

Исправления

1)[0|1(0(1)*0)*1][0|1(0(1)*0)*1]*

Сначала я понял номер, как-будто я считываю число с конца, поэтому автомат был некорректен. Сейчас же идея в том, что каждый новый символ домножает прошлый остаток на 2, и добавляет 0 или 1 в зависимости от переданного символа. Тогда видно что 0 сохраняет корректность. а второй вариант регулярки - это просто описание того, как можно походить по автомату и вернуться в исходное состояние, довольно ясно.

При этом последовательность добавлена хотя бы один раз, чтобы отсеять пустые строки. Потом эту последовательность мы можем повторять сколько угодно.

2)Минимальное описание, как был построен этот автомат, что "значат"вершины:

Заметим что нам практически полностью достаточно лишь поддерживать модуль по 3, т.к. модуль пятёрки зависит лишь от последней идеи. Вершины 0,1,2 это модули тройки, в случае если число не кратно 5, 00 - вершина, кратная сразу 15. А если число не кратно 3, то какая разница, какое оно по модулю 5 собственно.

В таком случае ясно как проходят все рёбра, автомату достаточно информации.

Примечание: Можно добавить вершину начала, чтобы отрезать случаи с лишними нулями, но с точки зрения задания смысловой нагрузки это не имеет, а строить это руками неприятно.

3) Ну давай посмотрим на получившийся автомат. Что мы точно в нём знаем: Внутри себя вершины автомата все различны - это следовало из упрощения. Сейчас есть два минимальных автомата мощностью n (кол-во состояний), и все эти n состояний для каждого из них лежат в разных n классах эквивалентности. Значит всегда есть хотя бы n состояний после упрощения. Теперь подумаем, что означает, что вершин стало ровно n , это будет значить что у нас есть n классов эквивалентности в каждом из которых по 2 вершины, которые очевидно из разных половин. А это в свою очередь равносильно тому, что автоматы одинаковы, что все их вершины можно разбить на пары, которые являются одинаковыми состояниями. на самом деле это почти про то же, что ты говорил на паре про упрощение автоматов.