1. $\mathscr{L}_1 = \left\{ \nu_1 z \nu_2 z \mid |z| \geqslant 2 \;\&\; z, \nu_2 \in \{a, b, c\}^* \;\&\; \nu_1 \in \{a, b\}^* \right\}$

2. Множество троичных чисел, кратных 5.

3. Множество трасс грамматик с правилами вида $N_i \to \gamma N_j$, $N_i \to N_j \gamma$, $N_i \to \varepsilon$.

4. Множество трасс грамматики G_k с правилами вида $S \to S_1 S_1$, $S_1 \to S_2 S_2 \ldots$, $S_k \to a$, $S_k \to SS$.

Решение задачи I

«Интуитивно» язык не регулярный — есть требование вхождения одинаковых подслов, которые «по идее» можно как-то отделить от аморфных значений ν_1 и ν_2. Попробуем провести это отделение более последовательно.

Два момента в данном языке обращают на себя внимание:

- словарь (язык) ν_1 отличается от словаря смежного с ним слова z. Это значит, что если взять первую букву z из разности языка z и языка ν_1, то никакое значение ν_1 не сможет её поглотить.

- словарь (язык) ν_2 точно такой же, как у смежных вхождений z. Поэтому в принципе ν_2 может поглотить любой суффикс первого вхождения z, а также любой префикс второго вхождения.

Чтобы слово ν_2 не могло поглотить префикс второго вхождения z, надо сделать так, чтобы его положение определялось однозначно. Мы уже знаем, что выгодно взять z начинающимся с буквы c (это исключит поглощение префикса z значением ν_1). Значит, если в слове будет всего две буквы c, то ровно одна из них должна начинать первое вхождение z, и ровно одна должна начинать второе вхождение. Осталось дополнить z достаточно длинными суффиксами, исключающими возможность «накачки» их одновременно.

Кандидат на контрпример — серия слов $ca^{n+2}ca^{n+2}$, где n — длина накачки.

Теперь можно доказать нерегулярность языка посредством теоремы Майхилла–Нероуда. Действительно, слова вида $ca^m ca^{m+k+1}$ языку не принадлежат, а $ca^{m+k}ca^m$ — точно принадлежат (при этом $z = ca^m$, $\nu_2 = a^k$), что порождает нижнетреугольную матрицу принадлежности: наименования строк здесь — это префиксы слов, а столбцов — соответствующие суффиксы.

	ca	ca^2	ca^3	$\ldots$
ca	$+$	$-$	$-$	$-$
ca^2	$+$	$+$	$-$	$-$
ca^3	$+$	$+$	$+$	$-$
$\ldots$		$\ldots$		

Также можно использовать этот контрпример для построения короткого доказательства нерегулярности $\mathscr{L}_1$, если пересечь его с языком (регулярным) $R = ca^*ca^*$. В слове $ca^{n+2}ca^{n+2}$, принадлежащем пересечению $R \cap \mathscr{L}_1$, можно накачивать лишь фрагмент, состоящий только из букв a, чтобы не выйти из языка R. Пусть такой фрагмент имеет длину k (где $k > 0$). Тогда при отрицательной накачке получится слово $ca^{n-k+2}ca^{n+2}$, которое не входит в $\mathscr{L}_1$.

Наиболее неприятный путь — прямое применение леммы о накачке к слову $ca^{n+2}ca^{n+2}$ без использования свойств замыканий. На этом пути придётся разобрать два случая.

- Фрагмент накачки имеет вид a^k. Этот случай аналогичен уже рассмотренному в решении с пересечением.

- Фрагмент накачки имеет вид ca^k. Отметим, что $k < n$. Тогда при положительной накачке в одну итерацию получим слово $ca^kca^{n+2}ca^{n+2}$. Поскольку это слово начинается с c, то значение z должно начинаться с c, а значит, должно заканчиваться на a^{n+2} (потому что первое с конца вхождение c уж точно будет относиться к z). Но это значит, что z должна содержать также и фрагмент ca^kc (иначе первое вхождение z не сможет заканчиваться на a^{n+2}), а он в этом слове не повторяется.

В этой задаче у большинства возникла одна из двух проблем:

- Или взято значение z с префиксом в языке $\{a, b\}$, из-за чего оно смешалось со значением v_1.

- Или взято значение z, равное c^n (очевидно, вы заметили, что в противном случае анализу мешает v_1). Тогда мы можем весь суффикс z, кроме двух первых букв, положить в v_2. Вообще, в этой задаче не стоит брать значения z, значение префикс-функция у которых больше, чем 0 (и осторожнее с такими словами в других задачах).

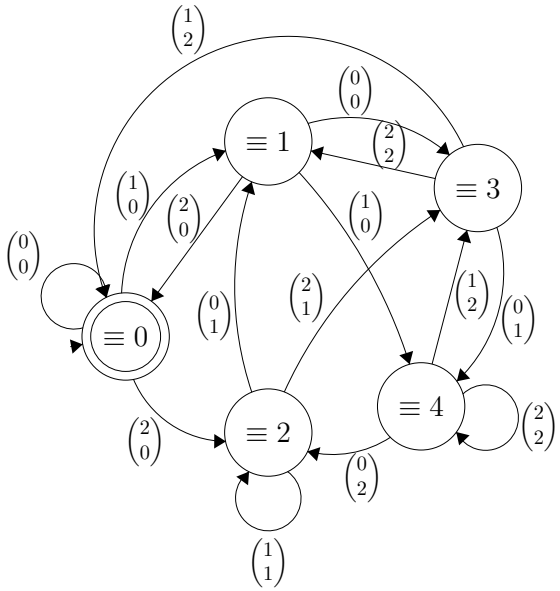
Решение задачи II

Если x кратно 5, то $\exists y(x = 5 \cdot y)$. Что указывает: можно использовать метод построения автоматов для пар $\binom{x}{y}$, таких что $x = 5 \cdot y$, а потом взять в нём проекцию по x.

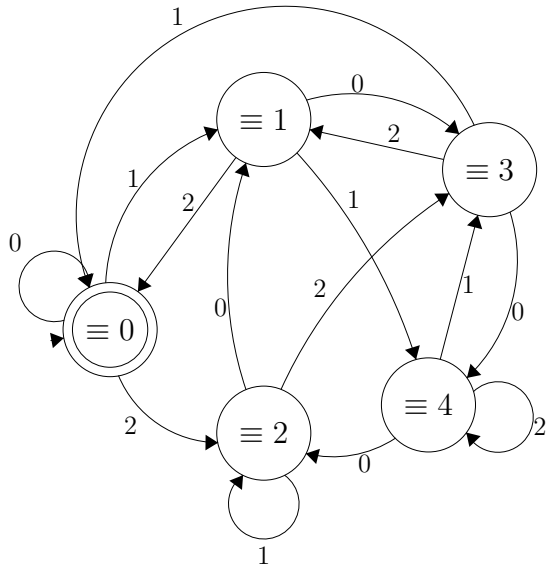
Для этого сначала найдём все пути, ведущие в ловушки, а именно, определим критическую разницу между x и y снизу и сверху, которая не может быть исправлена приписыванием никаких младших разрядов. Если $x - 5y = k$, то максимально большой выигрыш в пользу x в следующем разряде будет, если к x припишется 2, а к y — 0. Тогда очередное значение $x' - 5y'$ будет $3x + 2 - 15y = 3k + 2$. $3k + 2 \leqslant k$, если $k \leqslant -1$, и это критическая разница в пользу y, которая гарантирует, что все пути в автомате, на которых получено это значение, будут тупиковыми.

Обратно, максимально большой выигрыш в пользу y в следующем разряде будет при приписывании к x нуля, а к y — двойки. Тогда $x' - 5y' = 3x - 15y - 10 = 3k - 10$. Если $3k - 10 \geqslant k$, то $k \geqslant 5$, поэтому все пути в автомате, на которых встретится такая разница между x и $5y$, также будут тупиковыми.

Значит, в состояниях, не являющихся ловушками, величина $x - 5y$ варьирует от 0 до 4. Дальнейшее построение автомата — техническая процедура.



Автомат-проекция указанного ДКА по первому компоненту также будет детерминированным (вообще говоря, это не гарантируется для проекций).



К аналогичному решению задачи можно было прийти и без проекций. Действительно, рассмотрим, как будут изменяться остатки от деления x на 5 при приписывании очередного разряда.

Остаток	Приписан 0	Приписана 1	Приписана 2
0	$0 \cdot 3 + 0 = 0$	$0 \cdot 3 + 1 = 1$	$0 \cdot 3 + 2 = 2$
1	$1 \cdot 3 + 0 = 3$	$1 \cdot 3 + 1 = 4$	$1 \cdot 3 + 2 = 0$
2	$2 \cdot 3 + 0 \equiv 1$	$2 \cdot 3 + 1 \equiv 2$	$2 \cdot 3 + 2 \equiv 3$
3	$3 \cdot 3 + 0 \equiv 4$	$3 \cdot 3 + 1 \equiv 0$	$3 \cdot 3 + 2 \equiv 1$
4	$4 \cdot 3 + 0 \equiv 2$	$4 \cdot 3 + 1 \equiv 3$	$4 \cdot 3 + 2 \equiv 4$

Минимальность ДКА почти очевидна: состояния $\equiv 1, \equiv 3$ различимы друг от друга и от остальных состояний на переходах по 2 и 1 соответственно; если состояния $\equiv 1$, $\equiv 3$ доказуемо различимы, то $\equiv 2$ и $\equiv 4$ после этого можно различить поведением на переходах по любому значению. На основе этих наблюдений построим таблицу классов эквивалентности.

	00	01	02	10	11
ε	+	−	−	−	−
1	−	+	−	−	−
2	−	−	+	−	−
10	−	−	−	+	−
11	−	−	−	−	+

Она не только свидетельствует, что построенный ДКА минимален, но и обосновывает, что никакой НКА для этого языка не может иметь меньше 5 состояний (согласно расширенному критерию Глайстера–Шаллита, нижняя граница на число состояний в НКА — число строк в верхнетреугольной матрице).

Осталось разобраться с вопросом о регулярном выражении для данного языка. Минимальность представленного ДКА даже в классе недетерминированных автоматов наводит на мысль, что регулярка может получиться очень длинной. И действительно, после устранения состояний $\equiv 3$ и $\equiv 4$ мы получаем ДКА, представляющий собой полный граф переходов из трёх вершин, а языки таких ДКА порождают максимальное разрастание длины при переходе к регулярным выражениям. Таким образом, мы имеем дело именно с таким языком, когда представление в форме ДКА оказывается экспоненциально более выгодным, чем в форме регулярного выражения (как минимум, при применении алгоритма устранения состояний напрямую).

Решение задачи III

Множество трасс грамматик с правилами вида $N_i \to \gamma N_j$, $N_i \to N_j \gamma$, $N_i \to \varepsilon$

Трассы — это слова в алфавите $\Sigma \cup \{N_i\} \cup \{\to\} \cup \{;\}$, где ; — разделитель между смежными применениями правил.

Поскольку грамматика в задаче содержит по одному нетерминалу в правых частях правил, то каждое очередное правило будет применяться именно к этому нетерминалу. Далее воспользуемся следующими двумя фактами:

- число правил грамматики конечно
- раскрываемый в новом правиле нетерминал является либо крайним (то есть непосредственно предшествует ;), либо вторым с края (за ним следует единственная буква $\gamma_i \in \Sigma$).

Построим регулярные языки R_1 и R_2, соответствующие каждому из указанных условий, тогда корректные трассы грамматики будут принадлежать языку $R_1 \cap R_2$.

- R_1 определяет, что трасса составлена из правил именно данной грамматики. То есть $R_1 = (N_1 \to \Phi_1 \mid \ldots \mid N_k \to \Phi_k;)^*(M_0 \to \varepsilon \mid \ldots M_s \to \varepsilon)$, где $N_i \to \Phi_i$ — рекурсивные правила грамматики (т.е. содержащие нетерминал в правой части); $M_i \to \varepsilon$ — финальные правила грамматики.
- R_2 определяет, что в трассе раскрываются именно те нетерминалы, которые были порождены в предшествующей правой части. То есть $R_2 = S \to .?(N_1.?;N_1 \to .? \mid \ldots \mid N_k.?;N_k \to .?)^*\varepsilon$, где $.?$ — опциональная возможность прочитать один символ, N_i — все нетерминалы грамматики. Этот язык позволяет правилам не содержать ни одного терминального символа, либо иметь несколько вхождений нетерминалов в правые части. Но это в данном случае не существенно, поскольку строки языка R_2, содержащие применения таких правил в трассе, не входят в язык R_1 и потому не содержатся в пересечении.

Определять язык корректных трасс путём явного описания регулярного выражения было бы намного труднее: пришлось бы ветвить регулярное выражение по всем возможным переходам от одного нетерминала к другому. В худшем случае, это привело бы к экспоненциальному росту регулярного выражения. Действительно, если объявить базисными состояниями ДКА

нетерминалы, и задать правила перехода из каждого нетерминала в каждый другой по различным терминальным символам, мы получим чуть-чуть другое представление ДКА — полного графа, минимальное регулярное выражение для которого экспоненциально зависит от числа нетерминалов.