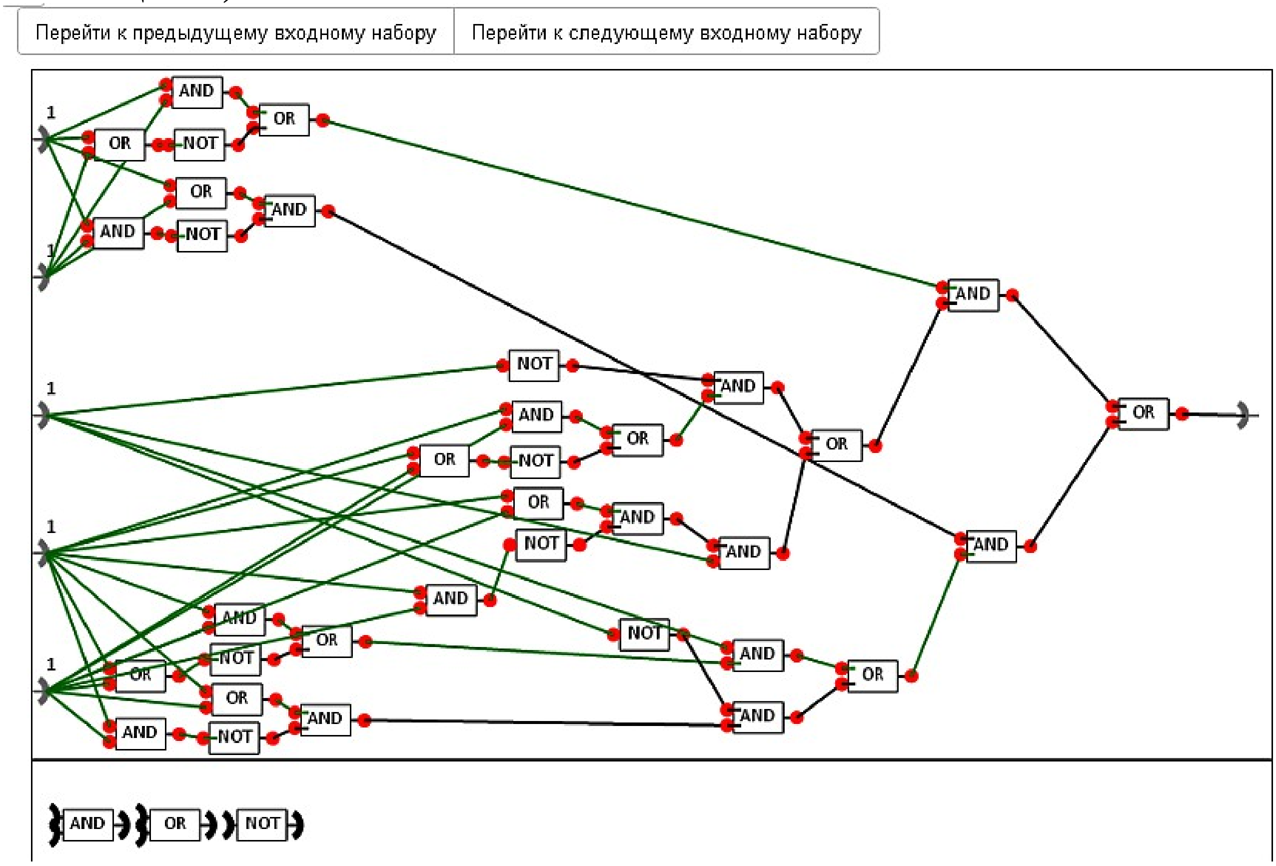


Замечание. Автоматическая обработка решения (в процессе работы с задачей) позволяет участнику проводить эксперименты и испытывать разные логические формулы, получая в качестве реакции рамку вокруг той картинки, для которой утверждение, заданное формулой, выполнено.

Функция проверки четности числа единиц

Задание 6. Постройте логическую схему, которая будет определять чётность числа единиц во входном наборе, то есть выдавать значение 1 ("истина") тогда и только тогда, когда количество единиц во входном наборе четно.

Манипулятор к задаче (на нём демонстрируется одно из правильных решений): манипулятор позволяет строить логические схемы, подавать поочередно различные комбинации входных сигналов и наблюдать результат (соединения, по которым подаётся сигнал - «1» - выделены зелёным цветом).



Обозначим ЧЁТ(x;y) логическую функцию (схему), которая даёт 1, если на входе два 0 или две 1 (то есть определяет чётность набора), аналогично её отрицание для удобства обозначим НЕЧ(x;y). Тогда, схему проверки чётности для 5 входов x, y, z, u, v можно свести к схемам с 2 и 3 входами следующим образом:

ЧЁТ(x;y;z;u;v)=ЧЁТ(x;y) И ЧЁТ(z;u;v) ИЛИ НЕЧ(x;y) И НЕЧ(z;u;v)=

=ЧЁТ(x;y) И {НЕ z И ЧЁТ(u;v) ИЛИ z И НЕЧ(u;v)} ИЛИ

НЕЧ(x;y) И {НЕ z И НЕЧ(u;v) ИЛИ z И ЧЁТ(u;v)}

Для двух переменных формулы уже записать нетрудно:

ЧЁТ(x;y)=НЕ x И НЕ y ИЛИ x И y = НЕ (x ИЛИ y) ИЛИ x И y

НЕЧ(x;y)=(НЕ x ИЛИ НЕ y) И (x ИЛИ y)= НЕ (x И y) И (x ИЛИ y)

Результирующая схема показана на рисунке.

Замечание. Если в качестве логических элементов использовать операцию "исключающее ИЛИ" (XOR), которая может рассматриваться как операция нахождения остатка от деления суммы входов на 2, то ответ получается очень быстро:

ЧЁТ(x;y;z;u;v)=НЕ (x XOR y XOR z XOR u XOR v)