ПРИКЛАДНАЯ ТЕОРИЯ АВТОМАТОВ

DOI 10.17223/20710410/12/4

УДК 519.713

О ПОГРЕШНОСТИ ПОЛИНОМИАЛЬНОГО ВЫЧИСЛЕНИЯ ОПТИМАЛЬНОЙ РАСКРАСКИ ГРАФА В СИНХРОНИЗИРУЕМЫЙ АВТОМАТ[1]

М. В. Берлинков

Уральский государственный университет, г. Екатеринбург, Россия

E-mail: berlm@mail.ru

Сильносвязный орграф называется допустимым, если исходящие степени всех вершин в нём одинаковы и длины циклов взаимно просты в совокупности. Раскраской допустимого графа G назовем произвольный автомат A, граф которого совпадает с G. Слово называется синхронизирующим автомат A, если оно переводит его в одно и то же состояние вне зависимости от исходного состояния автомата A. Оптимальная раскраска допустимого графа — это раскраска с кратчайшим синхронизирующим словом среди всех синхронизирующих раскрасок. Длину соответствующего синхронизирующего слова назовем значением оптимальной раскраски. Доказано, что любой приближенный полиномиальный алгоритм для вычисления оптимальной раскраски или ее значения имеет относительную погрешность не меньше 2 в случае трехбуквенного алфавита при предположении $\mathcal{P} \neq \mathcal{NP}$. Также показано, как результат можно перенести на случай двухбуквенного алфавита.

Ключевые слова: *проблема раскраски дорог, поиск оптимальной раскраски, кратчайшее синхронизирующее слово, синхронизируемые автоматы.*

Введение и история задачи

Введём определения синхронизируемого автомата и оптимальной раскраски. Пусть $A = \langle Q, \Sigma, \delta\rangle$ — полный детерминированный конечный автомат, где Q — множество состояний; Σ — входной алфавит автомата; $\delta : Q \times \Sigma \to Q$ — функция переходов. Функция δ продолжается единственным образом на множество $Q \times \Sigma^*$, где Σ^* обозначает свободный моноид над Σ. Таким образом, каждое слово из Σ^* действует на множестве Q в соответствии с функцией переходов δ. Через $S.v$ будем обозначать образ подмножества $S \subseteq Q$ под действием слова v, т. е. $S.v = \{\delta(q, v) : q \in S\}$.

Автомат A называется *синхронизируемым*, если существует слово $w \in \Sigma^*$, действие которого «перезагружает» автомат A, т. е. переводит автомат в некоторое состояние вне зависимости от исходного состояния автомата: $q.w = p.w$ для всех $q, p \in Q$, что эквивалентно равенству $|Q.w| = 1$. Произвольное слово w с таким свойством называется *синхронизирующим* автомат A, а длина кратчайшего синхронизирующего слова для A называется значением функцией Черни и обозначается через $\mathfrak{C}(A)$. На рис. 1 в центре нарисован автомат Черни C_4 с кратчайшим синхронизирующим словом ba^3ba^3b длины 9 и, следовательно, $\mathfrak{C}(C_4) = 9$.

[1]Работа выполнена при поддержке Федерального агентства по образованию России (грант №2.1.1/13995) и РФФИ (грант №10-01-00524).

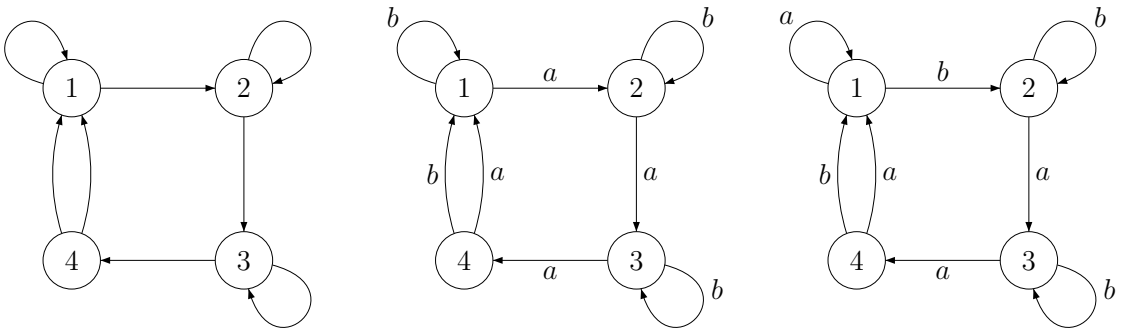

Рис. 1. Автомат Черни, его граф и оптимальная раскраска

Задачи, возникающие в теории синхронизируемых автоматов, являются не просто алгебраическими головоломками, но и имеют приложения в различных областях компьютерных наук, таких, как кодирование или построение вычислительных систем. Вычислительные системы зачастую могут быть смоделированы как конечные автоматы. Под действием *шума* или внутренних причин в системе могут происходить некорректные переходы. Системы, соответствующие синхронизируемым автоматам, более устойчивы к таким ошибкам, так как могут быть возвращены в корректное состояние применением перезагрузочной последовательности операций (синхронизирующего слова). Ясно, что важно не только наличие такой последовательности, но и то, насколько короткой ее можно выбрать. Таким образом, две ключевые задачи этой теории — это задача проверки заданного автомата на синхронизируемость и задача поиска кратчайшего синхронизирующего слова или его длины для заданного синхронизируемого автомата.

При моделировании систем конечными автоматами иногда встречаются такие, в которых определены возможные переходы между состояниями и набор допустимых операций, но не определено соответствие («раскраска») между операциями и переходами, либо соответствие задано, но его можно переопределить («перераскрасить»). При этом ключевые задачи остаются теми же, но ставятся с учетом возможности перераскраски: задача проверки возможности перераскраски в синхронизируемый автомат и задача поиска перераскраски с кратчайшим синхронизирующим словом среди всех синхронизируемых перераскрасок. Для перехода к этим задачам нам понадобится определение *графа автомата*.

Орграф $G = UG(A)$ называется *графом автомата* A, если его множество вершин совпадает с множеством состояний автомата A, а дуги соответствуют переходам автомата, т. е. из вершины q есть стрелка в вершину q' тогда и только тогда, когда $\delta(q, a) = q'$ для некоторой буквы $a \in \Sigma$. Тогда G — орграф с одинаковыми степенями исхода вершин, и автомат A называется *раскраской* графа G. Сильносвязный орграф G называется *допустимым*, если исходящие степени всех вершин в нём одинаковы и длины циклов взаимно просты в совокупности. Условие одинаковых исходящих степеней вершин необходимо для возможности раскраски, а условия взаимной простоты длин циклов и сильной связности необходимы для разрешимости и неприводимости задачи нахождения синхронизирующей раскраски [1 – 3]. Очевидно, что любой допустимый орграф может быть получен как граф подходящего автомата и любая перераскраска автомата — это раскраска его графа.

Пусть G — допустимый орграф. Раскраску A назовем *оптимальной раскраской* G и положим $\mathrm{OPT}(G) = \mathfrak{C}(A)$, если для любой другой раскраски B графа G выполняется $\mathfrak{C}(A) \leqslant \mathfrak{C}(B)$; в этом случае $\mathrm{OPT}(G)$ называется *значением оптимальной раскраски*. Другими словами, раскраска графа G оптимальна, если она есть раскраска с кратчайшим синхронизирующим словом среди всех возможных синхронизируемых раскрасок графа G. На рис. 1 показаны один четырёхвершинный орграф и две его синхронизируемые раскраски. В центре нарисован автомат Черни с кратчайшим синхронизирующим словом ba^3ba^3b длины 9, а справа — оптимальная раскраска заданного орграфа с синхронизирующим словом a^3 длины 3.

Вопрос проверки автомата на синхронизируемость полиномиально разрешим (см. для примера [4]). Гораздо более интересным является вопрос о том, можно ли перераскрасить любой автомат так, чтобы он стал синхронизируемым. Этот вопрос известен как *проблема раскраски дорог*. Задачу впервые поставили в 1970 г. Р. Л. Адлер и Б. Вэйс в работе [5], окончательно сформулировав её в следующем виде для допустимых графов в [1] (1977 г.).

Гипотеза. Любой допустимый граф имеет синхронизирующую раскраску.

Несмотря на многочисленные исследования, эта гипотеза оставалась открытой более 30 лет и была положительно решена А. Н. Трахтманом [2] только в 2008 г. Доказательство в [2] конструктивно и дает кубический (от числа состояний) алгоритм для нахождения синхронизирующей раскраски любого допустимого графа. В [6] было показано, как можно решать эту задачу квадратичным алгоритмом.

Рассмотрим теперь задачу минимизации длины синхронизирующего слова. Для автоматов ее можно поставить следующим образом.

Задача Min-Synch-Length. **Дан** синхронизируемый автомат A, **найти** $\mathfrak{C}(A)$.

Д. Эпштейн [7] показал, что эта задача NP- и co-NP-трудна, и, следовательно, даже недетерминированный полиномиальный алгоритм не может решать эту задачу в предположении $\mathcal{P} \neq \text{co-}\mathcal{NP}$. Из этого результата, однако, не следует, что не существует полиномиального алгоритма, находящего синхронизирующее слово на один символ длиннее кратчайшего. Тем не менее в [8] показано, что в предположении $\mathcal{P} \neq \mathcal{NP}$ не существует полиномиального алгоритма, решающего последнюю задачу даже для класса двухбуквенных автоматов.

Заметим, что для допустимых графов можно поставить две задачи, связанные с минимизацией длины синхронизирующего слова:

1) задачу вычисления значения оптимальной раскраски:

Задача OCV (Optimal Coloring Value). **Дан** допустимый граф G, **найти** $\mathrm{OPT}(G)$;

2) задачу построения оптимальной раскраски:

Задача OC (Optimal Coloring). **Дан** допустимый граф G, **найти** его оптимальную раскраску.

Заметим, что хотя задача Min-Synch-Length соответствует задаче OCV с зафиксированной раскраской, из её полиномиальной неаппроксимируемости не следует такой же результат для задачи вычисления значения оптимальной раскраски OCV, так как для нахождения $\mathrm{OPT}(G)$ не требуется искать значения $\mathfrak{C}(A(G))$ для всех раскрасок $A(G)$ графа G.

Далее докажем, что в предположении $\mathcal{P} \neq \mathcal{NP}$ любой приближенный полиномиальный алгоритм решения задачи OCV в трёхбуквенном алфавите имеет относительную погрешность не меньше 2, и покажем, как этот результат перенести на двухбуквенный случай и применить к задаче OC.

В п. 1 вводятся вспомогательные понятия и обозначения. В п. 2 доказано, что в предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм для вычисления значения оптимальной раскраски имеет относительную погрешность не меньше 2 в случае трехбуквенного алфавита, и показано, как перенести результат на задачу поиска оптимальной раскраски. В п. 3 показано, что при помощи некоторых изменений в структуре графа можно доказать такой же результат для раскраски в двухбуквенном алфавите. Из этого, в частности, следует, что задача поиска оптимальной раскраски NP-трудна даже в случае двухбуквенного алфавита.

1. Вспомогательные определения

Пусть A — сильносвязный синхронизируемый автомат, Q — множество его состояний. Назовем подмножество $S \subseteq Q$ *занятым* после применения некоторого слова v, если $S \subseteq Q.v$. Будем обозначать $v(i \dots j)$ подслово слова v с i-го по j-й символ включительно; $v(i)$ — i-й символ слова.

Пусть S, T — два непустых множества состояний автомата A. Через $LRadius_A(S, T)$ обозначим минимальное число r, такое, что найдется $t \in T$ и для каждого элемента s из S есть путь из s в t с длиной, в точности равной r. В роли множества T обычно будет выступать Q, поэтому для краткости обозначим $LRadius_A(S, Q)$ через $LRadius_A(S)$. Например, на рис. 1 для автомата Черни $LRadius_{C_4}(\{1, 2\}, \{2\}) = 1$. Через $LSynch_A(S)$ для непустого множества состояний S обозначим длину кратчайшего слова v, синхронизирующего множество S в автомате A, т. е. такого v, что $|S.v| = 1$. Заметим, что на рис. 1 для автомата Черни $LSynch_{C_4}(\{1, 2\}) = 4$, и $LSynch_A(Q)$ совпадает с $\mathfrak{C}(A)$ для любого автомата A.

Пусть r — натуральное число; тогда через $Image_A(S, r)$ (соответственно через $Preimage_A(S, r)$) обозначим множество состояний $q \in Q$, таких, что в $UG(A)$ есть путь длины не больше r из множества S в q (соответственно в множество S из q). Другими словами, $Image_A(S, r)$ (соответственно $Preimage_A(S, r)$) — это полный образ (соответственно прообраз) S под действием слов длины не более r в автомате A. Отметим несколько простых свойств введенных выше функций.

Лемма 1.

1) Функция $LRadius$ везде определена.
2) $LSynch_A(S) \geqslant LRadius_A(S)$.
3) Пусть T — объединение двух непустых множеств T_1 и T_2; тогда
$$LRadius_A(S, T) = \min(LRadius_A(S, T_1), LRadius_A(S, T_2)).$$
4) Пусть S_1 — непустое подмножество S; тогда $LRadius_A(S_1, T) \leqslant LRadius_A(S, T)$.
5) Функции $LRadius$, $Image$ и $Preimage$ зависят только от графа автомата.
6) Пусть $S \nsubseteq Preimage(T, r)$ для некоторого $r > 0$; тогда $LRadius_A(S, T) > r$.

Доказательство. Пусть w есть кратчайшее слово, синхронизирующее S в автомате A; тогда $S.w = q_0$ для некоторого $q_0 \in Q$. Так как автомат сильносвязный, то существует слово v, такое, что $q_0.v \in T$. Тогда $S.wv$ — это одноэлементное множество в T, поэтому $LRadius_A(S, T)$ определен и не превосходит $|wv|$. Таким образом, п. 1 леммы 1 доказан. Если в качестве T выбрать множество всех состояний, то слово v можно выбрать пустым. Следовательно, $LRadius_A(S) \leqslant |w| = LSynch_A(S)$, и п. 2 также доказан. Пункты 3–7 непосредственно следуют из определений. Действительно, проверим для примера п. 3. Заметим, что для функции $LRadius$ справедливо соотношение $LRadius_A(S, T) = \min\limits_{t \in T} LRadius_A(S, t)$. Следовательно, п. 3 следует из свойства минимума и условия $T = T_1 \cup T_2$. ■

Поскольку все автоматы, рассматриваемые далее в доказательстве теоремы, имеют один и тот же граф, то ввиду п. 5 леммы 1 можно не указывать автомат в записи функций $LRadius$, $Image$ и $Preimage$.

Используемые в работе определения из теории сложности вычислений можно найти в [9]. Приведем понятие аппроксимирующего алгоритма для нашей задачи. Пусть $\mathcal{K}$ — некоторый класс допустимых графов. Следуя определению погрешности алгоритма из [9], скажем, что алгоритм Alg *приближенно находит* значение оптимальной раскраски в классе $\mathcal{K}$, если для любого допустимого графа $G \in \mathcal{K}$ алгоритм возвращает натуральное число $Alg(G) \geqslant \mathrm{OPT}(G)$. Тогда говорят, что алгоритм Alg решает OCV с *относительной погрешностью* $k \in R$ для класса $\mathcal{K}$, если

$$\sup\left\{\frac{Alg(G)}{\mathrm{OPT}(G)} : G \in \mathcal{K}\right\} = k.$$

2. Случай трехбуквенного алфавита

Теорема 1. В предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм, решающий OCV для класса трехбуквенных автоматов, имеет относительную погрешность не меньше 2.

Доказательство. План доказательства следующий. Покажем, что полиномиальный алгоритм Alg, решающий OCV с относительной погрешностью меньше 2, может быть использован для полиномиального алгоритма, решающего NP-полную задачу ВЫПОЛНИМОСТЬ (SAT). Таким образом, получим противоречие с предположением $\mathcal{P} \neq \mathcal{NP}$. Более точно, пусть полиномиальный алгоритм Alg решает OCV с относительной погрешностью $2-\varepsilon$, где $0 < \varepsilon < 2$. Возьмем произвольный экземпляр задачи SAT — формулу ψ с n переменными и m дизъюнктами (дизъюнкциями переменных и их отрицаний). Построим допустимый граф $G(\psi)$, имеющий полиномиальный размер от m, n, и полином $p(m, n)$, такие, что:

— если ψ выполнима, то $\mathrm{OPT}(G(\psi)) \leqslant p(m, n)$ (этот случай назовем $GOODCASE$);
— если ψ невыполнима, то $\mathrm{OPT}(G(\psi)) \geqslant (2 - 0{,}5\varepsilon)p(m, n)$ (случай $BADCASE$).

Из построения следует, что ψ выполнима тогда и только тогда, когда Alg возвращает значение, меньшее чем $(2 - 0{,}5\varepsilon)p(m, n)$. Следовательно, найдем решение NP-полной задачи SAT, используя полиномиальный алгоритм Alg.

Выполним теперь описанное сведение формально. Рассмотрим некоторый экземпляр ψ задачи SAT с набором переменных $x_1, x_2, \ldots, x_n$ и набором дизъюнктов от них $c_1, c_2, \ldots, c_m$. Будем писать $x_i \in c_j$ или $\neg x_i \in c_j$, если дизъюнкт c_j содержит в своей записи переменную x_i или ее отрицание соответственно. Без ограничения общности можно предполагать, что $m > 3$, $n > 3$ и задача ψ нормализована, т. е. выполняются следующие условия:

(C1) Для любой переменной x_i есть дизъюнкт c_j, состоящий из переменной и ее отрицания, т. е. $c_j = (x_i \vee \neg x_i)$.
(C2) Переменная x_n равна 0 в любом наборе, выполняющем ψ.
(C3) Для любой переменной x_i и дизъюнкта c_j если ($x_i \in c_j$ и $\neg x_i \in c_j$), то $c_j = x_i \vee \neg x_i$.

Для того чтобы выполнялись условия (C1), (C3), нужно добавить дизъюнкты вида $c_j = (x_i \vee \neg x_i)$, если их еще нет в формуле, и исключить все дизъюнкты, содержащие x_i, $\neg x_i$ и еще какие-то переменные. Если (C2) не выполняется, то можно добавить переменную x_{n+1}, дизъюнкт $c_{m+1} = (\neg x_{n+1})$, а также дизъюнкт $c_{m+2} = (x_{n+1} \vee \neg x_{n+1})$,

чтобы удовлетворять $(C1)$. Ясно, что такие преобразования формулы ψ не влияют на ее выполнимость.

Вместо определения дуг графа $G(\psi)$ будем сразу определять переходы автомата A этого графа так, что если ψ выполнима, то $\mathfrak{C}(A) \leqslant p(m,n)$. Из этого следует корректность сведения для случая *GOODCASE*. Для случая *BADCASE*, т. е. когда ψ невыполнима, сделаем предположение от противного о том, что есть некоторая раскраска B графа $G(\psi)$, такая, что

$$\mathfrak{C}(B) < (2 - 0{,}5\varepsilon)p(m,n). \qquad (E_v)$$

Получив противоречие с этим предположением, докажем корректность сведения для случая *BADCASE*.

Таким образом, для случая *GOODCASE* достаточно показать, как можно за полиномиальное от n, m время построить автомат A графа $G(\psi)$, для которого есть синхронизирующее слово u длины не больше $p(m,n)$. Для *BADCASE* нужно получить противоречие с тем, что найдется слово v длины не больше $(2 - 0{,}5\varepsilon)p(m,n)$, синхронизирующее автомат B. Заметим также, что из доказательства будет видно, что граф $G(\psi)$ строится за полиномиальное от размера ψ время.

Схема автомата A показана на рис. 2. Множество состояний Q автомата A является объединением шести *компонент* (подмножеств состояний автомата, играющих определенную роль в доказательстве): R_{init}, R_{sat}, T_{sat}, R_{fix}, R_{acc}, Q_{spec}. Алфавит Σ автомата A состоит из трех букв a, b, c, а функция переходов будет определена по ходу доказательства. Заметим, что компонента Q_{spec} состоит из подмножеств C_0, C_1, C_2, R_0 и состояния z и не окружена рамкой. Компонента R_{acc} состоит из набора подмножеств $D_1, D_2, \ldots, D_{n-1}$ и подкомпоненты Row_{acc}, также явно на рисунке не обозначенной.

Компоненты R_{init} и T_{sat} являются трехмерными, а компоненты R_{sat} и Row_{acc} — двумерными массивами состояний, т. е. каждое состояние из этих компонент может быть записано как $K(i_1, i_2, i_3)$, где K — одна из компонент R_{init}, R_{sat}, T_{sat}, Row_{acc} (индекс i_3 имеет смысл только для R_{init} и T_{sat}). *Строкой* и *столбцом* этих компонент автомата будем называть подмножество состояний массива компоненты с фиксированным первым и вторым индексом соответственно. Например, на рис. 2 отмечены 1-я и 2-я строки R_{init} ($R_{init}(1,*,*)$ и $R_{init}(2,*,*)$); 1-я и $(n+1)$-я строки R_{sat} ($R_{sat}(1,*)$ и $R_{sat}(n+1,*)$); 4-й, 6-й и $(4n+2)$-й столбцы Row_{acc} ($Row_{acc}(*,4)$, $Row_{acc}(*,6)$, $Row_{acc}(*,4n+2)$) и др. Одномерные массивы состояний C_0, C_1, C_2, R_{fix} назовем столбцами, а R_0 — строкой. Заметим, что строкам и столбцам компонент на рис. 2 соответствуют горизонтальные и вертикальные наборы состояний соответственно.

Введем теперь полиномы, требующиеся для определения компонент:

$p_1 = p_1(m,n) = m + 3$ — число столбцов в R_{init} и R_{sat};

$p_2 = p_2(m,n) = 15n^2 + m$ — вспомогательный полином;

$p_3 = p_3(m,n) = 4(p_1 + p_2)$ — количество значений, которые может принимать третий индекс в R_{init};

$p = p(m,n) = ((kn-1)p_3+2)+(n+1)+(18+(4n+13)(n-2))$ — полином, участвующий в оценках на $\mathrm{OPT}(G)$;

$p_{add} = p_{add}(m,n) = p_3(kn-1) = 4(15n^2+2m+2)(kn-1)$ — высота столбцов C_0, C_1, C_2 компоненты Q_{spec}.

Константа k выбрана так, чтобы выполнялось неравенство $2p_{add} \geqslant (2 - 0{,}5\varepsilon)p$ для всех $m > 3$, $n > 3$. Заметим, что k не зависит от задачи, т. е. от ψ, m или n.

Перед тем как переходить к формальному определению компонент автомата и рассмотрению их свойств, объясним, для чего необходима каждая из компонент на соответствующем шаге доказательства.

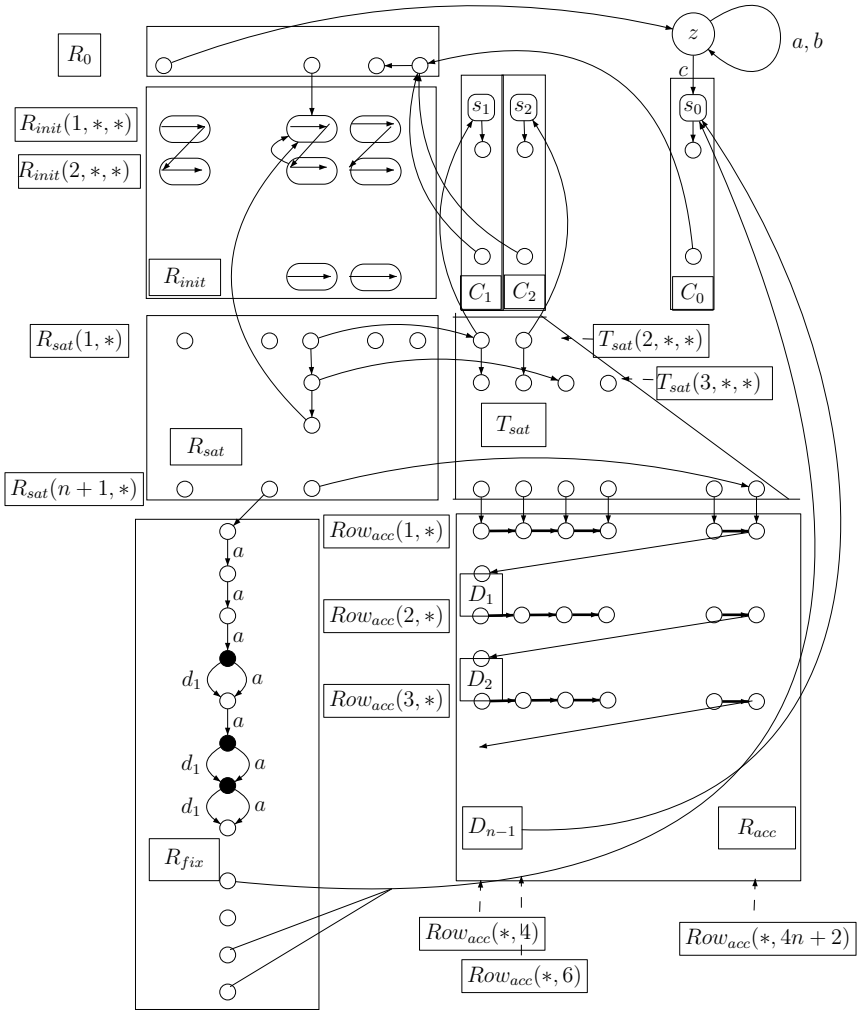

Рис. 2. Схема автомата A

1) При помощи свойств компоненты R_{init} построим слово u_{init} длины $(kn-1)p_3+$ $+2$ для автомата A, отображающее $Q \setminus Q_{spec}$ в первую строку R_{sat} $(R_{sat}(1,*))$. Таким образом, фактически после этого шага потребуется отслеживать образ только одной строки. Для случая *BADCASE* докажем, что можно найти префикс v_{init} слова v со свойством $R_{sat}(1,*) \subseteq R_{init}.v_{init}$ и длиной не меньше $(kn-1)p_3$.
2) Компоненты R_{sat} и T_{sat} построены для определения выполнимости ψ. В случае *GOODCASE* для автомата A найдется слово u_{sat} длины $n+1$, такое, что ровно n состояний заняты в строке $Row_{acc}(1,*)$ после применения этого слова к первой строке R_{sat}, полученной после шага 1. В случае *BADCASE* для автомата B докажем, что в $Row_{acc}(1,*)$ занято более n состояний после применения любого слова длины $n+1$ к первой строке R_{sat}, полученной после шага 1.
3) Компоненты R_{fix} и R_{acc} требуются для подсчета числа занятых состояний в строке $Row_{acc}(1,*)$ после предыдущего шага. Компонента R_{fix} нужна для «фиксации» (наложения ограничений на структуру) слова, используемого на этом шаге для *BADCASE*.

4) Компонента Q_{spec} играет специальную роль; в частности, с её помощью проверяется, что после предыдущего шага нет занятых состояний в $Q \setminus Q_{spec}$. Кроме того, компонента Q_{spec} помогает сделать граф сильносвязным.

Определим формально множества состояний компонент автомата A:

$$R_{init} = \{R_{init}(i_r, i_c, i_f) : i_r \in [1, kn], i_c \in [-2, m], i_f \in [0, p_3 - 1]\},$$

где i_r — номер строки R_{init}; i_c — номер столбца R_{init}, при $i_c > 0$ соответствующий номеру дизъюнкта; i_f — номер элемента в $R_{init}(i_r, i_c, *)$.

Следующие компоненты:

$$T_{sat} = \{T_{sat}(i_r, i_v, i_{vv}) : i_r \in [2, n+1], i_v \in [1, i_r], i_{vv} \in \{0, 1\}\},$$

где i_r — номер строки T_{sat}, при $i_r \leqslant n$ соответствующий номеру переменной плюс 1, а i_v (при $i_v \neq n+1$) и i_{vv} соответствуют номеру переменной и ее значению.

Компонента R_{sat} состоит из подкомпонент R^+_{sat} и R^-_{sat}, где

$$R^-_{sat} = \{R_{sat}(i_v, i_c) : i_v \in [1, n+1], i_c \in [-2, 0]\},$$
$$R^+_{sat} = \{R_{sat}(i_v, i_c) : i_v \in [1, n], i_c \in [1, m], i_v = \max(i \in [1, n] : x_i \in c_{i_c} \text{ или } \neg x_i \in c_{i_c})\}.$$

В этом определении i_v — это номер строки R_{sat}, при $i_v \neq n+1$ соответствующий номеру переменной, а i_c — номер столбца R_{sat}, при $i_c > 0$ соответствующий номеру дизъюнкта.

Пусть $h = (4n + 13)(n - 2) + 20$, тогда $R_{fix} = \{R_{fix}(i_r) : i_r \in [1, h]\}$, т. е. компонента R_{fix} состоит из одного столбца высоты h.

Компонента R_{acc} состоит из $n - 1$ строки $Row_{acc}(i, *)$, и любой путь в R_{acc} между двумя последовательными строками будет проходить через подкомпоненту D_i. Формально:

$$Row_{acc} = \{Row_{acc}(i_v, i_c) : i_v \in [1, n-1], i_c \in [4, 4n+2]\},$$
$$\forall i_v \in [1, n-1] \quad D_{i_v} = \{D_{i_v}(i_e) : i_e \in [1, 35]\},$$

где i_v — номер строки Row_{acc}, i_c — номер столбца Row_{acc} и i_e — номер элемента в D_i. Для $i \in [1, n-2]$ положим $D_i(35) = Row_{acc}(i+1, 4)$,

$$R_{acc} = Row_{acc} \cup D_1 \cup D_2 \cup \ldots \cup D_{n-1},$$

$h_s = p_{add}$ и определим Q_{spec} следующим образом: $Q_{spec} = \{z\} \cup R_0 \cup C_0 \cup C_1 \cup C_2$, где $C_i = \{C_i(i_r) : i_r \in [1, h_s]\}$ для $i \in \{0, 1, 2\}$ — это столбец Q_{spec} из h_s элементов и $R_0 = \{R_0(i_c) : i_c \in [-2, m]\}$ — строка Q_{spec} из $m + 3$ элементов. Для i из $\{0, 1, 2\}$ обозначим $s_i = C_i(1)$.

При определении переходов в автомате A будем использовать метки LD_XY, где X — метка компоненты, а Y — порядковый номер метки. Каждое такое определение переходов в автомате A однозначно задает определение дуг графа $G(\psi)$, которое будем обозначать, заменяя первую букву L на E (например, $LD_{init}1$ и $ED_{init}1$). Будем говорить, что слово w перемещает/отображает состояние q в состояние q', если $q.w = q'$, и слово w «сжимает», или «синхронизирует» множество S, если $|S.w| = 1$. Термины направления перемещения следует понимать в соответствии с рис. 2, и они будут использоваться только для визуального представления.

Переходы из состояний R_{init} в автомате определены следующим образом: буква a перемещает состояния R_{init} внутри столбцов по «спирали» относительно номера строки и третьего индекса. Формально:

$$R_{init}(i_r, i_c, i_f).a = \begin{cases} R_{init}(i_r, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 < p_3, & LD_{init}1 \\ R_{init}(i_r+1, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 = p_3 \text{ и } i_r < kn, & LD_{init}2 \\ R_{init}(1, i_c, (i_f+1) \bmod p_3), \text{ если } i_f + 1 = p_3 \text{ и } i_r = kn. & LD_{init}3 \end{cases}$$

Буква b отображает состояния последней строки R_{init} в первую строку R_{sat} (т. е. $R_{init}(kn, *, *).b \subseteq R_{sat}(1, *)$). Более того, состояния нулевого столбца последней строки компоненты R_{init} отображаются на первую строку R_{sat} (т. е. $R_{init}(kn, 0, *).b = R_{sat}(1, *)$). Другие строки перемещаются в первую строку R_{init} по этой букве:

$$R_{init}(i_r, i_c, i_f).b = \begin{cases} R_{init}(1, i_c, (i_f+1) \bmod p_3), \text{ если } i_r < kn, & LD_{init}4 \\ R_{sat}(1, i_c), \text{ если } i_r = kn, i_c \neq 0, & LD_{init}5 \\ R_{sat}(1, ((i_f+1) \bmod p_1) - 1), \text{если } i_r = kn, i_c = 0. & LD_{init}6 \end{cases}$$

Буква c перемещает все состояния R_{init} вверх, в первую строку R_{init}:

$$R_{init}(i_r, i_c, i_f).c = R_{init}(1, i_c, (i_f+1) \bmod p_3). \qquad LD_{init}7$$

Следующее замечание следует из определений дуг в R_{init}.

Замечание 1.

(A) Все буквы в автоматах A и B сохраняют «полное» множество остатков от деления третьего индекса в R_{init} на p_3, пока состояния находятся в R_{init}, т. е. если для $x \in \Sigma$, $S \subseteq R_{init}$ выполняется $S.x \subseteq R_{init}$ и $\forall i \in [0, p_3 - 1] \exists q \in S \cap R_{init}(*, *, i)$, то также выполняется $\forall i \in [0, p_3 - 1] \exists q \in S.x \cap R_{init}(*, *, i)$.

(B) $Image(R_{init}(1, *, 0), p_{add}) \subseteq R_{init}$; другими словами, кратчайший путь, ведущий из $R_{init}(1, *, 0)$ вне R_{init}, имеет длину больше $p_{add} = p_3(kn - 1)$.

Рассмотрим теперь компоненты R_{sat} и T_{sat}, которые «кодируют» ψ. Заметим, что образ множества $\{-2, -1, 0\}$ под действием функции $\varphi(x) = -0{,}5x^{2\smile}2{,}5x + 1$ равен $\{1, 3, 4\}$, и определим:

$$R_{sat}(i_v, i_c).x = \begin{cases} R_{init}(1, i_c, 0), \text{ если } x = c, & LD_{sat}1 \\ R_{fix}(\varphi(i_c)), \text{ если } x \in \{a, b\}, i_c \leqslant 0, i_v = n+1, & LD_{sat}2 \\ R_{sat}(i_v + 1, i_c), \text{ если } x \in \{a, b\}, i_c \leqslant 0, i_v \leqslant n. & LD_{sat}3 \end{cases}$$

Для $i_v \leqslant n$ и $i_c > 0$:

$$R_{sat}(i_v, i_c).x = \begin{cases} T_{sat}(i_v + 1, i_v, 0), \text{ если } x = a \text{ и } \neg x_{i_v} \in c_{i_c}, & LD_{sat}4 \\ T_{sat}(i_v + 1, i_v, 1), \text{ если } x = b \text{ и } x_{i_v} \in c_{i_c}, & LD_{sat}5 \\ R_{sat}(i_v + 1, i_c), \text{ если } x = a \text{ и не выполняется } (\neg x_{i_v} \in c_{i_c}), & LD_{sat}6 \\ R_{sat}(i_v + 1, i_c), \text{ если } x = b \text{ и не выполняется } (x_{i_v} \in c_{i_c}); & LD_{sat}7 \end{cases}$$

$$T_{sat}(i_r, i_v, i_{vv}).x = \begin{cases} s_{i_{vv}+1}, \text{ если } x = c, & LD_{sat}8 \\ T_{sat}(i_r + 1, i_v, i_{vv}), \text{ если } x \in \{a, b\}, i_r \leqslant n, & LD_{sat}9 \\ Row_{acc}(1, 2(2i_v + i_{vv})), \text{ если } x \in \{a, b\}, i_r = n + 1. & LD_{sat}10 \end{cases}$$

Будем писать $S \mapsto^r T$, если любой путь из подмножества S длины не меньше r содержит вершины из подмножества T. Заметим, что

$$\text{если} \quad S \mapsto^{r_1} T_1 \mapsto^{r_2} T_2, \ \text{ то} \quad S \mapsto^{r_1+r_2} T_2. \tag{1}$$

Следующее замечание непосредственно следует из определений дуг в R_{sat} и T_{sat} и свойств введенного отношения.

Замечание 2. Пусть $i_c \in [-2, m]$; тогда

(A) Если $i_c \leqslant 0$, то $R_{sat}(*, i_c) \mapsto^{n+1} R_{init}(1, i_c, 0) \cup R_{fix}$ (по $ED_{sat}1, 2, 3$).

(B) Если $i_c > 0$, то $R_{sat}(*, i_c) \mapsto^{n+1} R_{init}(1, i_c, 0) \cup \{s_1, s_2\} \cup R_{acc}$ (по $ED_{sat}1, 4, \ldots, 7$).

Введем линейный порядок между состояниями, находящимися в одной строке Row_{acc} в соответствии с номерами столбцов компоненты Row_{acc}. Пусть S — некоторое множество. Назовем две вершины из S, лежащие в одной строке Row_{acc}, *последовательными*, если между ними (в соответствии с порядком) нет других вершин из S. Назовем *d-переходами* в графе $G(\psi)$ переходы, помеченные буквой $d \in \Sigma$ в автомате A. Следующая лемма показывает, как компоненты R_{sat} и T_{sat} кодируют формулу ψ.

Лемма 2.

GOODCASE: Существует слово u_{sat} длины $n+1$, такое, что $R_{sat}(1, *).u_{sat} \cap R_{fix} = = R_{fix}(\{1, 3, 4\})$, и если положить $S = R_{sat}(1, *).u_{sat} \cap R_{acc}$, то для S верны следующие свойства:

(G0) $S \subseteq Row_{acc}(1, *)$, т. е. S полностью содержится в первой строке;

(G1) номера столбцов состояний из S четны;

(G2) $|S| = n$;

(G3) расстояние между любыми двумя последовательными состояниями из S в строке равно $2, 4$ или 6;

(G4) $Row_{acc}(1, 4n) \in S$, $Row_{acc}(1, 4n + 2) \notin S$.

BADCASE: Пусть при применении слова v_{sat} длины $n+1$ к $R_{sat}(1, *)$ используются только a- и b-переходы; тогда $R_{sat}(1, *).v_{sat} \cap R_{fix} = R_{fix}(\{1, 3, 4\})$, и если положить $S = R_{sat}(1, *).v_{sat} \cap R_{acc}$, то для S верны следующие свойства:

(B0) $S \subseteq Row_{acc}(1, *)$, т. е. S полностью содержится в первой строке;

(B1) номера столбцов состояний из S четны;

(B2) $|S| > n$.

Доказательство. Рассмотрим сначала ситуацию *GOODCASE*. Так как ψ выполнима, то существует набор значений переменных $x_1 = b_1$, $x_2 = b_2$, …, $x_n = b_n$, такой, что $\psi(b_1, b_2, \ldots, b_n)$ истинна, или, другими словами, существует минимальный номер переменной $j = j(i)$ для каждого дизъюнкта c_i, такой, что либо $b_j = 1$, $x_j \in c_i$, либо $b_j = 0$, $\neg x_j \in c_i$. Сопоставим слово u_{sat} этому набору значений переменных следующим образом. Положим $u_{sat}(n + 1) = a$; $u_{sat}(i) = a$, если $b_i = 0$, и $u_{sat}(i) = b$ в противном случае.

Из определений $LD_{sat}2, 3$ и условия $u_{sat} \in \{a, b\}^{n+1}$ следует, что $R^-_{sat}(1, *).u_{sat} = = R_{fix}(\{1, 3, 4\})$. Осталось доказать, что выполняются свойства множества $S = = R_{sat}(1, *).u_{sat} \cap R_{acc}$. Для этого рассмотрим образ некоторого состояния $q = R_{sat}(1, i_c)$ под действием u_{sat}, где $i_c > 0$:

$$\begin{aligned} q.u_{sat} &= R_{sat}(1, i_c).u_{sat}(1 \ldots j(i_c) \ldots n + 1) = \\ &= T_{sat}(j(i_c) + 1, j(i_c), b_{j(i_c)}).u_{sat}(j(i_c) + 1 \ldots n + 1) = \\ &= T_{sat}(j(i_c) + (n + 1) - j(i_c), j(i_c), b_{j(i_c)}).u_{sat}(n + 1) = \\ &= T_{sat}(n + 1, j(i_c), b_{j(i_c)}).u_{sat}(n + 1) = Row_{acc}(1, 2(2j(i_c) + b_{j(i_c)})). \end{aligned} \tag{EQ_1}$$

Из условия нормализации (C1) и равенства $R_{sat}(1,i_c).u_{sat} = R_{acc}(1, 2(2j(i_c)+b_{j(i_c)}))$ следует свойство (G2), потому что $\{j(i_c) : i_c \in [1,m]\} = [1,n]$. Свойства (G1), (G3) выполняются, потому что разность между номерами столбцов $2(2j+b_j)$ и $2(2(j+1)+$ $+b_{j+1})$ четна и меньше либо равна 6 и $\{j(i_c) : i_c \in [1,m]\} = [1,n]$. Свойство (G0) выполняется, так как $Row_{acc}(1, 2(2j(i_c)+b_{j(i_c)})) \in Row_{acc}(1,*)$, а свойство (G4) следует из равенства (EQ_1) и того, что $b_n = 0$ по (C2). Итак, случай *GOODCASE* доказан.

Рассмотрим теперь *BADCASE*. Так как при применении слова v_{sat} к $R_{sat}(1,*)$ нет использования c-переходов, то по определениям $ED_{sat}2,\ldots,7,10$ получаем, что

$$R^-_{sat}(1,*).v_{sat} = R_{fix}(\{1,3,4\}) \quad \text{и} \quad R^+_{sat}(1,*).v_{sat} \subseteq R_{acc}(1,*).$$

Осталось доказать свойства (B1) и (B2) для $S = R_{sat}(1,*).v_{sat} \cap R_{acc}$. В силу (C1) для любого $j \in [1,n]$ существует номер $i_c = i_{c(j)}$, такой, что $c_{i_c} = (x_j \vee \neg x_j)$. Следовательно,

$$\begin{aligned} R_{sat}(1,i_{c(j)}).v_{sat} &= R_{sat}(j,i_{c(j)}).v_{sat}(j\ldots n+1) = \\ &= T_{sat}(j+1,j,a_1).v_{sat}(j+1\ldots n+1) = \\ &= T_{sat}(n+1,j,a_1).v_{sat}(n+1) = \\ &= R_{acc}(1,2(2j+a_1)),\ \text{где } a_1 \in \{0,1\}, j \in [1,n]. \end{aligned} \qquad (EQ_2)$$

Из этого равенства следуют (B1) и неравенство $|R_{sat}(1,*).v_{sat} \cap R_{acc}(1,*)| \geqslant n$. Чтобы показать, что неравенство строгое, ввиду равенства (EQ_2) достаточно доказать, что существует номер $j \in [1,n]$, такой, что

$$R_{acc}(1,2(2j)) \in R_{sat}(1,*).v_{sat} \quad \text{и} \quad R_{acc}(1,2(2j+1)) \in R_{sat}(1,*).v_{sat}. \qquad (EQ_3)$$

Так как $R^+_{sat}(1,i).v_{sat} \subseteq R_{acc}(1,*)$, то для любого номера дизъюнкта $i_c \in [1,m]$ существует номер переменной $i_v(i_c) \in [1,n]$, такой, что

$$R_{sat}(1,i_c).v_{sat}(1\ldots i_v(i_c)) = R_{sat}(i_v(i_c),i_c).v_{sat}(i_v(i_c)) = T_{sat}(i_v(i_c)+1, i_v(i_c), i_{vv}(i_c)).$$

Предположим, что для всех $i_{c_1}, i_{c_2} \in [1,m]$ равенство номеров переменных $i_v(i_{c_1}) =$ $= i_v(i_{c_2})$ влечет равенство значений переменных $i_{vv}(i_{c_1}) = i_{vv}(i_{c_2})$. Для $j \in [1,n]$ положим $b_j = 1$, если $i_v(i_c) = j$, $i_{vv}(i_c) = 1$, и $b_j = 0$ в противном случае; тогда $\psi(b_1,b_2,\ldots,b_n) = 1$. Это верно, потому что для любого номера дизъюнкта i_c выполняется $b_{i_v(i_c)} = i_{vv}(i_c)$. Следовательно, предположение противоречит тому, что ψ невыполнима в *BADCASE*. Поэтому существуют некоторые числа j, i_{c_1}, i_{c_2}, такие, что $i_v(i_{c_1}) = i_v(i_{c_2}) = j$, $i_{vv}(i_{c_1}) = 0$, $i_{vv}(i_{c_2}) = 1$. Из этих равенств и определений дуг $ED_{sat}4,5,9$ непосредственно следует (EQ_3), а следовательно, и доказательство леммы 2 в случае *BADCASE*. ■

Перейдем теперь к определению переходов в компонентах R_{fix} и R_{acc}. Дуги D_i одинаковы для всех i и изображены на рис. 3. Символ E на рисунке обозначает алфавит $\{a,b\}$, а метки E^k на дугах заменяют $k-1$ промежуточных вершин и соответствующих переходов по a,b между начальной и конечной вершинами соответствующей дуги. Символы $e_1(i)$ и $e_2(i)$ могут быть одной из букв a,b и будут определены в процессе доказательства.

Переходы для R_{fix} и R_{acc} удобно представить, используя длины некоторых префиксов слова, фиксируемого при помощи компоненты R_{fix} на шаге 3. По определению положим $h_1 = 3$, $h_2 = h_3 = \ldots = h_{n-1} = 12+4n$, $h_n = 13$ и обозначим

$w_i = w_i(d_1, d_2, \ldots, d_i) = a^3 d_1 a^{h_2} d_2 \ldots d_{i-1} a^{h_i} d_i$ для $i \in [1, n]$ и некоторых букв d_1, d_2, …, d_n. Заметим, что высота h столбца R_{fix} равна $|w_n| + 3$.

Обозначим $R^2_{fix} = \{R_{fix}(\{|w_i|, |w_i| + 2, |w_i| + 3\}) : i \in [1, n-1]\}$ и определим

$$q.x = \begin{cases} R_{init}(1,0,0), \text{ если } x = c \text{ и } q \in R_{fix} \setminus R_{fix}(\{h-3, h-1, h\}), & LD_{acc}1 \\ R_{fix}(i_r + 1), \text{ если } q = R_{fix}(i_r), i_r < h \text{ и} & \\ \qquad (x = a \text{ или } (x = b \text{ и } q \in R^2_{fix})), & LD_{acc}2 \\ R_{init}(1,0,0), \text{ если } x = b \text{ и } q \in R_{fix} \setminus R^2_{fix}, & LD_{acc}3 \\ s_0, \text{ если } x = c \text{ и } q \in R_{fix}(\{h-3, h-1, h\}) \cup D_{n-1}(35), & LD_{acc}4 \\ s_1, \text{ если } x = c \text{ и } q \in R_{acc} \setminus D_{n-1}(35), & LD_{acc}5 \\ Row_{acc}(i_v, i_c + 1), \text{ если } q = Row_{acc}(i_v, i_c), i_c < 4n + 2 \text{ и } x \neq c, & LD_{acc}6 \\ D_{i_v}(1), \text{ если } q = Row_{acc}(i_v, 4n + 2) \text{ и } x \neq c. & LD_{acc}7 \end{cases}$$

Переходы в D_i по a, b (кроме помеченных $e_1(i)$, $e_2(i)$, $E - e_1(i)$, $E - e_2(i)$) определяются по рис. 3. Для возможности сослаться на определения дуг в D_i будем использовать метку $ED_{acc}8$. Следующее замечание говорит о назначении компоненты R_{fix}.

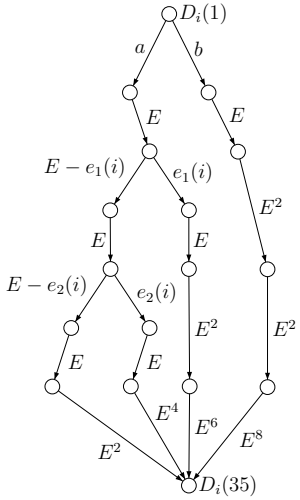

Рис. 3. Подкомпонента D_i

Замечание 3.

GOODCASE: Пусть $w = w_n(b, b, \ldots, b, c)$; тогда w имеет следующие свойства: $w(1) = a$, $R_{fix}(\{1,3,4\}).w = s_0$ и для любого собственного префикса w' слова w справедливо $R_{fix}(\{1,3,4\}).w' \subseteq R_{fix}$.

BADCASE: Пусть слово w обладает свойствами, указанными в утверждении для *GOODCASE*; тогда $w = w_n(d_1, d_2, \ldots, d_n)$ для некоторых $d_1, d_2, \ldots, d_n \in \Sigma$.

Доказательство. В случае *GOODCASE* докажем по индукции для $i \in [0, |w|-1]$, что $R_{fix}(\{1,3,4\}).w(1 \ldots i) = R_{fix}(\{i+1, i+3, i+4\})$. База индукции для $i = 0$ очевидна. Докажем шаг индукции для i. Если $w(i) = a$, то по $LD_{acc}2$ получаем $R_{fix}(\{i, i+2, i+3\}).a = R_{fix}(\{i+1, i+3, i+4\})$. В противном случае $w(i) = b$; тогда $i = |w_j|$ для некоторого j, поэтому $R_{fix}(\{i, i+2, i+3\}) \subseteq R^2_{fix}$, и по $LD_{acc}2$ снова получаем

$R_{fix}(\{i, i+2, i+3\}).b = R_{fix}(\{i+1, i+3, i+4\})$. Следовательно, шаг индукции доказан. По заключению индукции получаем, что

$$R_{fix}(\{1,3,4\}).w = R_{fix}(\{1,3,4\}).w(1\ldots|w|-1)c = R_{fix}(\{h-3, h-1, h\}).c = s_0$$

и при применении слова $w(1\ldots|w|-1)$ состояния из $R_{fix}(\{1,3,4\})$ перемещаются внутри R_{fix}.

Рассмотрим теперь случай *BADCASE*. Так как только состояния из $R_{fix}(\{h-3, h-1, h\})$ могут перейти в s_0 из R_{fix} и по условию $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) \subseteq R_{fix}$, то $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) \subseteq R_{fix}(\{h-3, h-1, h\})$. Все переходы внутри R_{fix} (по $ED_{acc}2$) увеличивают на 1 индекс состояния в столбце R_{fix}, поэтому $R_{fix}(\{1,3,4\}).w(1\ldots|w|-1) = R_{fix}(\{h-3, h-1, h\})$ и для любого $i \in [0, h-4]$ и $j \in \{1,3,4\}$ должно выполняться $R_{fix}(j).w(1\ldots i) = R_{fix}(i+j)$ и $|w| = |w_n| = h-3$.

Покажем индукцией по $k < |w_n|$, что если для $i \in [1, k-1]$ слово $w(1\ldots k-1)$ является префиксом w_n для некоторых $d_1, d_2, \ldots, d_n$, то и $w(1\ldots k)$ — также префикс w_n. База индукции очевидна, так как по условию $w(1) = a$.

Если для некоторого $k \in [3, h-2]$ из состояния $R_{fix}(3).w(1\ldots k-3) = R_{fix}(k)$ есть только один переход внутри R_{fix}, то $w(k) = w(k-2)$, так как $R_{fix}(k)$ также равно $R_{fix}(1).w(1\ldots k-1)$ (т. е. $R_{fix}(1)$ также проходит через эту вершину под действием перехода $w(k)$ и должно остаться в R_{fix}). Аналогично, если для $k \in [3, h-3]$ из состояния $R_{fix}(4).w(1\ldots k-4) = R_{fix}(k)$ есть только один переход внутри R_{fix}, то $w(k) = w(k-3)$. Следовательно, для всех $k \in [2, h-2]$, таких, что $R_{fix}(k) \notin R^2_{fix}$, выполняется $w(k) = w(k-2)$, и для всех $k \in [4, h-3]$, таких, что $R_{fix}(k) \notin R^2_{fix}$, выполняется $w(k) = w(k-3)$.

Заметим, что для $k \in \{2,3\}$ утверждение следует из соотношений $a = w(1) = w(3)$, $w(2) = w(3) = w(5)$. Пусть $k > 3$; если k совпадает с длиной одного из w_i, то шаг индукции выполнен для $d_i = w(k)$. В противном случае, так как $k > 3$ и длины w_i отличаются более чем на 1, то или $k-2$, или $k-3$ не совпадает с длиной ни одного из w_i и, следовательно, выполняется одно из соотношений $w(k) = w(k-2) = a$ или $w(k) = w(k-3) = a$, что влечет шаг индукции. Заключение индукции дает необходимый результат. ■

Отметим также несколько несложных свойств компонент R_{fix}, R_{acc}.

Замечание 4.

(A) $R_{fix} \mapsto^h \{R_{init}(1,0,0), s_0\}$ (по $ED_{acc}1\ldots4$).
(B) $Row_{acc}(k,*) \mapsto^{4n-2} Row_{acc}(k, 4n+2)$ (по $ED_{acc}5, 6$).
(C) $Row_{acc}(k, 4n+2) \mapsto^{15} \{D_k(35), s_1\}$ (по $ED_{acc}7, 8$).
(D) $Row_{acc}(1,*) \mapsto^{h+4n-7} \{s_0, s_1\}$ (по B, C и равенству $(4n-2)(n-1) + 15(n-1) = h + 4n - 7$).
(E) $R_{sat}(1, i_c) \mapsto^{h+5n-6} \{R_{init}(1,0,0), R_{init}(i_c,0,0), s_0, s_1, s_2\}$ (по свойству $\mapsto$, пунктам A, B замечания 2 и пунктам A, D этого замечания).
(F) $Row_{acc}(k, 4n+2) \mapsto^{4n+13} \{Row_{acc}(k+1, 4n+2), s_1\}$ (по B и C).
(G) $Row_{acc}(k, 4n+2) \mapsto^{(n-k-1)(4n+13)+15} \{s_0, s_1\}$ (по F и C).

В замечании 3 показано, как при помощи R_{fix} фиксируется шаблон слова. Далее отметим, как могут перемещаться состояния в R_{acc} под действием таких слов в обоих случаях.

Замечание 5. Пусть S — такое подмножество состояний, что для $k = 1$ выполняются следующие свойства.

GOODCASE:

(G0) $S \subseteq Row_{acc}(k, *)$, т. е. S полностью содержится в k-й строке;
(G1) номера столбцов состояний из S четны;
(G2) $|S| \leqslant n - k + 1$;
(G3) расстояние между любыми двумя последовательными состояниями из S в строке равно $2, 4$ или 6;
(G4) $Row_{acc}(k, 4n) \in S, Row_{acc}(k, 4n + 2) \notin S$.

Тогда существует дораскраска компоненты R_{acc} (т. е. определение $e_1(i), e_2(i)$), такая, что $S.w_n(b, b, \ldots, b, c) = s_0$.

BADCASE:

(B0) $S \subseteq Row_{acc}(k, *)$, т. е. S полностью содержится в k-й строке;
(B1) номера столбцов состояний из S четны;
(B2) $|S| > n - k + 1$.

Тогда для всяких таких $d_1, d_2, \ldots, d_n \in \Sigma$, что любой собственный префикс w' слова $w_n(d_1, d_2, \ldots, d_n)$ оставляет S внутри R_{acc} (т. е. $S.w' \subseteq R_{acc}$), выполняется $S.w_n(d_1, d_2, \ldots, d_n) \neq s_0$.

Доказательство. Рассмотрим случай *GOODCASE*.

Обозначим $w = w_n(b, b, \ldots, b, c)$ и докажем индукцией по $k \in [1, n]$ утверждение о том, что $D_1, D_2, \ldots, D_{k-1}$ можно раскрасить так, что для множества $T_k = S.w(1 \ldots |w_k| - 4)$ выполнены свойства (G0) – (G4) для k. База индукции следует из условий замечания. Докажем шаг индукции для $k + 1$. Пусть t обозначает разность номеров столбцов между максимальным (равным $Row_{acc}(k, 4n)$ по (G4)) и предшествующим состоянием $Row_{acc}(k, 4n - t)$ из T_k в k-й строке. Раскрасим D_k в зависимости от значения t так, чтобы путь, помеченный a^{14-t}, помечал путь из $D_k(1)$ в $D_k(35)$. Это можно сделать, положив

$$\begin{aligned} &e_1(k) = a, \quad e_2(k) = a, \quad \text{если} \quad t = 2; \\ &e_1(k) = b, \quad e_2(k) = a, \quad \text{если} \quad t = 4; \\ &e_1(k) = b, \quad e_2(k) = b, \quad \text{если} \quad t = 6. \end{aligned}$$

Тогда под действием слова $w(|w_k| - 3 \ldots |w_{k+1}| - 4) = a^3 b a^{9+4n}$ (при $k < n - 1$) состояние $Row_{acc}(k, 4n)$ будет двигаться по пути

$$\begin{aligned} &Row_{acc}(k, 4n) \to Row_{acc}(k, 4n + 1) \to Row_{acc}(k, 4n + 2) \to \\ &\to D_k(1) \to \ldots \to D_k(35) = Row_{acc}(k + 1, 4) \to Row_{acc}(k + 1, 4n), \end{aligned}$$

где путь из $D_k(1)$ в $D_k(35)$ помечен ba^{13}, а все остальные состояния из $Row_{acc}(k, i) \in T_k$ будут двигаться по пути

$$\begin{aligned} &Row_{acc}(k, i) \to Row_{acc}(k, i + 1) \to \ldots \to D_k(1) \to \\ &\to \ldots \to D_k(35) = Row_{acc}(k + 1, 4) \to Row_{acc}(k + 1, i + t), \end{aligned}$$

где путь из $D_k(1)$ в $D_k(35)$ помечен a^{14-t}. Таким образом, оба состояния $Row_{acc}(k, 4n - t)$ и $Row_{acc}(k, 4n)$ перейдут в состояние $Row_{acc}(k + 1, 4n)$, а разность номеров столбцов между остальными состояниями не изменится. Из этого следует, что все свойства (G0) – (G4) выполнены для $k + 1$ и множества $T_{k+1} = S.w(1 \ldots |w_{k+1}| - 4)$. Заметим, что для случая $k = n - 1$ доказательство такое же, но рассматриваются пути только до

$D_k(35)$. В конечном итоге получаем, что $S.w_n(b,b,\ldots,b,c) = D_{n-1}(35).c = s_0$, и случай *GOODCASE* доказан.

Докажем *BADCASE* аналогичным способом, добавив к утверждению свойство (B5): $Row_{acc}(k, 4n{+}2) \notin S$. Обозначим $w = w_n(d_1, d_2, \ldots, d_n)$ и докажем индукцией по $k \in [1, n]$ свойства (B0, B1, B2, B5) для $T_k = S.w(1 \ldots |w_k| - 4)$. База индукции снова следует из условий, кроме свойства (B5), которое докажем на шаге индукции без использования предположения. Докажем шаг индукции для $k+1$. Рассмотрим, как перемещаются вершины из T_k под действием слова $f = w(|w_k| - 3 \ldots |w_{k+1}| - 4) = a^3 d_k a^{9+4n}$. Докажем (B0) и (B5). Так как нет переходов из $Row_{acc}(k+1, *)$ в $Row_{acc}(k, *) \cup D_k$, то ввиду свойств B, C замечания 4 получаем, что $T_{k+1} \subseteq Row_{acc}([k{+}1, n{-}1], *) \cup D_{k+1} \ldots \cup D_{n-1}$. Пусть некоторое состояние $q \in T_{k+1}$ совпадает с $Row_{acc}(k{+}1, 4n{+}2)$ или не содержится в $Row_{acc}(k{+}1, *)$. Ввиду условия G замечания 4 максимальный путь из q внутри R_{acc} имеет длину меньше $(n-k-2)(4n+13)+15$. Однако по условиям путь из q под действием слова $w(|w_{k+1}| - 3 \ldots |w| - 1)$ длины $(n-k-2)(4n+13)+17$ должен лежать в R_{acc}. Получили противоречие, и свойства (B0) и (B5) доказаны. Заметим, что для доказательства (B5) не использовалось предположение индукции. Следовательно, оно выполнено и для базы индукции.

Так как только на 4-й позиции слова f стоит буква, отличная от a, и состояние $Row_{acc}(k, 4n+2) \notin T_k$, то по (B1) $Row_{acc}(k, 4n+1) \notin T_k$, поэтому длина кратчайшего пути из T_k в состояние $D_k(1)$ не меньше 4. Следовательно, все состояния из T_k, кроме, быть может, одного ($Row_{acc}(k, 4n+2)$), перемещаются по пути, помеченному $a^{|f|}$ (так как по $ED_{acc}6$ для состояний из Row_{acc} a- и b-переходы совпадают). Поскольку этот путь имеет четную длину и не может содержать циклов, получается, что разница номеров столбцов состояний из T_{k+1} оставшихся состояний такая же, как и у прообразов из T_k. Отсюда следуют свойства (B1) и (B2). ■

Рассмотрим компоненту Q_{spec}. Пусть $q \in Q_{spec}$ и $x \in \Sigma$. Тогда

$$q.x = \begin{cases} R_{init}(1, i_c, 0), \text{ если } q = R_0(i_c) \text{ и } x = c, & LD_{spec}1 \\ R_0(i_c - 1), \text{ если } q = R_0(i_c) \text{ и } i_c > 1, x \neq c, & LD_{spec}2 \\ z, \text{ если } q = R_0(1) \text{ и } i_c > 1, x \neq c, & LD_{spec}3 \\ s_0, \text{ если } q = z \text{ и } x = c, & LD_{spec}4 \\ C_i(i_r + 1), \text{ если } q = C_i(i_r), i_r < h_s, & LD_{spec}5 \\ R_0(m), \text{ если } q = C_i(h_s). & LD_{spec}6 \end{cases}$$

Заметим, что граф $G(\psi)$ теперь полностью определен, хотя раскраска A еще задана не до конца (не определены $e_i(1), e_i(2)$).

Для доказательства сильной связности графа рассмотрим рис. 4. Здесь вершины графа соответствуют подмножествам графа $G(\psi)$, а переход из подмножества S_1 в подмножество S_2, помеченный некоторым числом k (если не указано, то предполагается 1) и, возможно, некоторым условием (в круглых скобках) на индексы вершин множеств S_1, S_2, соответствует тому, что $Image(S'_1, k) \supseteq S'_2$ и из любой вершины множества S'_1 можно перейти в вершину из S'_2, где множества S'_1 и S'_2 получены из S_1 и S_2 соответственно наложением указанного на переходе условия (если множество содержит индекс, на который накладывается условие). Например, переход из $R_{sat}(*, i_c)$ в множество T_{sat}, помеченный числом $n+1$ и условием $i_c > 0$, означает, что $Image(R^+_{sat}, n+1) \supseteq T_{sat}$ и из любой вершины из R^+_{sat} можно перейти в некоторое состояние T_{sat}. Достижимость T_{sat} для этого перехода следует из условий (C1)

и $ED_{sat}4, 5, 9$, а существование перехода из R^+_{sat} в T_{sat} следует из определения R^+_{sat} и $ED_{sat}4, 5$. Остальные переходы в графе подмножеств следуют непосредственно из определений переходов графа $G(\psi)$.

Заметим также, что любая вершина q графа $G(\psi)$ принадлежит, по крайней мере, одному подмножеству S на рис. 4, причем множество $Image(\{q\}, 1)$ содержится в объединении множеств, в которые ведут переходы из множества S, и самого множества S (кроме вершин, окруженных пунктирной линией — s_0, s_1, s_2). Таким образом, по рисунку для любой вершины можно найти множество, в котором содержится $Preimage(\{q\}, 1)$.

Сильная связность графа $G(\psi)$ следует из того, что граф подмножеств на рис. 4 является сильносвязным, так как из z есть путь в любую вершину графа подмножеств и из любой вершины графа подмножеств есть путь в z.

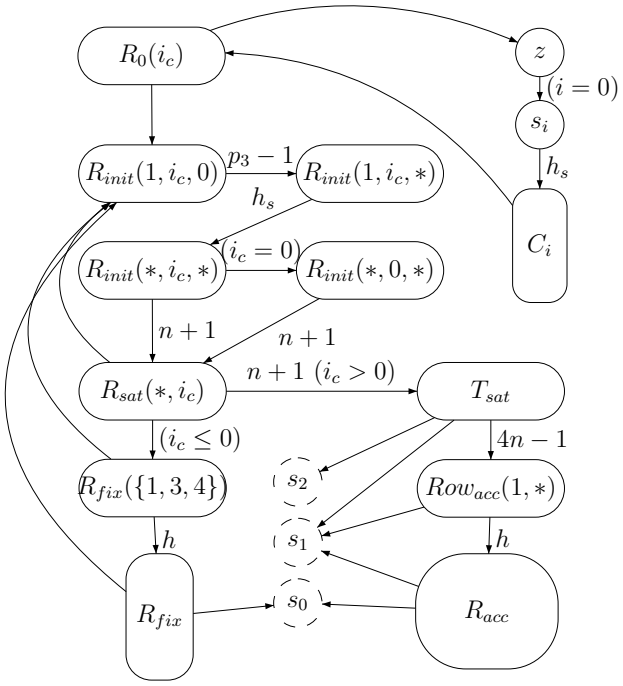

Рис. 4. Граф подмножеств $G(\psi)$

В следующем замечании показаны основные свойства Q_{spec}.

Замечание 6.

(A) Для $i \in \{0, 1, 2\}$ любой путь, начинающийся вне C_i и проходящий через C_i, содержит s_i (так как только переходы из $ED_{spec}4$ оканчиваются в $C_i(j)$ для $j > 1$).

(B) Для $r \in [0, h_s - 1]$ и $i \in \{0, 1, 2\}$ любой путь длины r из s_i оканчивается в $C_i(r+1)$ и любой путь длины r, оканчивающийся в $C_i(r + 1)$, начинается в s_i (по (A) и $ED_{spec}4$).

(C.1) $Preimage(R_0, h_s) = Q_{spec} \setminus \{z\}$.

(C.2) $Preimage(R_{init}(*, i, *), 1) \subseteq R_{init}(*, i, *) \cup R_{sat}(*, i) \cup R_0$ для $i \neq 0$.

(C.3) $Preimage(R_{init}(*, 0, *), 1) \subseteq R_{init}(*, 0, *) \cup R_{sat}(*, 0) \cup R_0 \cup R_{fix}$.

(C.4) $Preimage(R_{sat}(*, i), 1) \subseteq R_{init}(*, \{0, i\}, *) \cup R_{sat}(*, \{0, i\})$.

(D) $Preimage(R_{init}(*,i,*),h_s) \subseteq R_{init}(*,\{0,i\},*) \cup R_{sat}(*,\{0,i\}) \cup R_{fix} \cup Q_{spec}$ для $i \in [-2,m]$ по пунктам (C.1 ... 4).
(E.1) $Image(R_{init}(*,i_c,*),1) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c)$ для $i_c > 0$ в силу $ED_{init}5$.
(E.2) $Image(R_{sat}(*,i_c),1) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c) \cup T_{sat}$ для $i_c > 0$ в силу $ED_{sat}1, 4 \dots 7$.
(E.3) $Image(T_{sat},1) \subseteq T_{sat} \cup R_{acc} \cup \{s_0,s_1,s_2\}$ в силу $ED_{sat}8,9,10$.
(E.4) $Image(R_{acc},1) \subseteq R_{acc} \cup \{s_0,s_1,s_2\}$ в силу $ED_{acc}5,6,7$.
(E.5) $Image(\{s_0,s_1,s_2\},h_s) \subseteq C_0 \cup C_1 \cup C2$ в силу $ED_{spec}4$.
(F) $Image(R_{init}(*,i_c,*),h_s) \subseteq R_{init}(*,i_c,*) \cup R_{sat}(*,i_c) \cup T_{sat} \cup R_{acc} \cup C_0 \cup C_1 \cup C2$ для $i_c > 0$ в силу $(E.1 \dots 5)$.

Доказательство оценок для длин синхронизирующих слов u и v автоматов A и B соответственно проведем следующим образом: для *GOODCASE* пошагово построим синхронизирующее слово. При этом на каждом шаге будем отслеживать образ множества состояний автомата после применения уже построенного слова. Для *BADCASE* также пошагово докажем, что синхронизирующее слово v должно перемещать состояния аналогично тому, как перемещаются состояния в *GOODCASE* под действием слова u, либо можно выбрать несколько состояний в образе соответствующего префикса v, для которых значение *LRadius* по следующей лемме будет достаточно большим для получения требуемой оценки длины оставшегося суффикса v (по п. 2 леммы 1).

Лемма 3. Справедливы следующие неравенства для функции *LRadius*:

(A) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 \in \{s_0,s_1,s_2\}, q_2 \neq q_1$.
(B) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 = R_{init}(1,i_{c_1},i_{f_1})$, $i_{c_1} \neq 0$, $q_2 \in R_{init}(*,i_{c_2},*) \cup \cup R_{sat}(*,i_{c_2}) \cup T_{sat} \cup R_{fix} \cup R_{acc}$, $0 \neq i_{c_2} \neq i_{c_1}$.
(C) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 = R_{init}(1,i_{c_1},i_{f_1})$, $i_{c_1} = 0$, $q_2 \in R_{init}(*,i_{c_2},*) \cup \cup R_{sat}(*,i_{c_2}) \cup T_{sat} \cup R_{acc}$, $0 < i_{c_2}$.
(D) $LRadius(\{q_f,q_2,q_3\}) \geqslant p_3(kn-1)$, где $q_f = R_{sat}(1,i_{c_1})$, $q_2 = R_{init}(i_{r_2},0,i_{f_2})$, $q_3 \in R_{init}(*,i_{c_3},*) \cup R_{sat}(1,i_{c_3})$, $i_{r_2} < kn$, $0 < i_{c_3} \neq i_{c_1}$ и ($i_{f_2} \leqslant 3p_2+1$ или $i_{r_2} = 1$).
(E) $LRadius(\{q_1,q_2\}) \geqslant p_3(kn-1)$, где $q_1 \in Image(R_{fix},1)$, $q_2 \in Image(R_{acc},1)$, $\{q_1,q_2\} \neq \{s_0\}$, $q_1 \notin (R_{fix} \cup R_{acc})$ или $q_2 \notin (R_{fix} \cup R_{acc})$.

Доказательство. Пункт (A) непосредственно следует из пункта (B) замечания 6, так как для $r \in [0, h_s-1]$ путь из q_2 длины r не может заканчиваться в $C_i(r+1)$, где заканчивается путь длины r из q_1.

По пункту (B) замечания 1 кратчайший путь из $R_{init}(1,i_{c_1},*)$ вне R_{init} не меньше $h_s = p_3(kn-1)$. Поэтому для пунктов (B), (C) выполняется $LRadius(q_1, Q \setminus R_{init}(1,i_{c_1},*)) \geqslant h_s$. Следовательно, по п. 3 леммы 1

$$LRadius(\{q_1,q_2\},Q) \geqslant \min(h_s, LRadius(\{q_1,q_2\}, R_{init}(1,i_{c_1},*))),$$

т. е. достаточно доказать, что $LRadius(\{q_1,q_2\}, R_{init}(1,i_{c_1},*)) \geqslant h_s$. По п. 6 леммы 1 для этого достаточно показать, что $q_2 \notin Preimage(R_{init}(1,i_{c_1},*),h_s)$. А это, в свою очередь, следует из пункта (D) замечания 6.

Для доказательства пункта (D) предположим от противного, что есть пути P_f, P_2, P_3 одинаковой длины r меньше $p_3(kn-1)$, ведущие в одну из вершин q_f, q_2, q_3 соответственно. Из выбора q_2 и $ED_{init}1,2,4,7$ следует, что любой путь из q_2 длины не больше $p_3 - 3p_2 > 15n^2$ заканчивается в $R_{init}(*,0,*)$. Если для некоторого $r \leqslant 15n^2$ путь $P_f(1 \dots r)$ или $P_3(1 \dots r)$ оканчивается в $\{s_0,s_1,s_2\}$, то мы получим противоречие

нашего предположения с пунктом (A), примененного для вершин $P_2(r)$, $P_f(r)$ или $P_2(r)$, $P_3(r)$ соответственно.

Из пункта (E) замечания 4 следует, что для некоторого $r_0 \leqslant h+5n-6 \leqslant 15n^2$ путь $P_f(1 \dots r_0)$ оканчивается в $\{R_{init}(1,0,0), R_{init}(1,i_c,0), s_0, s_1, s_2\}$. Так как $h+5n-6 \leqslant \leqslant 15n^2$, то путь $P_f(1 \dots r_0)$ должен оканчиваться в $R_{init}(1,i_c,0)$.

Из того, что $i_{c_3} > 0$ и путь $P_3(1 \dots 15n^2)$ не проходит через $\{s_0, s_1, s_2\}$, следует, что $P_3(1 \dots 15n^2)$ целиком лежит в $R_{init}(*, i_{c_3}, *) \cup R_{sat}(*, i_{c_3}) \cup T_{sat} \cup R_{acc}$. Следовательно, мы получаем противоречие с пунктом (C) для вершин $P_f(r_0) = R_{init}(1,i_c,0)$ и $P_3(r_0)$. Таким образом, пункт (D) доказан.

Докажем теперь пункт (E). По определению переходов ED_{acc}, ED_{fix} получаем, что $q_1 \in R_{fix} \cup \{R_{init}(1,0,0), s_0\}$, а $q_2 \in R_{acc} \cup \{s_1, s_0\}$. Тогда из условия

$$q_1 \notin (R_{fix} \cup R_{acc}) \text{ или } q_2 \notin (R_{fix} \cup R_{acc})$$

следует, что или $q_1 \in \{R_{init}(1,0,0), s_0\}$, или $q_2 \in \{s_1, s_0\}$. Если q_1 или q_2 совпадает с s_0, то другая вершина по условию не совпадает с s_0, и по пункту (A) получим требуемое неравенство. Если q_2 совпадает с s_1, то из $q_1 \in R_{fix} \cup \{R_{init}(1,0,0), s_0\}$ следует $q_1 \neq s_1$, и снова по пункту (A) получаем требуемое неравенство. Осталось рассмотреть случай, когда $q_1 = R_{init}(1,0,0)$ а $q_2 \in R_{acc}$. В этом случае получаем нужное неравенство по пункту (C). ■

В следующей лемме полностью доказывается корректность сведения для *GOOD-CASE* при помощи утверждений о свойствах компонент графа $G(\psi)$.

Лемма 4. Существуют дораскраска автомата A (определение $e_1(i)$, $e_2(i)$) и слово u длины не больше $p(m,n)$, синхронизирующее автомат A.

Доказательство. Положим $u_{init} = ca^{p_3(kn-1)}b$. Тогда $|u_{init}| = (kn-1)p_3 + 2$ и по определениям переходов LD_{init} получаем, что $R_{init}.u_{init} = R_{sat}(1,*)$. По замечанию 2, существует слово u_{sat} длины $n+1$, такое, что множество $S = R_{sat}(1,*).u_{sat} = = R_{init}.u_{init}u_{sat}$ удовлетворяет замечанию 5, поэтому можно раскрасить R_{acc} (определить $e_1(i)$, $e_2(i)$) так, чтобы для $u_{acc} = w_n(b,b,\dots,b,c)$ выполнялось равенство $s_0 = S.u_{acc} = R_{init}.u_{init}u_{sat}u_{acc}$. Заметим также, что $|u_{acc}| = |w_n| = (4n+13)(n-2)+18$, поэтому

$$|u| = |u_{init}| + |u_{sat}| + |u_{acc}| = ((kn-1)p_3 + 2) + (n+1) + (18 + (4n+13)(n-2)) \leqslant p.$$

Осталось доказать, что $Q.u = s_0$. Так как $u(1) = c$, то $Q.u(1) = Q.c \subseteq R_{init}(1,0,0) \cup \cup Q_{spec}$. Поскольку $R_{init}(1,*,0).u(2 \dots |u|) = s_0$, то нужно показать, что $Q_{spec}.u(2 \dots |u|) = = s_0$. Так как высота любого столбца C_i плюс ширина строки R_0 меньше, чем $|u(2 \dots |u|-1)| = |u|-2$, и алфавит слова $u(2 \dots |u|-1)$ состоит из букв $\{a,b\}$, то по определениям $LD_{spec}1 \dots 6$ получаем, что $Q_{spec}.u(2 \dots |u|-1) = z$. Следовательно, $Q_{spec}.u(2 \dots |u|) = s_0$ и слово u синхронизирующее. ■

Рассмотрим теперь *BADCASE*. Следующая лемма соответствует шагу 1 и использует свойства компоненты R_{init}.

Лемма 5. Существует префикс v'_{init} слова v, такой, что $R_{init}(1,*,*).v'_{init} \cap R_{sat} \neq \neq \varnothing$. Если v'_{init} — кратчайший префикс с этим свойством, тогда $|v'_{init}| > (kn-1)p_3$ и существует слово v_{dop}, такое, что $v_{init} = v'_{init}v_{dop}$ — префикс слова v и $R_{sat}(1,*) \subseteq \subseteq R_{init}.v'_{init}v_{dop}$.

Доказательство. Прежде всего, такой префикс v'_{init} у слова v существует, потому что любые два состояния из различных столбцов R_{init} не могут быть сжаты внутри R_{init} и все исходящие дуги из R_{init} заканчиваются либо в R_{init}, либо в R_{sat}. Заметим также, что $|v'_{init}| > p_{add} = p_3(kn-1)$ по пункту (B) замечания 1.

Для доказательства леммы осталось показать, что найдется слово v_{dop}, такое, что $R_{sat}(1,*) \subseteq R_{init}.v'_{init}v_{dop}$.

Можно предполагать, что $R_{sat}(1,t) \not\subseteq R_{init}.v'_{init}$ для некоторого $t \in [-2,m]$. В противном случае $R_{sat}(1,*) \subseteq R_{init}(1,*,*).v'_{init}$ и в качестве v_{dop} можно взять пустое слово. Во всех остальных случаях покажем, что можно либо найти подходящее слово v_{dop}, либо выбрать несколько состояний из $Q.v'_{init}$, для которых значение $LRadius$ не меньше $p_3(kn-1)-1$, и тем самым получить противоречие с (E_v), используя неравенства

$$|v| \geqslant |v'_{init}| + LSynch_B(Q.v'_{init}) > p_3(kn-1) + LRadius(Q.v'_{init}) \geqslant \\ \geqslant 2p_3(kn-1) \geqslant 2p_{add} \geqslant (2-0{,}5\varepsilon)p. \qquad (EN_1)$$

Так как по условию множество $R_{init}(1,*,*).v'_{init} \cap R_{sat}$ не пустое, то из него можно выбрать состояние $q_f = R_{sat}(1,i_{c_1})$ для некоторого $i_{c_1} \in [-2,m]$.

Без ограничения общности предположим, что $v'_{init}(|v'_{init}|) = a$, т. е. $v'_{init} = wa$ для подходящего слова w. Тогда по выбору w и по пункту (A) замечания 1 для любого номера столбца i_c любой префикс слова w сохраняет полное множество остатков от деления на p_3 координат вектора множества $R_{init}(1,i_c,*)$. Формально:

$$\forall i_c \in [-2,m],\ i_f \in [0,p_3-1],\ l_1 \in [1,|w|] \\ \exists i_r \in [1,kn]\ R_{init}(i_r,i_c,i_f) \in R_{init}(1,*,*).w(1\ldots l_1). \qquad (E_0)$$

Вспомним, что $v'_{init} = wa$, и обозначим $S = R_{init}(1,*,*).w$. Рассмотрим все возможные случаи.

С л у ч а й 1. $\exists j \in [0,3p_2]\ R_{init}(kn,0,j) \notin S$. Тогда по (E_0) существует номер $i < kn$, такой, что $q'_2 = R_{init}(i,0,j) \in S$. Далее, так как $m > 3$, то по (E_0) для $0 < i_{c_3} \neq i_{c_1}$ можем выбрать состояние $q'_3 = R_{init}(i_{r_3},i_{c_3},i_{f_3})$ из множества S и положить $q_2 = q'_2.a$, $q_3 = q'_3.a$. Таким образом, есть три занятых состояния $q_f,q_2,q_3 \in S.a = R_{init}(1,*,*).v'_{init}$, такие, что $q_f = R_{sat}(1,i_{c_1})$, $q_2 = R_{init}(i,0,j).a = R_{init}(i_{r_2},0,i_{f_2})$, $q_3 \in R_{init}(*,i_{c_3},*) \cup \cup R_{sat}(1,i_{c_3})$, где $i_{r_2} < kn$, $0 < i_{c_3} \neq i_{c_1}$ и ($i_{f_2} \leqslant 3p_2+1$ или $i_{r_2} = 1$). По пункту (D) леммы 3 $LRadius(\{q_f,q_2,q_3\}) \geqslant (kn-1)p_3$, и получаем противоречие по (EN_1).

С л у ч а й 2. $\forall j \in [0,3p_2]\ R_{init}(kn,0,j) \in S$. Обозначим через x букву $w(|w|)$. Тогда

$$\forall j \in [0,3p_2-1]\ R_{init}(kn,0,j).x = R_{init}(kn,0,j+1), \qquad (E_2)$$

потому что только $R_{init}(kn,0,j)$ может перейти по x в $R_{init}(kn,0,j+1)$ и какое-то состояние из $S.x^{-1}$ переходит по x в $R_{init}(kn,0,j+1)$.

Пусть t — максимальное число, такое, что $v = wx^tv_2$ и $v_2(1) \neq x$. Заметим, что если $t > 0$, то $x = a$, потому что $v'_{init} = wa$.

С л у ч а й 2.1. $t > p_2$. Тогда $x = a$ и $R_{init}(kn,0,0) \in S$. По (E_0) выберем состояние в множестве S из столбца R_{init} с положительным номером, т. е. $q_3 = R_{init}(i_{r_3},i_{c_3},i_{f_3}) \in S$, где $i_{c_1} \neq i_{c_3} > 0$. Хотя кратчайший путь из $q_2 = R_{init}(kn,0,0)$ имеет длину меньше p_2, но так как wa^t — префикс v, $q_2 \in Q.w$ и выполняется равенство (E_1), то q_2 остается в $R_{init}(kn,0,*)$ при применении a^t и $t > p_2$. Следовательно, оценка на $LRadius$ для состояний q_f,q_2,q_3 получается так же, как и в пункте (D) леммы 3, с той разницей, что в этом случае путь из q_2 длины p_2 зафиксирован. Следовательно,

$$LRadius(q_f.x,q_2.x,q_3.x) \geqslant LRadius(q_f,q_2,q_3) - 1 \geqslant p_3(kn-1) - 1,$$

и получаем противоречие по (EN_1).

С л у ч а й 2.2. $t \leqslant p_2$. Так как $3p_2 \geqslant p_1$ и выполняются условия случая 2, то $R_{init}(kn, 0, j) \in S$ для $j \in [0, p_1]$. По (E_2) получаем, что $R_{init}(kn, 0, j).x^t v_2(1) = R_{init}(kn, 0, j + t).v_2(1)$, потому что $j + t < p_2 + p_1 < 3p_2 - 1$. Так как $R_{init}(kn, 0, j + t).x = R_{init}(kn, 0, j + t + 1)$ и $v_2(1) \neq x$, то из определений дуг $ED_{1,6,7}$ следует, что $R_{init}(kn, 0, j+t).v_2(1)$ совпадает с $R_{sat}(1, ((j+t+1) \bmod p_1)-1)$ либо с $R_{init}(1, 0, j+t+1)$.

Если для любого $j \in [0, p_1]$ состояние $R_{init}(kn, 0, j).x^t v_2(1)$ совпадает с $R_{sat}(1, ((j + t + 1) \bmod p_1) - 1)$, то $R_{sat}(1, *) \subseteq S.x^t v_2(1)$, и в качестве v_{dop} можно выбрать $x^{t-1}v_2(1)$.

В противном случае для некоторого $j \in [0, p_1]$ состояние $R_{init}(kn, 0, j).x^t v_2(1)$ совпадает с $R_{init}(1, 0, j + t + 1)$. Обозначим это состояние через q_1. Используя (E_2), выберем $q'_2 \in R_{init}(*, i_{c_2}, *) \cap S$, где $i_{c_2} > 0$. Так как $t \leqslant p_2 < h_s$ и $i_{c_2} > 0$, то по пункту (F) замечания 6 получаем $q_2 = q'_2.x^t \in R_{init}(*, i_{c_2}, *) \cup R_{sat}(*, i_{c_2}) \cup R_{acc} \cup C_0 \cup C_1 \cup C_2$. Если $q_2 \in C_0 \cup C_1 \cup C_2$, то можно выбрать $t_1 \leqslant t$ так, чтобы $\widetilde{q_2} = q'_2.x^{t_1} \in \{s_0, s_1, s_2\}$. Тогда если положить $\widetilde{q_1} = R_{init}(kn, 0, j).x^{t_1}$, то $\widetilde{q_1} \notin \{s_0, s_1, s_2\}$, так как $\widetilde{q_1}.x^{t-t_1} = q_1 \in R_{init}(1, *, *)$ и $Image(\{s_0, s_1, s_2\}, h_s) \subseteq Q_{spec}$ в силу $ED_{spec}5$. Следовательно, $\widetilde{q_1} \neq \widetilde{q_2}$ и $\widetilde{q_2} \in \{s_0, s_1, s_2\}$, и получаем, что $LRadius(\{\widetilde{q_1}, \widetilde{q_2}\}) \geqslant p_3(kn - 1)$ по пункту (A) леммы 3 и $\widetilde{q_1}, \widetilde{q_2} \in Q.v(1 \dots |w| + t_1)$.

Иначе $q_2 \in R_{init}(*, i_{c_2}, *) \cup R_{sat}(*, i_{c_2}) \cup R_{acc}$, и тогда $LRadius(\{q_1, q_2\}) \geqslant p_3(kn - 1)$ по пункту (C) леммы 3 и $q_1, q_2 \in Q.v(1 \dots |w| + t)$.

Следовательно, в случае 2.2 после применения более длинного (с длиной больше $|v'_{init}|$) префикса v получаем несколько занятых состояний, значение $LRadius$ для которых не меньше $p_3(kn - 1)$, и тем самым получаем противоречие с (E_v) аналогично (EN_1). Таким образом, рассмотрены все возможные случаи, и лемма доказана. ■

Таким образом, шаг 1 завершен. Следующая лемма соответствует шагу 2 и позволяет отделить случай *BADCASE* от *GOODCASE*.

Лемма 6. Существует разложение $v = v_{init}v_{sat}v_2$, такое, что

$$|v_{sat}| = n + 1,\ R_{sat}(1, *).v_{sat} \cap R_{fix} = R_{fix}(\{1, 3, 4\})$$

и для $S = R_{sat}(1, *).v_{sat} \cap R_{acc}$ верны следующие свойства:

(B0) $S \subseteq Row_{acc}(1, *)$, т. е. полностью содержится в 1-й строке;
(B1) номера столбцов состояний из S четны;
(B2) $|S| > n$.

Доказательство. Так как $LRadius(R_{sat}(1, *)) > n + 1$ и по лемме 5 $R_{sat}(1, *) \subseteq R_{init}.v_{init}$, то существует разложение $v = v_{init}v_{sat}v_2$, где $|v_{sat}| = n + 1$. Если при применении v_{sat} к $R_{sat}(1, *)$ нет использования c-переходов, то утверждение доказано по лемме 2. Предположим от противного, что при применении v_{sat} используется c-переход; тогда существуют состояние $q'_1 = R_{sat}(1, i_{c_1})$ и минимальное натуральное число $t_1 \leqslant n + 1$, такие, что если $q_1 = q'_1.v_{sat}(1 \dots t_1)$, то

– $q_1 \in \{R_{init}(1, i_{c_1}, 0), s_1, s_2\}$, если $i_{c_1} > 0$; выберем $q_2 = R_{sat}(1, 0).v_{sat}(1 \dots t_1)$, тогда $q_2 \in R_{sat}(*, 0) \cup R_{fix} \cup R_{init}(1, 0, *)$;
– $q_1 = R_{init}(1, i_{c_1}, 0)$, если $i_{c_1} \leqslant 0$; выберем $q_2 = R_{sat}(1, 1).v_{sat}(1 \dots t_1)$, тогда $q_2 \in R_{sat}(*, 1) \cup R_{acc} \cup R_{init}(1, 1, *) \cup \{s_0, s_1, s_2\}$.

В обоих случаях $q_1, q_2 \in R_{init}.v_{init}v_{sat}(1 \dots t_1)$ и по пунктам (A),(B) или (C) леммы 3 получаем $|v| \geqslant |v_{init}| + LRadius(\{q_1, q_2\}) \geqslant 2(kn - 1)p_3 \geqslant 2p_{add} \geqslant (2 - 0{,}5\varepsilon)p$. Таким образом, получаем противоречие с (E_v), и лемма доказана. ■

Следующая лемма соответствует шагу 3 для *BADCASE* и завершает доказательство для этого случая.

Лемма 7. *BADCASE*: Предположим, что $v = v_{init}v_{sat}v_{acc}$ и слово v — синхронизирующее для B. Тогда $|v| > (2 - 0{,}5\varepsilon)p(m, n)$

Доказательство. Заметим, что по лемме 6 после применения $v_{init}v_{sat}$ занятые состояния остались как в R_{fix}, так и в R_{acc}. Так как между R_{fix} и R_{acc} нет дуг, а v_{acc} должно сжимать оставшиеся состояния, то у слова v_{acc} есть префикс, который перемещает некоторое состояние из $(R_{fix} \cup R_{acc}) \cap Q.v_{init}v_{sat}$ вне $R_{fix} \cup R_{acc}$. Пусть $w_{acc}x$ — разложение кратчайшего префикса v_{acc} с этим свойством.

По определению $w_{acc}x$ можно выбрать состояние q'_1 из $(R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}$ так, что если положить $q_1 = q'_1.x$, то $q'_1.x \notin (R_{acc} \cup R_{fix})$. Заметим, что по выбору префикса множество $Q.v_{init}v_{sat}w_{acc}$ пересекается как с R_{fix}, так и с R_{acc}. Следовательно, можно выбрать $q'_2 \in (R_{fix} \cup R_{acc}) \cap Q.v_{init}v_{sat}w_{acc}$ так, что если $q'_1 \in R_{fix}$, то $q'_2 \in R_{acc}$, или наоборот — если $q'_1 \in R_{acc}$, то $q'_2 \in R_{fix}$.

Если $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x \neq s_0$, то можно считать, что $\{q'_1.x, q'_2.x\} \neq \neq \{s_0\}$. Тогда по пункту (E) леммы 3 получаем, что $LRadius(\{q'_1.x, q'_2.x\}) \geqslant (kn-1)p_3$. Следовательно, $|v_{init}v_{sat}w_{acc}x| \geqslant |v_{init}v_{sat}w_{acc}x| + Lradius(\{q'_1.x, q'_2.x\}) \geqslant 2(kn-1)p_3$, и утверждение леммы доказано.

Осталось рассмотреть случай, когда $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$. Без ограничения общности предположим, что $w_{acc}(1) = a$ (в противном случае доказательство идентично с точностью до подстановки на алфавите). Тогда по выбору w_{acc} и равенству $((R_{acc} \cup R_{fix}) \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$ оказываемся в условиях замечания 3 и получаем, что слово $w_{acc}x$ равно w_n для некоторых $d_1, d_2, \ldots, d_n \in \Sigma$, т. е.

$$w_{acc}x = a^3 d_1 a^{12+4n} d_2 a^{12+4n} d_3 \ldots a^{12+4n} d_{n-1} a^{13} d_n.$$

Объединяя это с леммой 6 и выбором $w_{acc}x$, оказываемся в условиях замечания 5 для $S = Q.v_{init}v_{sat} \cap Row_{acc}(1, *)$ и получаем противоречие с равенством $((R_{acc} \cup R_{fix}) \cap \cap Q.v_{init}v_{sat}w_{acc}).x = s_0$. ■

Таким образом, теорема полностью доказана в обоих случаях. ■

Заметим, что из теоремы 1 напрямую не следует, что в предположении $\mathcal{P} \neq \mathcal{NP}$ не существует полиномиального алгоритма, находящего оптимальную раскраску, так как для её поиска не требуется находить длину кратчайшего синхронизирующего слова.

Если бы задача поиска кратчайшего синхронизирующего слова для автомата принадлежала классу $\mathcal{P}$, то полиномиальный алгоритм, находящий оптимальную раскраску, можно было бы достроить до полиномиального алгоритма, решающего задачу OCV, и получить противоречие с теоремой. Однако ввиду результата [8] для этой задачи не существует даже приближенного полиномиального алгоритма с любой конечной относительной погрешностью.

Тем не менее можно получить аналогичный результат для задачи поиска оптимальной раскраски. Заметим, что алгоритм, решающий OC с относительной погрешностью $\beta < 2$, должен для произвольного допустимого орграфа G возвращать его синхронизирующую раскраску A, такую, что $\mathfrak{C}(A) \leqslant \beta \cdot \mathrm{OPT}(G)$.

Следствие 1. В предположении $\mathcal{P} \neq \mathcal{NP}$ любой полиномиальный алгоритм, решающий OC для класса графов со степенью исхода вершин 3, имеет относительную погрешность не меньше 2.

Доказательство. Приведем здесь только план доказательства, опуская технические детали. Предположим от противного, что существует алгоритм Alg, решающий с относительной погрешностью $2-\varepsilon$ задачу OC. Используя такое же построение графа $G(\psi)$ по ψ, что и в теореме 1, для получения противоречия достаточно показать, как по раскраске B, возвращаемой алгоритмом Alg на графе $G(\psi)$, определить выполнимость ψ.

Выполнимость ψ определяется следующим образом. Без ограничения общности можно считать, что дуга из $R_{fix}(1)$ в $R_{fix}(2)$ помечена буквой a. Далее, для $i \in [1, n-1]$ рассчитаем длины путей внутри D_i, помеченных степенью буквы a, и в соответствии с этими значениями восстановим, какие состояния из $Row_{acc}(1,*)$ могли быть сжаты под действием слова $w_n(d_1, d_2, \ldots, d_n)$, где d_i равен символу на b-переходе из $D_i(1)$. Это можно сделать, используя рассуждения из замечания 5 или рассмотрев прообраз $D_{n-1}(35)$ под действием слова w_n без последнего символа. Заметим, что если для некоторого j в D_j нет пути, помеченного степенью a, то нужно доказать, что ψ невыполнима. Это можно сделать аналогично доказательству $BADCASE$ в лемме 7.

Далее, из определения переходов в R_{sat} и T_{sat} однозначно восстанавливаются значения переменных $b_1, b_2, \ldots, b_n$, которые должны удовлетворять ψ (так же, как это делалось для $BADCASE$ в замечании 2). Если эти значения действительно удовлетворяют ψ, то ясно, что формула ψ выполнима. В противном случае нужно показать, что ψ невыполнима. Для этого по построению достаточно доказать, что $\mathfrak{C}(B) \geqslant (2-0{,}5\varepsilon)p$.

Это утверждение доказывается аналогично случаю $BADCASE$, с той разницей, что вместо аргумента о том, что ψ невыполнима, в замечании 2 используем аргумент о том, что ψ должна быть выполнима именно на наборе $b_1, b_2, \ldots, b_n$, так как в противном случае получим такое множество занятых состояний после шага 2 внутри $Row_{acc}(1,*)$, которое ввиду раскраски R_{acc} не может быть сжато словом $w_n(d_1, d_2, \ldots, d_n)$. ■

3. Случай двухбуквенного алфавита

Теорема доказана для трехбуквенного алфавита. Очевидно, что для однобуквенного алфавита задача не имеет смысла, так как единственный сильносвязный автомат с одной буквой является циклом по этой букве и не может быть синхронизируемым. Несложно также проверить, что если в алфавите больше трех букв, то, продублировав необходимое количество раз переходы по букве c в автомате A из теоремы, можно доказать такой же результат и в этом случае. Возникает естественный вопрос: верно ли утверждение для двухбуквенного алфавита? Следующее следствие показывает справедливость результата и в этом случае.

Следствие 2. В предположении $\mathcal{P} \neq \mathcal{NP}$ любой приближенный полиномиальный алгоритм, решающий задачу OCV для подкласса автоматов с двухбуквенным алфавитом, имеет относительную погрешность не меньше 2.

Доказательство. Некоторые технические детали доказательства этого следствия опущены для краткости. Представим только изменения в структуре автомата A, которые показаны на рис. 5. Каждое состояние $q \in Q$, кроме s_0, заменяется тремя состояниями q^{top}, q^a, q^b, и переходы переопределяются следующим образом:

$$q^{top}.x = q^x, \quad q^x.a = (q.c)^{top}, \quad q^x.b = (q.x)^{top}.$$

Тогда буква c соответствует xa, а буква $y \in \{a, b\} - yb$.

Состояние s_0 заменяется двумя состояниями ${s_0}^{top}, {s_0}^{bot}$. Обозначим состояние $C_0(2)$ через g. Тогда

$${s_0}^{top}.a = {s_0}^{bot}, \quad {s_0}^{top}.b = g^{top}, \quad {s_0}^{bot}.a = {s_0}^{bot}.b = g^{top}.$$

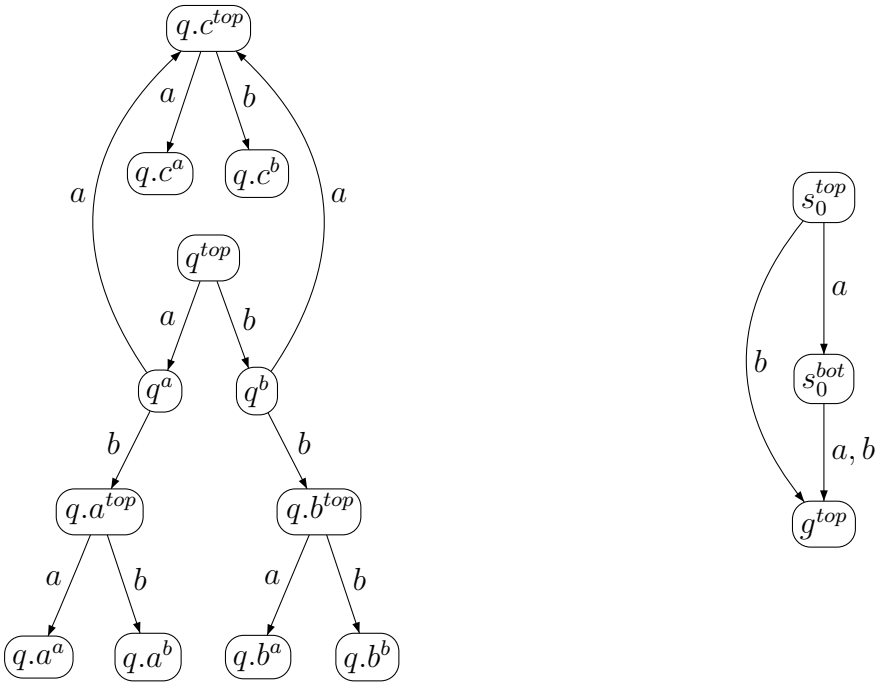

Рис. 5. Замены в автомате A

Легко проверить, что в случае *GOODCASE* $|u_{new}| = 2|u_{old}| + 1 \leqslant 2p(m,n)$, а в случае *BADCASE* $|v_{new}| \geqslant 2|v_{old}| \geqslant 2((2-0{,}5\varepsilon)p(m,n)) = (4-\varepsilon)p(m,n)$. Здесь u_{old}, v_{old} соответствуют синхронизирующим словам для старых автоматов A и B. Для завершения доказательства заметим, что отношение выражений $(4-\varepsilon)$ и $2p(m,n)$ стремится к 2 при $\varepsilon \to 0$. ■

Итак, мы доказали главный результат. Более того, мы показали, что он верен для случая общей степени исхода, равной 2.

В связи с доказанными результатами можно выделить два естественных открытых вопроса: о существовании эффективного алгоритма для некоторого подкласса допустимых графов и о минимальной возможной погрешности полиномиального алгоритма для задачи вычисления значения оптимальной раскраски.

ЛИТЕРАТУРА

1. *Adler R. L., Weiss B., and Goodwyn L. W.* Equivalents of topological Markov shifts // Isr. J. Math. 1977. No. 27. P. 49–63.
2. *Trahtman A. N.* The Road Coloring Problem // arxiv:0709.0099. 2007.
3. *Volkov M.* Synchronizing Automata and the Cerny conjecture // LNCS. 2008. V. 5196. P. 11–27.
4. *Černý J.* Poznámka k homogénnym eksperimentom s konečnými automatami // Matematicko-fyzikalny Časopis Slovensk. Akad. Vied (in Slovak). 1964. V. 14. No. 3. P. 208–216.
5. *Adler R. L. and Weiss B.* Similarity of automorphisms of the torus // Mem. Amer. Math. Soc. 1970. V. 98. P. 1–43.
6. *Beal M. and Perrin D.* A quadratic algorithm for road coloring // arXiv:0803.0726. 2008.

7. *Eppstein D.* Reset sequences for monotonic automata // SIAM J. Comput. 1990. V. 19. P. 500–510.
8. *Berlinkov M.* Approximating the Minimum Length of Synchronizing Words is Hard // LNCS. 2010. V. 6072. P. 37–47.
9. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.