1. Язык, описывающийся следующей атрибутной грамматикой:

$$
\begin{array}{lcl}
\mathsf{S} \to \mathsf{AT} & ; & \mathsf{T}.\mathtt{rng} > \mathsf{A}.\mathtt{iter} \\
\mathsf{A} \to \mathtt{a}\mathsf{A} & ; & \mathsf{A}_0.\mathtt{iter} := \mathsf{A}_1.\mathtt{iter} + 1 \\
\mathsf{A} \to \varepsilon & ; & \mathsf{A}.\mathtt{iter} := 0 \\
\mathsf{T} \to \mathsf{TcT} & ; & \mathsf{T}_0.\mathtt{rng} := \mathtt{max}(\mathsf{T}_1.\mathtt{rng}, \mathsf{T}_2.\mathtt{rng}) \\
\mathsf{T} \to \mathsf{K} & ; & \mathsf{T}.\mathtt{rng} := \mathsf{K}.\mathtt{rng} \\
\mathsf{K} \to \mathtt{a}\mathsf{K} & ; & \mathsf{K}_0.\mathtt{rng} := \mathsf{K}_1.\mathtt{rng} + 1 \\
\mathsf{K} \to \mathtt{b}\mathsf{K} & ; & \mathsf{K}_0.\mathtt{rng} := 0 \\
\mathsf{K} \to \varepsilon & ; & \mathsf{K}.\mathtt{rng} := 0
\end{array}
$$

2. Язык $\{wcvw_{\mathtt{pref}}zw_{\mathtt{suff}} \mid w, z \in \{\mathtt{a}, \mathtt{b}\}^* \ \& \ v \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*\}$. Здесь $w_{\mathtt{pref}}$ — непустой префикс слова w; $w_{\mathtt{suff}}$ — непустой суффикс слова w.

3. Язык, описывающийся следующей атрибутной грамматикой:

$$
\begin{array}{lcl}
\mathsf{S} \to \mathsf{SbS} & ; & \mathsf{S}_0.\mathtt{iter} := 2 \cdot \mathsf{S}_1.\mathtt{iter}, \mathsf{S}_1.\mathtt{iter} == \mathsf{S}_2.\mathtt{iter} \\
\mathsf{S} \to \mathtt{a} & ; & \mathsf{S}.\mathtt{iter} := 1
\end{array}
$$

4. Язык SRS с правилами $\mathtt{a} \to \mathtt{bab}$, $\mathtt{ba} \to \mathtt{ab}$ над базисным словом $\mathtt{aa}$ (единственным).

Решение задачи IV

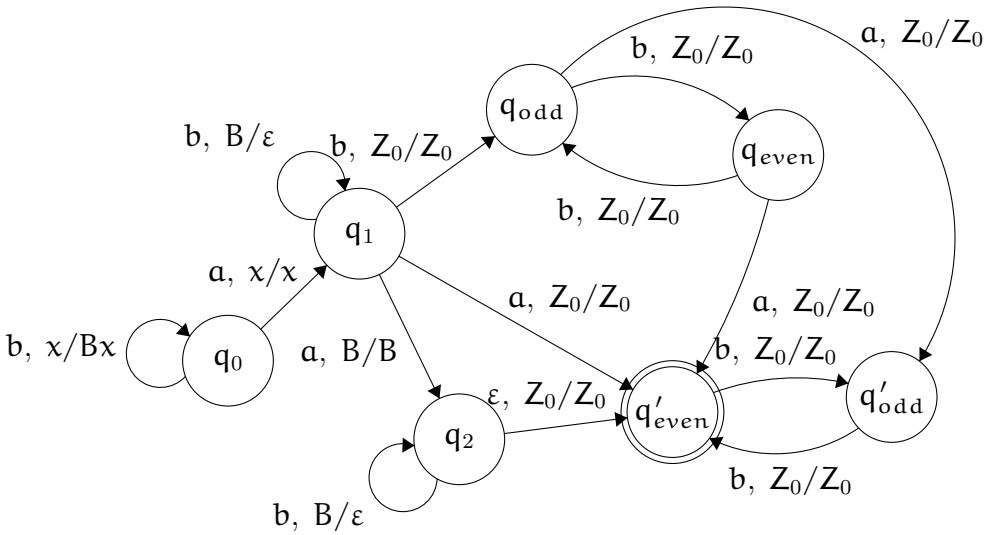
$\mathscr{L}$ — язык SRS с правилами $\mathtt{a} \to \mathtt{bab}$, $\mathtt{ba} \to \mathtt{ab}$ над базисным словом $\mathtt{aa}$ (единственным).

Правила переписывания показывают, что во-первых, буквы $\mathtt{a}$ в словах языка всегда ровно две, а во-вторых, относительно слов, порождаемых по первому правилу $\mathtt{b}^n\mathtt{ab}^{n+m}\mathtt{ab}^m$ они могут перемещаться только влево, причём как угодно далеко. При этом, если мы сначала применим первое правило, а потом второе, то получим композицию $\mathtt{a} \to \mathtt{abb}$, а если мы сначала применим второе, а потом первое — то $\mathtt{ba} \to \mathtt{babb}$, то есть правила перестановочны, коль скоро есть хотя бы одна буква $\mathtt{b}$ слева от $\mathtt{a}$. Значит, чтобы получить произвольное слово языка, можно сначала максимально применить первое правило, а потом второе.

Обратим также внимание, что если какое-то слово получилось из второй слева буквы $\mathtt{a}$ применением k правил 1, то мы могли бы получить его, применив те же k правил 1 к самой левой букве $\mathtt{a}$, а затем к ней же — k правил 2, и ко второй букве $\mathtt{a}$ — k правил 2. Пусть область, порождаемая правилами

$a \to bab$, обозначена красным для первой буквы a и синим — для второй. Тогда $ab^k ab^k$ — то же, что $b^k ab^k a \to ab^{2k} a \to ab^k ab^k$. Поэтому можно принять допущение, что правила $a \to bab$ применяются только к самой левой букве a.

Таким образом, можно установить, что слова языка $\mathscr{L}$ имеют вид $b^{n_1} ab^{n_2} ab^{n_3}$, где $n_2 + n_3 \geqslant n_1$ и сумма $n_1 + n_2 + n_3$ чётна. Это детерминированный контекстно-свободный (но не регулярный) язык, DPDA для которого выглядит так:



Язык не является LL(k), в чём можно убедиться, рассмотрев слова $b^{n+k} ab^{n+k} a$ и $b^{n+k} aab^{n+k}$ с предполагаемым lookahead для грамматики в ГНФ, равным k, и наблюдаемым префиксом b^n. В силу бесконечности классов по Майхиллу–Нероуду в языке b^* относительно $\mathscr{L}$, стек на этом префиксе разрастается как угодно сильно, в частности, можно взять такое n, чтобы он уж точно содержал $k+3$ элемента. Тогда в слове $b^{n+k} aab^{n+k}$ последний элемент стека заведомо разворачивается в $C_1 = b^s$ (т.к. префикс $b^k a^2$ точно прочитан раньше: каждый из $k+2$ первых элементов стека считал как минимум по одной букве). В слове $b^{n+k} ab^{n+k} a$ последний элемент стека точно разворачивается в $C_2 = b^t a$. Этот последний элемент один и тот же, и lookahead у него в обоих случаях равен концу строки. Подменой C_1 на C_2 (или наоборот) получаем слово не из $\mathscr{L}$.

Язык не является VPL-языком. По условию, в VPL-языке все терминалы строго делятся на непересекающиеся категории: вызывающие, возвращаю-

щие и внутренние. Если назначить $\mathtt{b}$ вызывающим терминалом, то стек (почти) всегда будет только наполняться; если назначить $\mathtt{b}$ читающим — стек (почти) всегда будет пуст. И тот, и другой случай приводят к тому, что количество букв $\mathtt{b}$ слева и справа от первой буквы $\mathtt{a}$ сравнить невозможно. «Почти» здесь — указание, что в принципе ничего не мешает назначить букву $\mathtt{a}$ читающей или вызывающей, однако поскольку их только две в слове, они изменят поведение стека лишь конечным образом.

Решение задачи III

Посмотрим на несколько первых итераций, порождающих слова языка $\mathscr{L}$:

$\mathtt{S} \to \mathtt{SbS}$; $\mathtt{S}_0.\mathtt{iter} := 2 \cdot \mathtt{S}_1.\mathtt{iter}, \mathtt{S}_1.\mathtt{iter} == \mathtt{S}_2.\mathtt{iter}$

$\mathtt{S} \to \mathtt{a}$; $\mathtt{S}.\mathtt{iter} := 1$

Слово $\mathtt{aba}$ семантическому критерию удовлетворяет, т.к. оно получается разбором двух нетерминалов с атрибутом $\mathtt{iter} = 1$. При этом для $\mathtt{aba}$ соответствующий атрибут равен 2.

Из этих базовых слов мы не можем получить слово $\mathtt{ababa}$, т.к. атрибуты его левых и правых частей не равны. То же самое верно и про $\mathtt{ababa}$. Слово $\mathtt{ababaaba}$ построить можно, и его атрибут $\mathtt{iter}$ будет равен 4.

Мы видим, что на каждой новой итерации мы получаем возможность скомбинировать только два слова с предыдущей итерации, при этом атрибут $\mathtt{iter}$ полученного слова будет не равен атрибутам $\mathtt{iter}$ никаких ранее построенных слов. Если на $\mathtt{i}$-й итерации было построено слово $(\mathtt{ab})^{\mathtt{k}}\mathtt{a}$, то на $\mathtt{i}+1$-й итерации будет построено слово $(\mathtt{ab})^{\mathtt{k}}\mathtt{ab}(\mathtt{ab})^{\mathtt{k}}\mathtt{a} = (\mathtt{ab})^{2\mathtt{k}+1}\mathtt{a}$. Отсюда следует, что множество слов языка в свёрточной форме выглядит так: $\{(\mathtt{ab})^{2^{\mathtt{i}}-1}\mathtt{a}\}$.

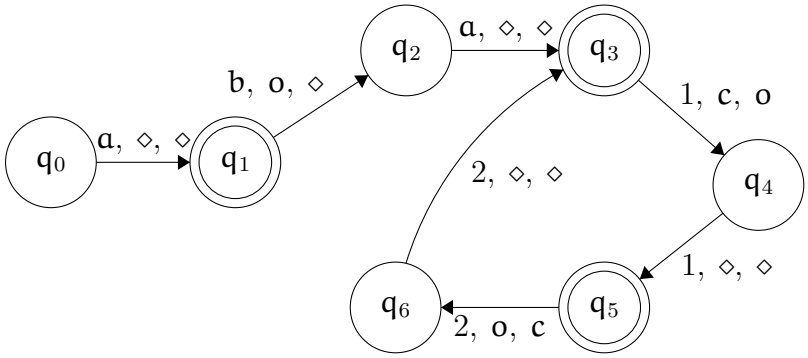
Проще всего обосновать не КС-свойство для данного языка с помощью теоремы Париха. Действительно, кратности букв $\mathtt{b}$ (впрочем, как и $\mathtt{a}$) в словах языка описываются экспоненциально растущей функцией, которая не может быть представлена как объединение линейных. Доказательство не КС-свойства языка $\mathscr{L}$ с помощью леммы о накачке предоставляется читателю.

Построим backref-регулярку для данного языка, пользуясь наблюдением, что $2^{\mathtt{n}} - 1 = 1 + 2 + 2^2 + \cdots + 2^{\mathtt{n}-1}$.

$$\mathtt{a}([_1\mathtt{ba}]_1([_2\mathtt{x}_1\mathtt{x}_1]_2[_1\mathtt{x}_2\mathtt{x}_2]_1)^*[_2\mathtt{x}_1\mathtt{x}_1]_2?)?$$

Заметим, что эту backref-регулярку можно преобразовать в детерминированный MFA. Единственной проблемой, которая в ней может возникнуть,

является переход от итерации Клини к последнему чтению x_1x_1, который решается тем, что промежуточное состояние внутри итерации перед чтением $[_1x_2x_2]_1$ нужно сделать финальным.



Решение задачи II

$$\mathscr{L} = \{wcvw_{\mathrm{pref}}zw_{\mathrm{suff}} \mid w, z \in \{a, b\}^* \ \& \ v \in \{a, b, c\}^*\}$$

Можно заметить, что для любого слова в $\mathscr{L}$ при рассмотрении $w_{\mathrm{pref}} = s_{\mathrm{fst}}w'_{\mathrm{pref}}$ и $w_{\mathrm{suff}} = w'_{\mathrm{suff}}s_{\mathrm{last}}$, где s_{fst} и s_{last} — первая и последняя буквы соответственно, получается слово вида $wcvs_{\mathrm{fst}}w'_{\mathrm{pref}}zw'_{\mathrm{suff}}s_{\mathrm{last}}$, в котором можно положить $z' = w'_{\mathrm{pref}}zw'_{\mathrm{suff}}$, а новыми префиксом и суффиксом — только первую и последнюю буквы. И это слово также принадлежит $\mathscr{L}$, однако его структура подходит под большее число случаев w, чем исходная. Поэтому язык $\mathscr{L}$ по факту представляет собой следующий:

$$\mathscr{L} = \{s_{\mathrm{fst}}ws_{\mathrm{last}}cvs_{\mathrm{fst}}zs_{\mathrm{last}} \mid s_{\mathrm{fst}}ws_{\mathrm{last}}, z \in \{a, b\}^* \ \& \ v \in \{a, b, c\}^*\}$$

Поскольку w, v, z здесь можно заменить регулярками $(a|b)^*$ и $(a|b|c)^*$, то данный язык регулярен и является объединением следующих языков:

- $ac(a|b|c)^*a(a|b)^*a$
- $bc(a|b|c)^*b(a|b)^*b$
- $a(a|b)^*ac(a|b|c)^*a(a|b)^*a$
- $a(a|b)^*bc(a|b|c)^*a(a|b)^*b$

- $\mathtt{b}(\mathtt{a}|\mathtt{b})^*\mathtt{ac}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{b}(\mathtt{a}|\mathtt{b})^*\mathtt{a}$
- $\mathtt{b}(\mathtt{a}|\mathtt{b})^*\mathtt{bc}(\mathtt{a}|\mathtt{b}|\mathtt{c})^*\mathtt{b}(\mathtt{a}|\mathtt{b})^*\mathtt{b}$

Префикс-свойство для него не выполняется, потому что внутри подвыражения $(\mathtt{a}|\mathtt{b}|\mathtt{c})^*$ или последнего подвыражения $(\mathtt{a}|\mathtt{b})^*$ может встретиться повтор последующего суффикса. В качестве контрпримера, подтверждающего это, достаточно взять слова $\mathtt{aacaa}$ и слово $\mathtt{aacaaa}$, принадлежащие $\mathscr{L}$.

Всякий регулярный язык является LL(1). Соответствующую грамматику можно построить, тупо взяв ДКА для него и преобразовав его в праволинейную грамматику с эпсилон-правилами для самой последней буквы (иначе будет LL(1)-конфликт; нигде, кроме как с финальными состояниями, конфликта быть не может в силу того, что автомат детерминирован).

Решение задачи I

Сначала посмотрим на атрибутную грамматику.

$$
\begin{array}{lcl}
\mathtt{S} \to \mathtt{AT} & ; & \mathtt{T}.\mathtt{rng} > \mathtt{A}.\mathtt{iter} \\
\mathtt{A} \to \mathtt{aA} & ; & \mathtt{A}_0.\mathtt{iter} := \mathtt{A}_1.\mathtt{iter} + 1 \\
\mathtt{A} \to \varepsilon & ; & \mathtt{A}.\mathtt{iter} := 0 \\
\mathtt{T} \to \mathtt{TcT} & ; & \mathtt{T}_0.\mathtt{rng} := \mathtt{max}(\mathtt{T}_1.\mathtt{rng}, \mathtt{T}_2.\mathtt{rng}) \\
\mathtt{T} \to \mathtt{K} & ; & \mathtt{T}.\mathtt{rng} := \mathtt{K}.\mathtt{rng} \\
\mathtt{K} \to \mathtt{aK} & ; & \mathtt{K}_0.\mathtt{rng} := \mathtt{K}_1.\mathtt{rng} + 1 \\
\mathtt{K} \to \mathtt{bK} & ; & \mathtt{K}_0.\mathtt{rng} := 0 \\
\mathtt{K} \to \varepsilon & ; & \mathtt{K}.\mathtt{rng} := 0
\end{array}
$$

Все атрибуты синтетические, кроме того, язык $\mathtt{K}$ явно регулярен, и атрибут $\mathtt{rng}$ в нём изменяется, только пока в префиксе встречаются исключительно буквы $\mathtt{a}$. То есть если $\mathtt{K} \to \mathtt{a}^{\mathtt{n}}(\mathtt{b}(\mathtt{a}|\mathtt{b})^*)?$, то $\mathtt{K}.\mathtt{rng} = \mathtt{n}$.

Теперь посмотрим на язык $\mathtt{T}$. Заметим, что его можно переписать в форме $(\mathtt{Kc})^*\mathtt{K}$, устранив левую рекурсию (поскольку язык $\mathtt{K}$ не содержит букв $\mathtt{c}$, а разделители между $\mathtt{K}$-развёртками не содержат $\mathtt{a}$ и $\mathtt{b}$, то можно $\mathtt{K}$ представить единственным токеном и далее, например, применить метод Блюма–Коха). При этом семантическое свойство будет выглядеть так: $\mathtt{T}.\mathtt{rng} = \mathtt{max}(\mathtt{K}.\mathtt{rng})$.

Осталось разобраться с атрибутом $\mathtt{A}$. Здесь, очевидно, если $\mathtt{A} \to \mathtt{a}^{\mathtt{m}}$, то $\mathtt{A}.\mathtt{iter} = \mathtt{m}$. Но именно в случае $\mathtt{A}$ есть неоднозначность границы разбора: если $\mathtt{S} \to \mathtt{AKcT}$, причём $\mathtt{K} \to \mathtt{a}^{\mathtt{n}}(\mathtt{b}(\mathtt{a}|\mathtt{b})^*)?$, $\mathtt{A} \to \mathtt{a}^{\mathtt{m}}$, то мы можем сделать

какие угодно разборы префикса a^{n+m} начиная от $K \to a^{n+m}(b(a|b)^*)?$, $A \to \varepsilon$ и кончая $K \to (b(a|b)^*)?, A \to a^{n+m}$. С точки зрения языка, наиболее выгодно положить $K \to a^{n+m}(b(a|b)^*)?$, $A \to \varepsilon$ — это даст возможность породить наиболее широкий класс слов.

Таким образом, условие на атрибуты вырождается в $T.rng > 0$, а грамматика для данного языка может быть преобразована в следующую форму:

$S \to T$	;	$T.rng > 0$
$T \to KcT$	;	$T_0.rng := max(K.rng, T_1.rng)$
$T \to K$	;	$T.rng := K.rng$
$K \to aK$	;	$K_0.rng := K_1.rng + 1$
$K \to bK$	;	$K_0.rng := 0$
$K \to \varepsilon$	;	$K.rng := 0$

Эта грамматика определяет регулярный язык $((a|b)^*c)^*a^+(a|b)^*(c(a|b)^*)$.

Если бы условие на атрибуты в корне было $T.rng < A.iter$, тогда наиболее выгодное с точки зрения языка решение по неоднозначности было бы $K \to (b(a|b)^*)?$, $A \to a^{n+m}$. То есть требовалось бы сравнить по длине максимальные подстроки из букв a в префиксах, идущих сразу после буквы c, с префиксом a^{n+m} (и если ни одной буквы c в слове нет, то это условие тривиально выполняется, если только в начале слова есть хотя бы одна буква a). Будем рассматривать только нетривиальную ситуацию — буквы c присутствуют. Упрощенная атрибутная грамматика станет такая:

$S \to ABcT$	;	$T.rng < A.iter$
$A \to aA$	;	$A_0.iter := A_1.iter + 1$
$A \to \varepsilon$	;	$A.iter := 0$
$T \to KcT$	;	$T_0.rng := max(T_1.rng, T_2.rng)$
$T \to K$	;	$T.rng := K.rng$
$K \to aK$	;	$K_0.rng := K_1.rng + 1$
$K \to B$	;	$K.rng := 0$
$B \to b(a|b)^* \mid \varepsilon$	;	атрибутов нет

В свёрточной форме язык переписывается как $\{a^n(b(a|b)^*)?(ca^i(b(a|b)^*)?)^+ \mid \forall i(n > i)\}$.

Похоже, что есть более чем двойственная взаимосвязь между блоками a^i (т.к. каждый из них нужно сравнивать с самым первым, что означало бы, что из первого блока нужно читать в стек несколько раз). Попробуем опровергнуть КС-свойство для полученного языка путём лёммы о накачке. Для этого сначала пересечём наш язык с $a^+ca^+ca^+$, чтобы избавиться от лишних подстрок, определяемых конечноавтоматным поведением, и от накачек, наращивающих число K-фрагментов.

Пусть p — длина накачки. Рассмотрим слово $a^{p+1}ca^pca^p$. Отдельно фрагмент a^{p+1} накачивать мы не можем — отрицательная накачка сразу же выведет из языка. Накачивать его вместе с фрагментом a^p тоже не получится — отрицательная накачка даст слово, в котором первый фрагмент не больше по длине, чем фрагмент a^p. Однако накачивать только a^pca^p (в любых вариациях) тоже нельзя — это сразу же приведёт к выходу из языка при повторной положительной накачке.

Можно заметить, что язык конъюнктивен. Действительно, для крайней пары нетерминалов A и K можно КС-свободно описать слова вида

$$a^{n+m+1}(b(a|b)^*|\varepsilon)c(a|b|c)^*ca^n(b(a|b)^*|\varepsilon)$$

Например, такой грамматикой:

$S' \to S'a \mid S'b \mid Cb \mid C$

$C \to aCa \mid aAc \mid aAcWc$

$A \to aA \mid bW \mid \varepsilon$

$W \to (a|b|c)W \mid \varepsilon$

Конъюнктивная грамматика готова:

$S \to S' \ \& \ ScK$

$S \to aK$

$K \to (a|b)K \mid \varepsilon$

Заметим, что этот язык — не DMFA. Используем замкнутость DMFA относительно пересечения с регулярными языками и пересечём наш язык с a^+ca^+. Получим язык слов вида $a^{n+m}ca^n$. Его префиксы a^* определяют бесконечное число классов эквивалентности по Майхиллу–Нероуду, поэтому для них можно применить JL (то есть положить v_n подсловом префикса a^+). Положим $v_n = a^n$. Для p_n есть два варианта: $p_n = a^{n+k}$ и $p_n = a^{n+k}ca^i$, где $i < 2n + k$. Существуют такие z_i, что слова $p_nv_nz_i$ входят в язык, а именно: $z_i = c$ в первом случае и $z_i = a$ во втором случае. Однако слова $a^{n+k}c$ и $a^{n+k}ca^ia$ также входят в язык, а их суффиксы (после p_n) не начинаются с v_n. Поэтому ни с какой позиции слова запомненный префикс v_n детерминированно прочитано быть не может.