

# СИНХРОНИЗАЦИЯ КОНЕЧНЫХ АВТОМАТОВ. I

М. В. Волков

## Аннотация

Дан обзор современного состояния теории синхронизируемых автоматов.

## Содержание

<b>Введение</b>	<b>2</b>
<b>1 Синхронизируемые автоматы: что, где, когда</b>	<b>3</b>
1.1 Конечные автоматы и их графы . . . . .	3
1.2 Синхронизируемые автоматы . . . . .	4
1.3 Как возникало понятие синхронизируемого автомата . . . . .	5
1.4 Синхронизируемые автоматы и коды . . . . .	8
1.5 Переоткрытие в промышленной механике . . . . .	10
1.6 Переоткрытие в теории подстановочных систем . . . . .	12
1.7 Алгебраические аспекты . . . . .	14
1.8 Другие приложения и связи . . . . .	16
<b>2 Алгоритмические и сложностные аспекты</b>	<b>17</b>
2.1 Проверка автомата на синхронизируемость . . . . .	17
2.2 Синхронизируемые автоматы и регулярные языки . . . . .	22
2.3 Сложность вычисления порога синхронизации . . . . .	24
2.4 Сложность аппроксимации порога синхронизации . . . . .	28
<b>3 Гипотеза Черни</b>	<b>29</b>
3.1 Гипотеза Черни: формулировка и немного истории . . . . .	29
3.2 Верхние оценки. Оценка Пэна–Франкля . . . . .	31
3.3 Верхние оценки. Недавние продвижения . . . . .	36
3.4 Метод расширения . . . . .	42
3.5 Сводка частичных результатов . . . . .	46
3.6 Экспериментальные результаты . . . . .	55
<b>Заключение</b>	<b>57</b>
<b>Список литературы</b>	<b>57</b>

---

Работа выполнена при поддержке РФФИ (грант №19-11-50120)

# Введение

Синхронизация конечных автоматов – направление дискретной математики, первоначальные понятия которого очень прозрачны, поскольку непосредственно апеллируют к повседневной интуиции. В то же время это направление имеет разнообразные и зачастую неожиданные связи со многими разделами чистой математики и компьютерных наук, а также вполне практические приложения. Его особенная привлекательность состоит в том, что оно быстро приводит к интересным и трудным задачам, среди которых имеется целая россыпь замечательных по простоте формулировки открытых проблем. Идеальным примером здесь может служить гипотеза Черни, обсуждаемая в §3.1 ниже: понятия, участвующие в её формулировке, предельно элементарны, но тем не менее гипотеза остаётся неподтверждённой и непровергнутой более полувека.

В ходе изучения синхронизируемых автоматов наработан обширный материал, который продолжает прирастать впечатляющими темпами. Вышедшие более 10 лет назад обзоры по синхронизируемости [93, 159, 182] заметно устарели. Поэтому естественно попытаться собрать и систематизировать новейшие достижения теории, чтобы вычленив точки её дальнейшего роста. Наряду с этим, уже упомянутая простота исходных понятий делает заманчивой и другую цель – дать доступное неспециалисту изложение начал теории на русском языке. Эта двойственность целей привела к двухслойной структуре текста: в нём развёрнутое представление базовых концепций и основополагающих фактов перемежается более компактно написанными дополнениями, включающими сводки свежих результатов и постановки новых исследовательских задач. Предполагается, что начинающий читатель может пропускать такие дополнения, в то время как эксперт может читать только их, выискивая интересные для себя сведения и вопросы.

В процессе работы задуманная статья быстро росла в объёме, и в конечном счёте пришлось разделить её на две части. Первая часть состоит из трёх параграфов. В §1 понятие синхронизируемости вводится для классического случая полных детерминированных автоматов. Объяснив, что такое синхронизируемость, мы также обсуждаем, где и когда возникало это понятие, – оно часто переоткрывалось математиками, информатиками и инженерами. В §2 представлены алгоритмы распознавания синхронизируемости; здесь же обсуждается вычислительная сложность основных проблем, связанных с синхронизацией. В §3 формулируется гипотеза Черни и даётся обзор связанных с ней результатов. Заключение кратко представляет содержание второй части статьи.

Я старался излагать ключевые результаты так, чтобы их можно было понять на основе знакомства с началами алгоритмики в духе учебника [55]. Часть обсуждений в §2 использует понятия и результаты теории сложности вычислений в объёме начальных глав учебника [125], но §3 от этих сведений практически не зависит.

Список литературы включает только источники, упоминаемые в статье; попытка составить сколько-нибудь полную библиографию по синхронизируемым автоматам противоречила бы аксиоме Козьмы Пруткова «Нельзя объять необъятное». Заранее приношу извинения тем авторам, чьи публикации не нашли отражения в этом обзоре. Как правило, в список литературы не включались препринты и статьи в трудах конференций, перекрытые последующими журнальными статьями.

Статья опирается на конспекты курсов лекций, исполнявшихся в разные годы в Уральском федеральном университете и за рубежом. Я признателен слушателям этих лекций за интерес и ценные замечания; особой благодарности заслуживают Грегори Джевенс и Леонид Лагунов. С 2010 г. вместе с Яркко Кари я работал над главой [96] для всё ещё не опубликованной справочной книги по теории автоматов; я весьма благодарен Яркко как за сотрудничество в работе над этой главой, так и за любезное разрешение включить некоторые материалы из неё в настоящую статью.

# 1 Синхронизируемые автоматы: что, где, когда

## 1.1 Конечные автоматы и их графы

*Конечный автомат* – это простое, но чрезвычайно продуктивное понятие, которое отражает очень важную идею взаимодействия объекта с окружающей средой. Есть много разновидностей конечных автоматов; здесь мы имеем дело с самой простой из них: полными детерминированными конечными автоматами. *Полный детерминированный конечный автомат* (кратко ДКА) – это тройка  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ , где  $Q$  – множество состояний,  $\Sigma$  – входной алфавит, а  $\delta: Q \times \Sigma \rightarrow Q$  – функция переходов.<sup>1</sup> Элементы множеств  $Q$  и  $\Sigma$  называются *состояниями* и (*входными*) *буквами* соответственно. Мы считаем, что ДКА эволюционирует в дискретном времени. В каждый момент времени он находится в определённом состоянии  $q \in Q$ . В течение следующей единицы времени на вход автомата поступает ровно одна буква  $a \in \Sigma$ , в результате чего автомат переходит в состояние  $\delta(q, a)$ .

Прокомментируем три прилагательных (полный, детерминированный и конечный) в приведённом выше определении. Атрибут «конечный» относится к тому факту, что как множество состояний, так и входной алфавит всегда предполагаются конечными; мы также молчаливо предполагаем, что оба эти множества не пусты. Атрибут «детерминированный» означает, что состояние  $\delta(q, a)$  однозначно определяется парой  $(q, a) \in Q \times \Sigma$ , так что правило перехода действительно является функцией. Атрибут «полный» подчеркивает, что функция переходов предполагается полностью определённой:  $\delta(q, a)$  должно существовать для каждой пары  $(q, a) \in Q \times \Sigma$ .

Конечные автоматы допускают очень удобное визуальное представление в виде помеченных графов. Мы предполагаем, что читатель знаком с идеей графа; уточним, какие именно графы мы будем использовать. В данной статье *граф* – это четвёрка множеств и отображений: множество *вершин*  $V$ , множество *ребер*  $E$ , отображение  $h: E \rightarrow V$ , которое отображает каждое ребро в его *начало* и отображение  $t: E \rightarrow V$ , которое отображает каждое ребро в его *конец*. Отметим, что в наших графах допускаются рёбра с общим началом и общим концом; такие рёбра называются *параллельными*. Допускаются и петли (*петлёй* называют ребро, начало и конец которого совпадают). Таким образом, наши графы на самом деле являются ориентированными мультиграфами, но, поскольку другие виды графов нам не встретятся, мы используем короткий термин. *Помеченный граф* оснащён дополнительным отображением  $E \rightarrow \Lambda$ , где  $\Lambda$  – множество, именуемое *алфавитом меток*; это отображение сопоставляет каждому ребру его *метку*. Ребро с началом  $v$ , концом  $v'$  и меткой  $a$  обозначим знакосочетанием  $v \xrightarrow{a} v'$ .

ДКА  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$  изображается как помеченный граф с множеством вершин  $Q$ , алфавитом меток  $\Sigma$  и множеством ребер  $\{q \xrightarrow{a} q' : q, q' \in Q, a \in \Sigma, \delta(q, a) = q'\}$ . Таким образом, переход из состояния  $q$  в состояние  $q'$ , вызванный входной буквой  $a$ , изображается ребром с началом  $q$ , концом  $q'$  и меткой  $a$ . Например, рисунок 1 изображает ДКА  $\mathcal{C}_4$  с четырьмя состояниями 0, 1, 2, 3, двумя входными буквами  $a$  и  $b$  и функцией перехода

$$\delta(i, a) = \begin{cases} 1, & \text{если } i = 0, \\ i, & \text{если } i = 1, 2, 3; \end{cases} \quad \delta(i, b) = i + 1 \pmod{4} \text{ для } i = 0, 1, 2, 3.$$

Здесь и далее рёбра с несколькими метками заменяют пучки параллельных ребер. Так, ребро  $0 \xrightarrow{a,b} 1$  на рис. 1 заменяет параллельные рёбра  $0 \xrightarrow{a} 1$  и  $0 \xrightarrow{b} 1$ .

<sup>1</sup>В такой форме конечный автомат впервые определил, по-видимому, Ю. Т. Медведев [118].

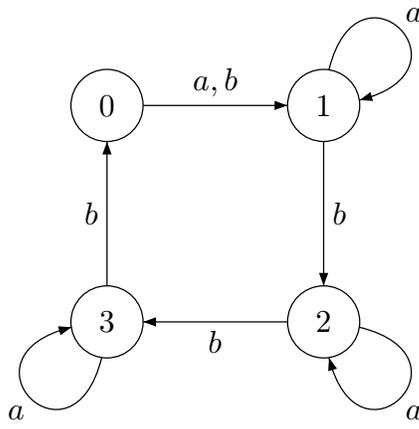


Рис. 1: Автомат  $\mathcal{C}_4$

Если опустить метки на графе, изображающем некоторый ДКА, получится граф, называемый *носителем* этого ДКА. Мы свободно применяем понятия и термины теории графов к ДКА, подразумевая, что они применяются к соответствующим носителям. Например, автомат  $\mathcal{C}_4$  на рис. 1 сильно связан, поскольку таков его носитель.

*Словом* над алфавитом  $\Sigma$  называется конечная (возможно, пустая) последовательность букв из  $\Sigma$ . За *пустым словом* закрепим символ  $\varepsilon$ . Множество всех слов над  $\Sigma$ , включая  $\varepsilon$ , обозначается  $\Sigma^*$ . Оно является *моноидом*, т.е. полугруппой с единицей относительно операции приписывания слов; единицей этого моноида служит  $\varepsilon$ . Множество всех непустых слов над  $\Sigma$  обозначается  $\Sigma^+$ . Если  $w = a_1 \cdots a_\ell \in \Sigma^+$ , то число  $\ell$  называется *длиной* слова  $w$  и обозначается через  $|w|$ . Считаем, что  $|\varepsilon| = 0$ .

Функцию переходов ДКА  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$  распространяют до функции  $Q \times \Sigma^* \rightarrow Q$  (по-прежнему обозначаемой  $\delta$ ) следующим образом. Для каждого  $q \in Q$  положим  $\delta(q, \varepsilon) := q$ , и если  $w = a_1 \cdots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , то  $\delta(q, w) := \delta(\dots\delta(\delta(q, a_1), a_2), \dots, a_\ell)$ . Когда мы работаем с каким-то фиксированным ДКА, то, как правило, не указываем функцию переходов явно: пишем  $\langle Q, \Sigma \rangle$  вместо  $\langle Q, \Sigma, \delta \rangle$  и  $q \cdot w$  вместо  $\delta(q, w)$ .

## 1.2 Синхронизируемые автоматы

*Определение 1.1.* Полный детерминированный конечный автомат  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *синхронизируемым*, если существуют слово  $w \in \Sigma^*$  и состояние  $s \in Q$  такие, что  $q \cdot w = s$  для всех  $q \in Q$ . В этом случае слово  $w$  называют *синхронизирующим* и говорят, что оно *синхронизирует автомат  $\mathcal{A}$  к состоянию  $s$* .

*Пример 1.1.* Автомат  $\mathcal{C}_4$ , изображённый на рис. 1, синхронизируем.

Утверждение примера 1.1 не совсем тривиально: не так легко угадать синхронизирующее слово для  $\mathcal{C}_4$ , глядя на рис. 1. Одним из возможных синхронизирующих слов служит  $abbbabbba$ : оно синхронизирует автомат  $\mathcal{C}_4$  к состоянию 1. (Позже мы увидим, что  $abbbabbba$  на самом деле является кратчайшим синхронизирующим словом для  $\mathcal{C}_4$ .) Это следует из непосредственных вычислений, представленных в табл. 1.

Вычисления, продемонстрированные в табл. 1, поясняют выбор термина «синхронизация». Представим, что имеются четыре копии автомата  $\mathcal{C}_4$ , изначально находящиеся в различных состояниях. Если одновременно применить слово  $abbbabbba$  к каждой из копий, то они «синхронизируются», т.е. перейдут в одно и то же состояние (и их дальнейшее поведение при одновременном применении любого другого слова будет оставаться синхронным).

Таблица 1: Пошаговое применение слова *abbbabbba* к состояниям автомата  $\mathcal{C}_4$

Состояние	<i>a</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>a</i>	<i>b</i>	<i>b</i>	<i>b</i>	<i>a</i>
0	1	2	3	0	1	2	3	0	1
1	1	2	3	0	1	2	3	0	1
2	2	3	0	1	1	2	3	0	1
3	3	0	1	2	2	3	0	1	1

Здесь уместен терминологический комментарий. В англоязычной литературе синхронизируемые автоматы обычно называют *synchronizing automata* или *synchronizable automata*, а синхронизирующие слова – *synchronizing words* или *reset words*. Встречаются и другие термины, такие, как ‘directable’, ‘cofinal’, ‘collapsible’, ‘resettable’, ‘recurrent’, ‘initializable’ для синхронизируемых автоматов и такие, как ‘directing’, ‘recurrent’, ‘initializing’ для синхронизирующих слов. С другой стороны, термин «синхронизация» и его английский аналог часто используются в значениях, отличных от введённого выше.

Сразу же зафиксируем очевидное, но полезное свойство синхронизирующих слов:

**Лемма 1.1.** *Если слово  $w \in \Sigma^*$  синхронизирует ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  к состоянию  $s \in Q$ , то для любых  $u, v \in \Sigma^*$  слово  $uwv$  синхронизирует  $\mathcal{A}$  к состоянию  $s.v$*

На языке алгебры лемма 1.1 означает, что для любого ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  множество  $\text{Sync } \mathcal{A}$  всех его синхронизирующих слов образует идеал в моноиде  $\Sigma^*$ . (Подмножество  $I$  моноида  $M$  называется *идеалом*, если  $m_1 i m_2 \in I$  для любых  $i \in I$  и  $m_1, m_2 \in M$ .)

### 1.3 Как возникало понятие синхронизируемого автомата

Представленная выше концепция синхронизируемого автомата оформилась в начале 1960-х гг. В литературе по синхронизируемым автоматам стандартной ссылкой служит статья Яна Черни [44], опубликованная в 1964 г. В действительности по крайней мере два исследователя пришли к тому же понятию несколько раньше. Диссертация Чунг Лаунга Лю [112], выполненная в 1962 г. в Массачусетском технологическом институте, содержит целую главу, посвящённую систематическому изучению синхронизируемых автоматов. Более того, сам термин «синхронизируемый автомат», по-видимому, происходит именно из этой диссертации: Лю использовал термин ‘synchronizable’, в то время как Черни пользовался термином ‘directable’. Лю, с которым (как и с Черни) я консультировался в своих изысканиях по истокам понятия синхронизации, подчеркнул (в электронном письме от 10 марта 2016 г.) роль своего научного руководителя Дина Ардена. Синхронизируемые автоматы появились также в отчётах [107, 108], представленном Артуром Э. Леммелем в Департамент армии США в 1956 и 1963 гг. соответственно; Леммель использовал термин «перезагружаемые машины» (‘resettable machines’). Отчёты Леммеля менее проработаны по сравнению с [112] и [44], но второй из них содержит эквивалент ключевого определения 1.1 (правда, только для случая сильно связанных автоматов) и некоторые ценные наблюдения. Не исключено, что изучение советских отчётов о НИР, когда таковые будут рассекречены, может добавить к списку первооткрывателей синхронизируемых автоматов и имена отечественных учёных.<sup>2</sup>

В статье Черни [44] понятие синхронизируемого автомата мотивировалось в духе «умозрительных экспериментов» Эдварда Ф. Мура [120]. Мур изучал конечные автоматы *с выходом*. В автоматах с выходом каждая пара (состояние, входная буква)

<sup>2</sup>Так, близкие идеи имелись в работах В. И. Левенштейна того же периода, см., например, [110].

определяет не только следующее состояние, в которое переходит автомат, но и символ некоторого выходного алфавита, показывающий отклик автомата на действие входной буквы. Автоматы с выходом рассматривались как математические модели устройств, работающих в дискретном режиме (таких, как компьютеры, например). Одна из естественных задач, возникающих при таком моделировании, состоит в определении неизвестного внутреннего состояния устройства по откликам этого устройства на различные воздействия. Мур [120] показал, что при определённых условиях состояние автомата однозначно определяется по откликам на подходящую последовательность входных сигналов (называемую *экспериментом*). Эксперименты Мура были адаптивными, т.е. в них входные сигналы выбирались с учётом откликов на предыдущие сигналы. Сеймур Гинзбург [76] рассмотрел более ограниченные эксперименты, которые он назвал *однородными*. Однородный эксперимент<sup>3</sup> – это просто фиксированная последовательность входных сигналов, т.е. слово над входным алфавитом; таким образом, в экспериментах Гинзбурга отклики использовались только по завершении эксперимента, чтобы установить состояние, в которое перешёл автомат. После этого оставался всего один шаг до постановки той задачи, которой занялся Черни, а именно, задачи об определении состояния автомата без какого-либо использования откликов. Отметим, что такая задача отнюдь не является искусственной – имеется много практических ситуаций, когда наблюдать отклики управляемой системы технически сложно или вообще невозможно.

В диссертации Лю [112] идея синхронизации мотивировалась тремя задачами. Первая – это задача о приведении автомата, текущее состояние которого неизвестно, к некоторому заданному состоянию, т.е. та же задача, которая мотивировала и Черни. Вторая задача – вариант первой, когда имеется несколько копий данного автомата, первоначально находящихся в разных состояниях, и нужно добиться синхронной работы всех этих копий. Третья задача связана с теорией кодирования – Лю показывает, как синхронизируемые автоматы порождают коды, способные восстановить синхронизацию между отправителем и получателем закодированного сообщения после сбоя в канале передачи информации. Связь синхронизируемых автоматов с кодами действительно имеет первостепенное значение, и мы отдельно обсудим её в следующем разделе.

Леммель в [108] использует сходную мотивацию: он, как и Черни, упоминает эксперименты Гинзбурга (со ссылкой на [77]) и, как и Лю, увязывает «перезагружаемые машины» с кодами, в частности, с так называемыми эргодическими кодами, рассматривавшимися Марселем-Полем Шютценберже [160]. Кроме того, Леммель проводит аналогию между понятием синхронизируемости и концепцией эргодичности в физике.

Неудивительно, что синхронизируемые автоматы были независимо и одновременно (можно сказать синхронно) придуманы несколькими исследователями: понятие было очень естественно само по себе и хорошо вписывалось в те направления, которое были модны в теории автоматов начала 1960-х гг. В неявной форме феномен синхронизируемости возникал в отдельных исследованиях по автоматам уже с середины 1950-х гг. В качестве примера приведём автомат из классической книги У. Росса Эшби «Введение в кибернетику» [17], см. с.60–61 оригинала или с.92–93 русского перевода. Эшби обсуждает головоломку, в которой нужно утихомирить Певуна и Хохотуна, двух беспокойных духов, обитающих в особняке с привидениями. Каждый из духов может либо шуметь, либо молчать, и их поведение зависит от комбинации двух возможных действий: игры на органе и сжигания ладана. При подходящем выборе обозначений ситуация описывается ДКА с четырьмя состояниями и четырьмя входными буквами, изображенным на рис. 2. Здесь 00 обозначает состояние, когда Певун и Хохотун молчат, 01 обозначает

---

<sup>3</sup>После выхода работы [75] за этим понятием закрепилось имя *homing sequence*; в русскоязычной литературе – *установочная последовательность*.

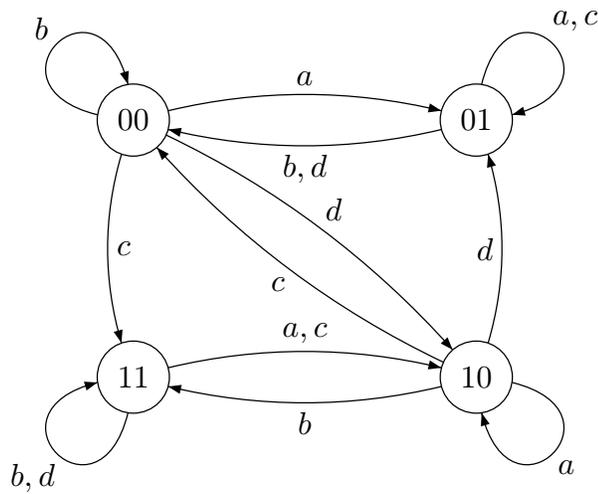


Рис. 2: Автомат Эшби

состояние, когда Певун молчит, а Хохотун хохочет, и т.д. Буква  $a$  обозначает переход, который происходит, когда на органе не играют и ладан не зажигают,  $b$  – переход, вызванный игрой на органе без поджигания ладана, и т.д. Чтобы оба духа умолкли, нужно перевести автомат на рис. 2 в состояние 00. В книге Эшби эта головоломка решена при дополнительном предположении, что оба духа активны, т.е. автомат находится в состоянии 11; предложенное решение описывается словом  $acb$ . Однако, как читатель может легко убедиться, автомат Эшби синхронизируем,<sup>4</sup> и слово  $acb$  синхронизирует его к состоянию 00. Поэтому дополнительное предположение несущественно, и применение соответствующей последовательности действий заставит Певуна и Хохотуна замолчать при любом их первоначальном поведении!

С 1960-х гг. понятие синхронизируемого автомата часто переоткрывалось. Одной из причин этого было то, что работы первопроходцев [44, 107, 108, 112] были труднодоступны. Статья Черни [44] была опубликована на словацком языке в местном журнале. Диссертация Лю [112] целиком не публиковалась; часть её вошла в статью [111], но результаты по синхронизации Лю в эту статью, к сожалению, не включил. Из-за этого его вклад в теорию синхронизируемых автоматов не стал широко известным и в конечном итоге был надолго забыт.<sup>5</sup> Отчёты Леммеля [107, 108], как и последующий отчет Леммеля и Бьюлы Раднер [109], выполнявшиеся по контракту армейского ведомства США, не были предназначены для широкого распространения (хотя и не были секретными).

Более концептуальной причиной переоткрытия синхронизируемых автоматов было то, что время от времени они неожиданно «всплывали» в разнообразных областях математики, информатики и технических наук, весьма далёких от тех задач, которые мотивировали [44, 107, 108, 112]. Мы приведем два примера такого рода в §1.5 и §1.6, после краткого обсуждения роли синхронизируемых автоматов в теории кодирования. В рамках этой теории синхронизация автоматов изучалась уже с середины 1950-х гг. (см., например, уже цитированные работы Шютценберже [160] и Леммеля [107]), т.е. до того, как было дано общепринятое теперь определение синхронизируемого автомата.

<sup>4</sup>Из книги не вполне ясно, осознавал ли Эшби эту «фишку» своего автомата.

<sup>5</sup>Это, впрочем, не помешало выдающейся карьере Лю как учёного и научного администратора, см. статью Википедии о нём, [https://en.wikipedia.org/wiki/Chung\\_Laung\\_Liu](https://en.wikipedia.org/wiki/Chung_Laung_Liu).

## 1.4 Синхронизируемые автоматы и коды

Здесь мы обсудим взаимосвязи между синхронизируемыми автоматами и кодами на идейном уровне, не предполагая никаких предварительных знаний из теории кодов.<sup>6</sup>

Говоря о кодировании, мы имеем в виду следующее. Имеются данные, представленные длинным словом над некоторым алфавитом  $\Theta$ . Хорошим примером может служить какой-нибудь текст на естественном языке, скажем, роман Марселя Пруста «В поисках утраченного времени», французский оригинал которого содержит примерно 9 609 000 знаков (каждая буква, знак препинания и пробел считаются за один знак). Для хранения или передачи таких данных их кодируют, заменяя каждый знак алфавита  $\Theta$  словом над некоторым другим алфавитом  $\Sigma$ , обычно над *двоичным алфавитом*  $\{0, 1\}$ . Итак, *кодирование* – это отображение  $\chi: \Theta \rightarrow \Sigma^+$ , такое, что продолжение  $\chi$  на  $\Theta^+$  инъективно, т.е. каждое слово  $w \in \Theta^+$  однозначно определяется тем словом из  $\Sigma^+$ , которое получится, если последовательно заменять каждую букву  $a$  слова  $w$  на слово  $a\chi$ . Множество  $\Theta\chi$  называется *кодом*, а его элементы – *кодowymi словами*.

Кодирование  $\chi: \Theta \rightarrow \{0, 1\}$  называют *двоичным*. Любое двоичное кодирование алфавита  $\Theta$  с помощью слов постоянной длины (таких, как ANSI-коды, например) требует  $\lceil \log_2 |\Theta| \rceil$  бит для каждой буквы. Следовательно, чтобы закодировать данное слово  $w \in \Theta^+$ , потребуется  $|w| \cdot \lceil \log_2 |\Theta| \rceil$  бит. Во многих случаях, однако, буквы неравноправны – одни встречаются чаще, другие реже. (Скажем, в упомянутом романе Пруста «е» встречается примерно вдвое чаще, чем «а», а частота появления «к» составляет менее 0,9% от частоты появления «l».) Поэтому можно сэкономить память (в случае хранения данных) и/или время (в случае передачи данных), если отказаться от равенства длин кодовых слов и кодировать буквы из  $\Theta$ , которые встречаются чаще, более короткими словами над  $\Sigma$ . Эта простая идея использовалась ещё в 19-м веке в телеграфном коде Сэмюэля Морзе: так, «е», самая распространённая буква английского языка, имеет самый короткий код Морзе (одну точку).

С другой стороны, при использовании кодовых слов различной длины может стать нетривиальной задача декодирования, т.е. восстановления слова  $w \in \Theta^+$  по последовательности кодов его букв. Приведем простейший пример. Нетрудно проверить, что отображение  $\chi: \{a, b\} \rightarrow \{0, 1\}^+$ , определённое правилом  $a\chi := 0$ ,  $b\chi := 01$ , будет кодированием. Допустим, что в декодер поступает последовательность из 0 и 1, начинающаяся с 0. Прочитав первый бит, декодер ещё не сможет определить, является ли данный 0 кодом буквы  $a$  или первым битом кода буквы  $b$ , и будет должен отложить решение до прочтения следующего бита, а прочитанный бит придётся до этого запомнить. При более сложном кодировании такая задержка декодирования может быть значительной и по времени, и по памяти, требуемой для хранения промежуточных данных.

Есть, однако, важный класс кодов, допускающих декодирование «на лету», т.е. с той же скоростью, с которой поступают кодовые слова. Скажем, что слово  $x \in \Sigma^*$  называется *префиксом* слова  $y \in \Sigma^+$ , если  $y$  можно представить, как  $y = xz$  для некоторого  $z \in \Sigma^+$ . (Заметим, что в соответствии в этим определением  $\varepsilon$  будет префиксом слова  $y$ , а вот само слово не является своим префиксом.) Множество  $X \subset \Sigma^+$  называется *префиксным кодом*, если слова из  $X$  не являются префиксами друг друга; кодирование  $\chi: \Theta \rightarrow \Sigma^+$  называют *префиксным*, если  $\Theta\chi$  – префиксный код.

При префиксном кодировании процедура декодирования очень проста: декодер читает поступающий поток символов из  $\Sigma$  слева направо, пока не прочтёт кодовое слово. Это слово не может быть префиксом никакого другого кодового слова в силу определения префиксного кода, поэтому прочитанное слово сразу заменяется соответствующей

---

<sup>6</sup>Обстоятельное обсуждение этих глубоких и полезных взаимосвязей можно найти в монографии Жана Берстеля, Доминика Перрэнна и Кристофа Рейтенауэра [29].

буквой из  $\Theta$  и удаляется из потока, после чего процесс повторяется. Классический результат теории кодирования (теорема Крафта–Макмиллана, см. [29, теорема 2.4.12]) гарантирует, что для любого кода найдётся префиксный код над тем же алфавитом с тем же набором длин кодовых слов. Отсюда следует, что наилучшее сжатие данных, возможное за счёт использования кодов переменной длины, реализуемо префиксным кодом. Это – наряду с отсутствием задержек декодирования – объясняет, почему в приложениях, как правило, встречаются именно префиксные коды.

Описанный выше алгоритм декодирования изящно реализуется конечным автоматом, вообще говоря, частичным.<sup>7</sup> Мы обсудим соответствующее построение во второй части статьи; сейчас же ограничимся специальным случаем конечных максимальных префиксных кодов. (Префиксный код  $X \subset \Sigma^+$  называется *максимальным*, если  $X$  не содержится ни в каком другом префиксном коде над  $\Sigma$ .) Если  $X$  – такой код, то в качестве его декодера используется ДКА  $\mathcal{A}_X = \langle Q, \Sigma \rangle$ , где  $Q$  – множество всех префиксов слов из  $X$ , а функция перехода определена так:

$$q \cdot a = \begin{cases} qa, & \text{если } qa \text{ – префикс слова из } X, \\ \varepsilon, & \text{если } qa \in X. \end{cases}$$

Автомат  $\mathcal{A}_X$  начинает работу в состоянии  $\varepsilon$  и разделяет поступающий поток символов из  $\Sigma$  на кодовые слова, соответствующие возвратам в это состояние. Рисунок 3 изображает ДКА  $\mathcal{A}_C$  для максимального префиксного кода

$$C := \{000, 0010, 0011, 010, 0110, 0111, 10, 110, 111\}. \quad (1)$$

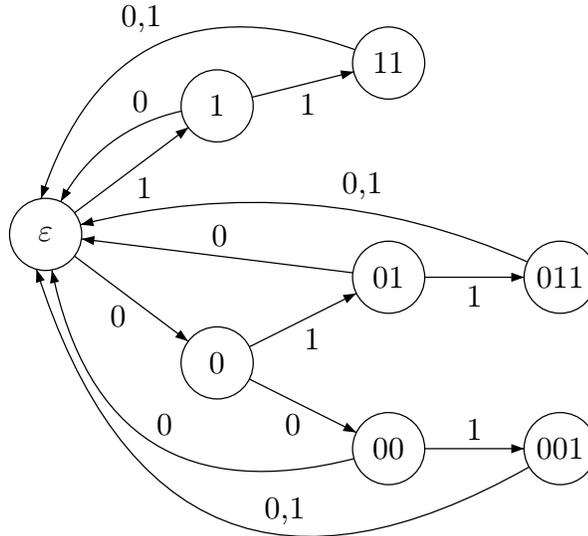


Рис. 3: Декодер кода  $\{000, 0010, 0011, 010, 0110, 0111, 10, 110, 111\}$

Максимальный префиксный код  $X$  над  $\Sigma$  называют *синхронизированным*, если существует такое слово  $z \in \Sigma^+$ , что для любого  $y \in \Sigma^*$  слово  $yz$  можно разложить в произведение слов из  $X$ . Такое  $z$  называется *синхронизирующим словом* для  $X$ . Преимущество синхронизированных кодов состоит в том, что при сбое синхронизации между декодером и кодером, вызванном ошибками передачи, достаточно передать синхронизирующее слово, и последующие символы будут декодированы верно. Более

<sup>7</sup> Частичные детерминированные автоматы отличаются от полных только тем, что их функция переходов может быть неопределена для некоторых пар (состояние, входная буква).

того, поскольку вероятность того, что слово  $x \in \Sigma^*$  содержит фиксированный блок  $z$ , стремится к 1 с ростом длины  $x$ , синхронизированные коды восстанавливают синхронизацию сами по себе после передачи достаточного количества символов. (Как показано в [39], последнее свойство фактически характеризует синхронизированные коды.)

Проиллюстрируем это понятие на примере кода  $C$  из (1). Легко проверить, что каждое из слов 010, 011110, 01111110, ... является синхронизирующим для  $C$ . Предположим, что было отправлено кодовое слово 000, но первый бит искажился при передаче и в результате было получено слово 100. Декодер, который не может знать об ошибке, интерпретирует 10 как кодовое слово и, таким образом, теряет синхронизацию, поскольку станет обрабатывать следующий бит как часть другого кодового слова. Кажалось бы, с этого момента всё дальнейшее декодирование будет неверным, однако это не так. С высокой вероятностью потеря синхронизации не распространится слишком далеко: синхронизация восстановится, как только декодер встретит в принимаемом потоке один из блоков 010, 011110, 01111110, ... . Несколько примеров указаны в табл. 2. (Вертикальные линии в табл. 2 не являются частью передаваемых данных и приведены только для удобства, чтобы было видно разделение каждого потока на кодовые слова.) Кодовые слова, выделенные жирным шрифтом, указывают блоки, начиная с которых синхронизация восстанавливается.

Таблица 2: Восстановление синхронизации при использовании кода  $C$  из (1)

Передано	000   0010   <b>0111</b>   ...
Получено	10   000   10   <b>0111</b>   ...
Передано	000   0111   110   0011   000   10   <b>110</b>   ...
Получено	10   0011   111   000   110   0010   <b>110</b>   ...
Передано	000   000   111   <b>10</b>   ...
Получено	10   000   0111   <b>10</b>   ...

Сходство терминологии намекает на родство синхронизированных кодов и синхронизируемых автоматов, и действительно, эти два понятия тесно связаны. В том специальном случае, который мы рассматриваем сейчас, их связь носит особенно непосредственный характер, как показывает следующий легко проверяемый результат.

**Предложение 1.1.** *Максимальный префиксный код синхронизирован тогда и только тогда, когда синхронизируем его декодер. При этом слова, синхронизирующие код  $X$ , – это в точности слова, которые синхронизируют ДКА  $\mathcal{A}_X$  к состоянию  $\varepsilon$ .*

Декодеры синхронизированных кодов образуют важный подкласс класса синхронизируемых автоматов, и изучению этого подкласса до сих пор уделяется значительное внимание, см., например, недавние публикации [28, 30–33, 87, 154].

## 1.5 Переоткрытие в промышленной механике

Задачи синхронизации естественным образом возникают в промышленной механике при разработке устройств, осуществляющих различные манипуляции с деталями (подача, сортировка, упаковка и т.п.). В русле этой проблематики понятие синхронизируемого автомата было независимо переоткрыто в середине 1980-х гг. Баласом Натараджаном [121, 122], показавшим, как такие автоматы могут быть использованы для создания чисто механических ориентаторов плоских многоугольных деталей. Мы объясним суть этого подхода с помощью иллюстративного примера, взятого из [9].

Пусть деталь некоторого устройства имеет форму, показанную на рис. 4. На точку

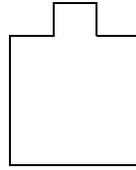


Рис. 4: Многоугольная деталь

сборки устройства такие детали должны поступать в определённой ориентации. Для простоты предположим, что возможны только четыре исходные ориентации деталей, а именно те, которые изображены на рис. 5.



Рис. 5: Четыре возможные ориентации

Далее, предположим, что к моменту сборки деталь должна быть сориентирована выступом влево (вторая слева ориентация на рис. 5). Ориентатор должен сориентировать деталь нужным образом при любом её первоначальном положении.

Есть разные подходы к конструированию ориентаторов, но на практике предпочтительны простые и надёжные механические ориентаторы, которые не требуют сенсорных датчиков, машинного зрения и т.п. В нашем примере цель достигается следующим образом. Поместим детали на ленту транспортера, которая доставит их к месту сборки, а вдоль ленты разместим ряд пассивных препятствий. Понадобятся препятствия двух типов: длинные и короткие. Длина длинного препятствия выбирается так, чтобы любая деталь на ленте наткнулась на это препятствие своим нижним правым углом (мы предполагаем, что лента движется слева направо). При этом деталь, увлекаемая лентой, поворачивается на  $90^\circ$  по часовой стрелке, как показано на рис. 6. Короткое препят-

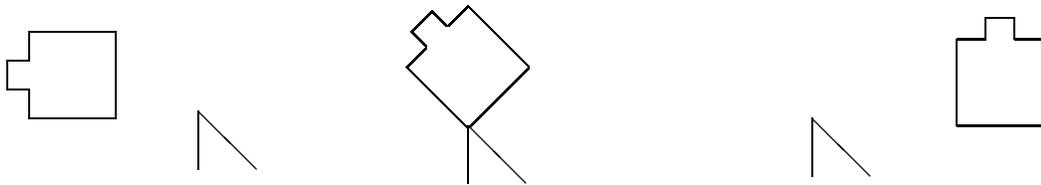


Рис. 6: Действие длинного препятствия

ствие оказывает тот же эффект, когда деталь ориентирована выступом вниз (первая слева ориентация на рис. 5) и препятствие зацепляет выступ; в противном случае оно не касается детали, и та проходит мимо препятствия, не меняя ориентацию.

Схема на рис. 7 показывает, как препятствия влияют на ориентацию детали. Надеемся, что читатель узнал здесь автомат  $\mathcal{C}_4$  (ср. с рис. 1). Поскольку слово *abbbabba* синхронизирует  $\mathcal{C}_4$  к состоянию 1 (см. обсуждение примера 1.1 в §1.2), при прохождении последовательности препятствий

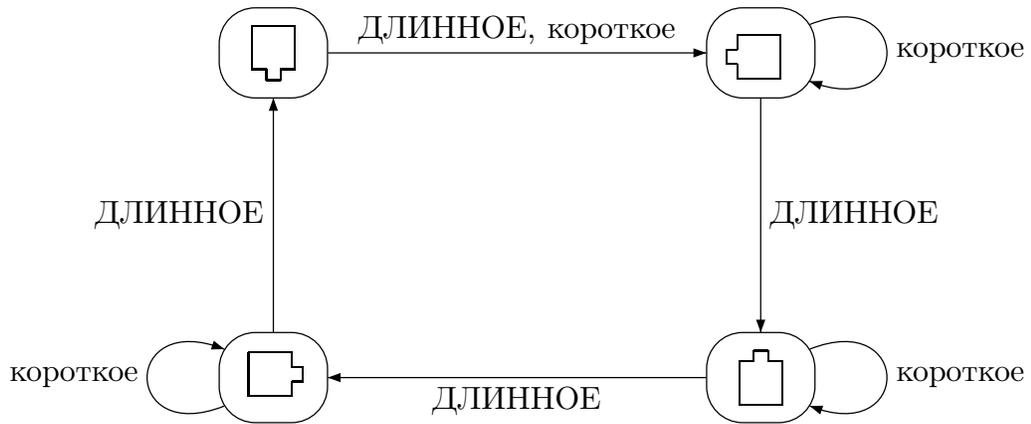


Рис. 7: Схема действия препятствий

короткое–ДЛИННОЕ–ДЛИННОЕ–ДЛИННОЕ–короткое–ДЛИННОЕ–ДЛИННОЕ–  
 ДЛИННОЕ–короткое

любая деталь примет нужную ориентацию.

С 1990-х гг. разработка устройств подачи и сортировки на основе синхронизируемых автоматов развилась в обширное исследовательское направление, но справедливо сказать, что большинство публикаций в этой области касается технических аспектов. Однако некоторые из этих публикаций содержат и важные для теории результаты, см. например, [47, 63, 78, 137].

### 1.6 Переоткрытие в теории подстановочных систем

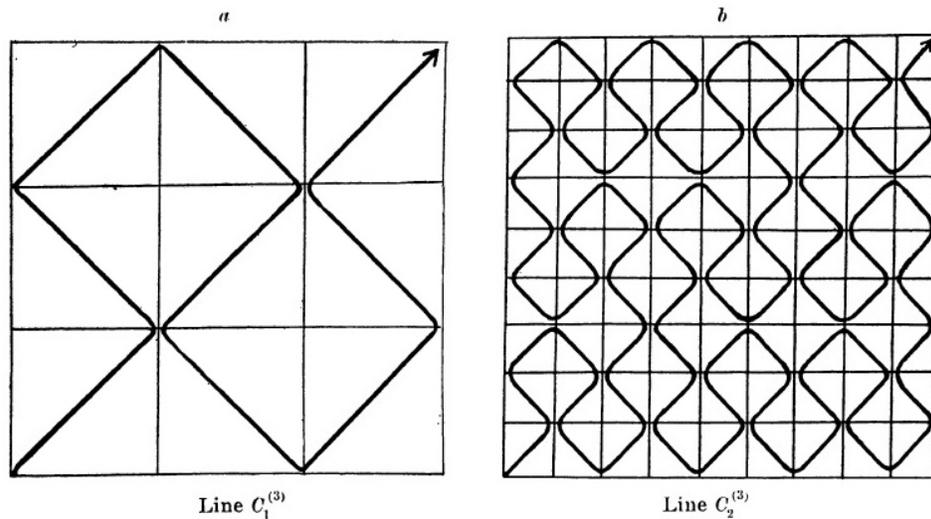


Рис. 8: Иллюстрация из статьи Мура [119]

В 1990-х гг. синхронизируемые автоматы были ещё раз переоткрыты в теории подстановок. Подстановка на конечном алфавите  $X$  – это отображение  $\sigma: X \rightarrow X^+$ . Тео-

рия подстановок изучает динамические аспекты итераций подстановок на множествах самой разной природы, см. коллективную монографию [134]. Говорят, что  $\sigma : X \rightarrow X^+$  – *подстановка постоянной длины*, если длины слов  $x\sigma$  одинаковы для всех  $x \in X$ . Такие подстановки используются, например, для порождения фрактальных объектов; в качестве иллюстрации приведём изящное построение кривой Пеано (непрерывной кривой, заполняющей единичный квадрат), принадлежащее Элиакиму Муру [119].

Идея Мура понятна из рис. 8 [119, рис. 3]. Для удобства обозначения направлений повернём единичный квадрат из левой части рис. 8 на  $45^\circ$  по часовой стрелке, как показано на рис. 9. Видно, что ломаная, с которой Мур начинает построение, состоит из 9 звеньев (обозначенных на рис. 9 стрелками). Эти звенья проходятся с левого угла квадрата в последовательности ENESWSENE, где буквы E, N, W и S обозначают стороны света. На каждой последующей итерации каждая стрелка заменяется аналогичной 9-звенной ломаной, т.е. производится подстановка постоянной длины 9 по правилу

$$E \mapsto ENESWSENE, N \mapsto NWNESENWN, W \mapsto WSWNENWSW, S \mapsto SESWNWSES.$$

Искомая кривая Пеано есть предел построенной так последовательности ломаных.

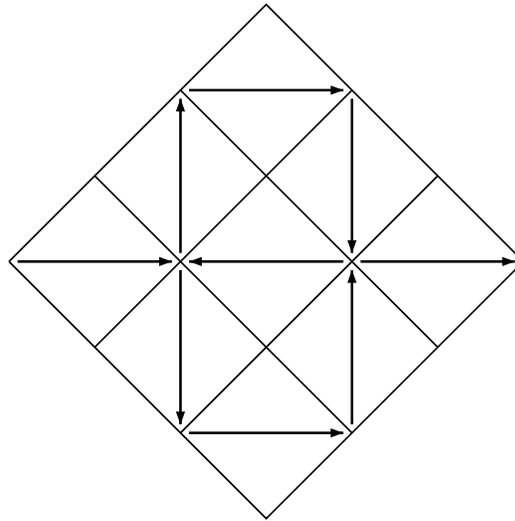


Рис. 9: Начальный квадрат Мура, повернутый на  $45^\circ$  по часовой стрелке

Говорят, что подстановка постоянной длины  $\sigma : X \rightarrow X^+$  удовлетворяет *условию совпадения* (coincidence condition), если существуют такие натуральные  $m$  и  $k$ , что в  $m$ -й позиции слов  $x\sigma^k$  для всех  $x \in X$  стоит одна и та же буква. В качестве примера рассмотрим подстановку  $\tau$  на  $X := \{0, 1, 2\}$ , определённую правилом

$$0 \mapsto 11, 1 \mapsto 12, 2 \mapsto 20. \quad (2)$$

Вычисляя итерации  $\tau$  до  $\tau^4$  (см. табл. 3), мы видим, что  $\tau$  удовлетворяет условию совпадения с  $k = 4$ ,  $m = 7$ .

Фредерик Мишель Деккинг [58] показал, что условие совпадения играет ключевую роль для характеристики ряда важных свойств динамических систем, определяемых подстановками постоянной длины, см. обзор в главе 7 книги [134]. Для нас же условие совпадения интересно как очередная инкарнация синхронизируемости. Чтобы увидеть это, установим взаимно однозначное соответствие между ДКА и подстановками постоянной длины. ДКА  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$  с алфавитом  $\Sigma = \{a_1, \dots, a_\ell\}$  определяет подстановку

Таблица 3: Подстановка, удовлетворяющая условию совпадения

0	↦	11	↦	1212	↦	12201220	↦	1220201112202011
1	↦	12	↦	1220	↦	12202011	↦	1220201120111212
2	↦	20	↦	2011	↦	20111212	↦	2011121212201220

постоянной длины  $\ell$  на множестве  $Q$  по следующему правилу:  $q \in Q$  отображается в слово  $(q \cdot a_1) \cdots (q \cdot a_\ell) \in Q^+$ . Например, автомат  $\mathcal{C}_4$  (см. рис. 1) определяет подстановку

$$0 \mapsto 11, 1 \mapsto 12, 2 \mapsto 23, 3 \mapsto 30.$$

Обратно, каждая подстановка  $\sigma: X \rightarrow X^+$  такая, что  $|x\sigma| = \ell$  для всех  $x \in X$ , определяет ДКА с множеством состояний  $X$  и  $\ell$  входными буквами  $a_1, \dots, a_\ell$ , действующими на  $X$  по правилу:  $x \cdot a_i$  – это символ в  $i$ -й позиции слова  $x\sigma$ . Например, подстановка (2) определяет автомат, изображенный на рис. 10.

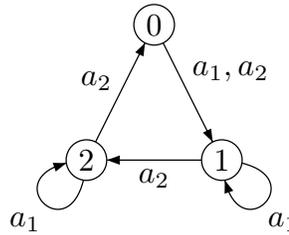


Рис. 10: Автомат, определяемый подстановкой (2)

Несложно видеть, что при указанном взаимно однозначном соответствии подстановки, удовлетворяющие условию совпадения, в точности соответствуют синхронизируемым автоматам. При этом номер итерации, на которой впервые появляется совпадение, равен минимальной длине синхронизирующего слова для соответствующего ДКА. Эти наблюдения были впервые сделаны, по-видимому, Дирком Фретлэ и Берндом Сингом в [70]; Фретлэ и Синг пользовались отличной от принятой здесь терминологией, поскольку не были знакомы с понятием синхронизируемого автомата (и переоткрыли некоторые результаты Черни [44]).

## 1.7 Алгебраические аспекты

Для ДКА  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$  каждое слово  $w \in \Sigma^*$  определяет преобразование  $[w]: Q \rightarrow Q$  по правилу  $[w]: q \mapsto q \cdot w$ . Обозначим совокупность всех таких преобразований через  $M(\mathcal{A})$ . Ясно, что множество  $M(\mathcal{A})$  всегда содержит тождественное преобразование  $[\varepsilon]$  и замкнуто относительно умножения преобразований: для любых слов  $u, v \in \Sigma^*$ , имеем  $[u][v] = [uv]$ , где  $uv$  – произведение  $u$  и  $v$  в  $\Sigma^*$ , т.е. результат приписывания  $v$  к  $u$ . Поэтому  $M(\mathcal{A})$  – подмоноид в моноиде  $\mathcal{T}(Q)$  всех преобразований множества  $Q$ ; он называется *моноидом переходов автомата  $\mathcal{A}$* . Если трактовать конечный автомат как вычислительное устройство, то моноид переходов можно мыслить себе как библиотеку программ этого устройства, содержащую описания всех доступных ему вычислений.

*Рангом преобразования* конечного множества называют число элементов в образе этого преобразования. Итак, преобразования ранга 1 – это в точности *константные*

преобразования, переводящие все элементы исходного множества в один и тот же элемент. Теперь понятно, как синхронизируемость ДКА выражается в терминах его моноида перехода: ДКА  $\mathcal{A}$  синхронизируем тогда и только тогда, когда в моноиде  $M(\mathcal{A})$  имеется преобразование ранга 1.

Возможность рассматривать феномен синхронизируемости как с комбинаторной, так и с алгебраической точки зрения нередко бывает полезной. В частности, алгебраический подход допускает естественную линейризацию, что позволяет задействовать арсенал линейной алгебры. А именно, пусть  $Q = \{q_1, \dots, q_n\}$ . Сопоставим состоянию  $q_i$  вектор  $[q_i] = (\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{n-i})$  пространства  $\mathbb{R}^n$  всех  $n$ -мерных строк над полем  $\mathbb{R}$  действительных чисел. Тогда вектора  $[q_1], \dots, [q_n]$  образуют базис в  $\mathbb{R}^n$ , и для каждого слова  $w \in \Sigma^*$  преобразование  $[w] \in M(\mathcal{A})$  однозначно продолжается до линейного оператора пространства  $\mathbb{R}^n$ . За матрицей этого оператора в указанном базисе мы сохраним обозначение  $[w]$ . Таким образом,  $M(\mathcal{A})$  превращается в моноид  $n \times n$ -матриц, а синхронизируемость автомата  $\mathcal{A}$  означает в точности наличие в этом моноиде матрицы, в которой один столбец состоит из единиц, а все остальные столбцы нулевые. Такая линейризация эффективно использовалась в доказательствах нескольких ключевых результатов, которые обсуждаются в §3.4.

Описанная линейризация есть не что иное как линейное представление моноида  $M(\mathcal{A})$  над полем  $\mathbb{R}$ , которое при необходимости можно считать и представлением над полем комплексных чисел. Изучение синхронизируемых автоматов средствами теории представлений принесло ряд содержательных результатов. Пионером в этом направлении был, по-видимому, И. К. Рысцов [151]; из других публикаций, в которых к синхронизируемым автоматам прилагались методы теории представлений, можно отметить статьи Жоржи Алмейды, Эмануэле Родаро, Бенджамина Стейнберга и их соавторов [3–5, 16, 141, 167].

С моноидами преобразований, а точнее с их группами обратимых элементов, связано одно направление теории синхронизируемых автоматов, ставшее в последнее время популярным среди специалистов по группам перестановок. Обозначим через  $\mathcal{S}(Q)$  симметрическую группу на множестве  $Q$ , т.е. группу всех перестановок  $Q$ . Подгруппу  $\mathcal{G}$  группы  $\mathcal{S}(Q)$  называют *синхронизирующей*, если для любого преобразования  $\theta \in \mathcal{T}(Q) \setminus \mathcal{S}(Q)$  подмоноид, порождённый  $\mathcal{G}$  и  $\theta$ , содержит преобразование ранга 1. На языке автоматов это означает, что наличие  $\mathcal{G}$  в качестве подгруппы в группе обратимых элементов моноида  $M(\mathcal{A})$  гарантирует синхронизируемость ДКА  $\mathcal{A}$  при условии, что среди входных букв этого автомата хотя бы одна не действует как перестановка множества состояний. Примерами синхронизирующих групп являются сама симметрическая группа, знакопеременная группа и, вообще, любая дважды транзитивная группа перестановок.<sup>8</sup> Это легко следует из критерия синхронизируемости, см. предложение 2.1 ниже. С другой стороны, нетрудно видеть, что синхронизирующая подгруппа в  $\mathcal{S}(Q)$  должна быть *примитивной*, т.е. не может сохранять никакое разбиение множества  $Q$  на несколько неоднородных блоков. Итак, понятие синхронизирующей группы выделяет подкласс класса примитивных групп, содержащий все дважды транзитивные группы. Задача о точной характеристике этого подкласса оказалась весьма содержательной, обнаружила ряд глубоких связей с теорией классических комбинаторных конфигураций (таких, как латинские квадраты, системы Штейнера, матрицы Адамара и т.п.) и породила большое число публикаций. Обзор работ, связанных со (всё ещё не завершённой) классификацией синхронизирующих групп, не включен в настоящую

<sup>8</sup>Напомним, что группа  $\mathcal{G}$  перестановок множества  $Q$  называется *дважды транзитивной*, если для любых  $p, q, r, s \in Q$  таких, что  $p \neq q$  и  $r \neq s$ , найдётся перестановка  $\gamma \in \mathcal{G}$  такая, что  $p\gamma = r$  и  $q\gamma = s$ .

статью, поскольку этот материал обстоятельно представлен в недавнем мемуаре Жоау Араужу, Питера Камерона и Бенджамина Стейнберга [15].

Завершая этот короткий экскурс в алгебру, заметим ещё, что и сами автоматы можно естественным образом интерпретировать как алгебраические структуры. Напомним, что *алгеброй типа*  $(n_1, n_2, \dots)$  с носителем  $Q$  называется непустое множество  $Q$ , оснащённое операциями  $f_1: \underbrace{Q \times Q \times \dots \times Q}_{n_1} \rightarrow Q$ ,  $f_2: \underbrace{Q \times Q \times \dots \times Q}_{n_2} \rightarrow Q$ ,  $\dots$ . Алгебра

$(Q; f_1, f_2, \dots)$  типа  $(1, 1, \dots)$  называется *унаром*. Понятно, что ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  – не что иное, как унар с носителем  $Q$ , на котором для каждой буквы  $a \in \Sigma$  определена унарная операция по правилу  $[a]: q \mapsto q \cdot a$ . Такая точка зрения позволяет применять к автоматам стандартные алгебраические понятия подалгебры (*подаавтомата*), *конгруэнции*, *фактор-автомата*, и т.п.; мы пользуемся ими ниже. Отметим, что класс синхронизируемых ДКА замкнут относительно взятия подавтоматов и фактор-автоматов; при этом каждое слово, синхронизирующее данный ДКА, будет синхронизировать все его подавтоматы и фактор-автоматы.

При интерпретации автоматов как унаров понятие синхронизируемого автомата достаточно естественно переводится на язык алгебры. Напомним, что *унарный терм* – это выражение  $\tau$  вида  $x \cdot w$ , где  $x$  – переменная, а  $w$  – слово над алфавитом  $\Sigma$ . *Тождество унаров* – это формальное равенство двух унарных термов. ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , рассматриваемый как унар, *удовлетворяет* тождеству  $\tau_1 = \tau_2$ , если термы  $\tau_1$  и  $\tau_2$  принимают одно и то же значение при каждой интерпретации их переменных в множестве  $Q$ . Тождества унаров могут иметь вид  $x \cdot u = x \cdot v$  (*однородные* тождества) или  $x \cdot u = y \cdot v$ , где  $x \neq y$  (*неоднородные* тождества). Легко понять, что ДКА синхронизируем тогда и только тогда, когда он удовлетворяет неоднородному тождеству. Действительно, если слово  $w$  синхронизирует ДКА  $\mathcal{A}$ , то  $\mathcal{A}$  удовлетворяет неоднородному тождеству  $x \cdot w = y \cdot w$ . Обратно, пусть  $\mathcal{A}$  удовлетворяет какому-то неоднородному тождеству  $x \cdot u = y \cdot v$ . Заменяя в нём  $x$  на  $y$ , получим тождество  $y \cdot u = y \cdot v$ , и по транзитивности в  $\mathcal{A}$  выполняется и тождество  $x \cdot u = y \cdot u$ . Последнее тождество означает, что слово  $u$  синхронизирует  $\mathcal{A}$ . Итак, синхронизируемые автоматы при желании можно изучать в рамках эквациональной логики унаров. Пока нельзя сказать, чтобы такой подход принес сколько-нибудь заметную пользу для понимания комбинаторной сути синхронизируемости; тем не менее, в его рамках выполнено немало публикаций, см. обзор Стояна Богдановича, Балажа Имре, Мирослава Чирича и Татьяны Петкович [34].

## 1.8 Другие приложения и связи

Если читатель добрался до этого раздела, можно надеяться, что он уже убедился в том, что понятие синхронизируемого автомата хорошо мотивировано, а области, в которых возникают синхронизируемые автоматы, весьма разнообразны. Здесь перечислены ещё несколько примеров задач, так или иначе связанных с синхронизацией ДКА. То, что эти задачи лишь перечисляются, а не обсуждаются в деталях в стиле предыдущих разделов, совсем не означает, что они менее важны или менее увлекательны!

Синхронизирующие слова давно применяются при тестировании схем и протоколов, см. статьи [36, 54, 92, 130, 139] в качестве типовых примеров исследований в этой области, а также обзор [159]. Когда испытываемая система подверглась очередному тесту, то перед следующим испытанием её нужно вернуть к первоначальному состоянию – в этом и состоит роль синхронизирующих слов. В аналогичной роли синхронизирующие слова используются в задачах планирования движения в роботике, когда «заблудившийся» робот должен вернуться в известную позицию, см., например, [170].

На противоположном, теоретическом конце спектра находятся связи синхронизируемых автоматов с алгеброй и логикой. Соотношения между синхронизируемыми автоматами и функциями многозначной логики активно изучались в школе Арто Саломаа, см. его собственные статьи [155–158] и обзор [117]. Некоторые задачи теории конечных полугрупп [6, 131] мотивировали интерес к *универсальным синхронизирующим словам*, т.е. словам, синхронизирующим все синхронизируемые автоматы с данным числом состояний над фиксированным алфавитом. Обзор результатов по универсальной синхронизации на середину 2000-х гг. см. в [8, 48]; среди более поздних публикаций в этом направлении см. [49–52].

В заключение раздела проясним одно недоразумение. В литературе нередко можно встретить заявления о применениях синхронизируемых автоматов в биоинформатике, подкрепляемые ссылками на статьи [20, 21]. В этих, без сомнения очень интересных и важных работах сообщаются результаты успешных экспериментов по созданию автоматов молекулярного размера, в которых и состояния, и входные буквы представляют собой короткие цепочки ДНК. Так, в [20] описан «автоматный бульон» – раствор, каждый миллилитр которого содержит  $3 \cdot 10^{12}$  идентичных микроавтоматов. Идея о том, что в перспективе такого рода системами молекулярных автоматов можно будет управлять, «приправляя» соответствующий бульон цепочками ДНК, кодирующими синхронизирующие слова, была высказана как гипотеза в обзоре [182]. Однако в [182] никоим образом не утверждалось, что эта высказанная с эпитетом ‘yet imaginary’ идея уже реализована в [20, 21] или в каких-либо других реальных экспериментах. (На самом деле молекулярные автоматы, построенные в [20, 21], не являются синхронизируемыми!)

Отметим, что содержательные связи между синхронизируемыми автоматами и биоинформатикой существуют, см., например, [35]; они будут обсуждаться во второй части статьи в параграфе, посвящённом синхронизации частичных детерминированных автоматов.

## 2 Алгоритмические и сложностные аспекты

### 2.1 Проверка автомата на синхронизируемость

Понятно, что не все ДКА синхронизируемы, и потому возникает вопрос, как узнать, синхронизируем ли данный автомат. Ответ может быть получен с помощью конструкции автомата подмножеств из классической статьи Майкла О. Рабина и Даны Скотта [135]. В учебниках по теории автоматов эта конструкция обычно используется для детерминизации недетерминированных автоматов (она даёт ДКА, распознающий тот же язык, что и данный недетерминированный автомат), но здесь мы применим её к детерминированным автоматам.

Обозначим через  $\mathcal{P}(Q)$  множество всех непустых подмножеств множества  $Q$ . Функция переходов ДКА  $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$  распространяется до функции  $\mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$ , за которой мы сохраним обозначение  $\delta$ , по правилу  $\delta(P, a) := \{\delta(p, a) : p \in P\}$  для всех множества  $P \in \mathcal{P}(Q)$  и всех букв  $a \in \Sigma$ . Возникает ДКА  $\mathcal{P}(\mathcal{A}) = \langle \mathcal{P}(Q), \Sigma, \delta \rangle$ , который называется *автоматом подмножеств* автомата  $\mathcal{A}$ . Рисунок 11 изображает автомат подмножеств автомата  $\mathcal{C}_4$ , см. рис. 1.

Применяя к автомату подмножеств  $\mathcal{P}(\mathcal{A}) = \langle \mathcal{P}(Q), \Sigma, \delta \rangle$  соглашение, введённое в §1.1, будем писать  $P.v$  вместо  $\delta(P, v)$  (здесь  $P$  – непустое подмножество в  $Q$ , а  $v$  – слово над  $\Sigma$ ). Тогда ясно, что слово  $w \in \Sigma^*$  синхронизирует ДКА  $\mathcal{A}$ , если и только если множество  $Q.w$  одноэлементно. Чтобы выразить последнее свойство в терминах носителя автомата  $\mathcal{P}(\mathcal{A})$ , напомним некоторые понятия из теории графов.

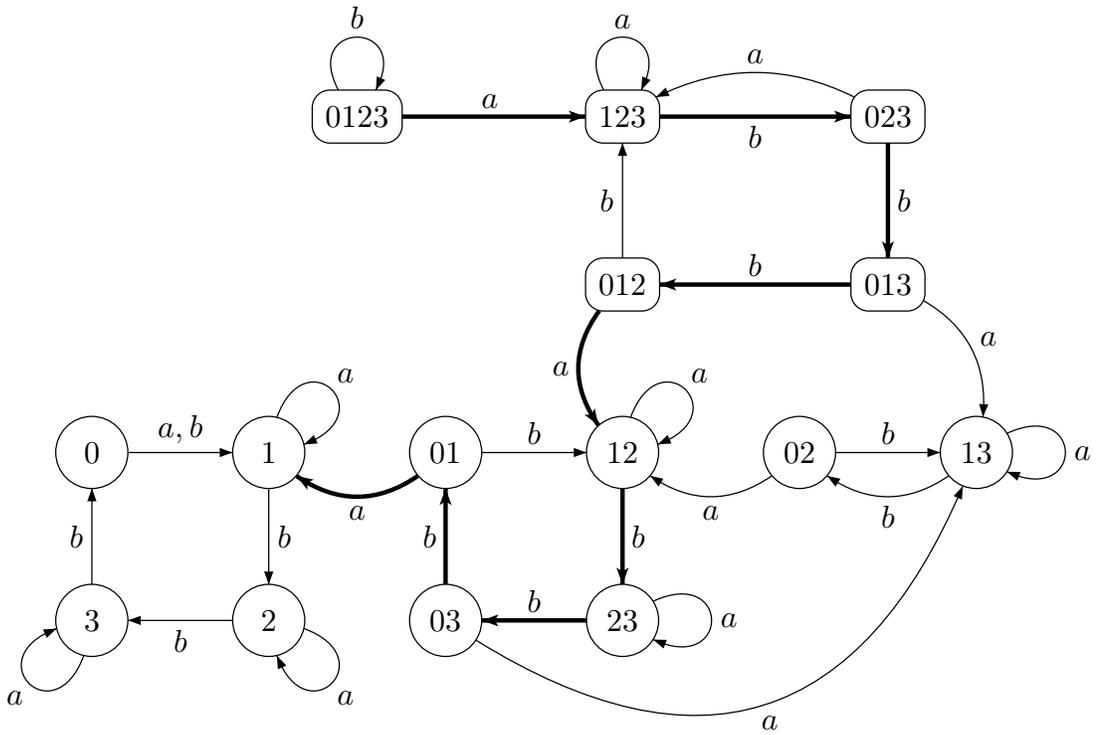


Рис. 11: Автомат подмножеств  $\mathcal{P}(\mathcal{C}_4)$

В соответствии с определением из §1.1, граф – это четвёрка множеств и отображений  $\langle V, E, h, t \rangle$ , где  $V$  и  $E$  – это, соответственно, множества вершин и ребер, а отображения  $h, t: E \rightarrow V$  сопоставляют ребру его начало и, соответственно, конец. Два ребра  $e, e' \in E$  называются *последовательными*, если  $t(e) = h(e')$ . *Путь* в графе – это слово над множеством  $E$  его ребер, в котором любые два соседних ребра являются последовательными; длина слова называется *длиной* пути. В частности, пустое слово над  $E$  – это путь (длины 0), называемый *пустым путём*. Говорят, что путь *начинается* в начале его первого ребра и *заканчивается* в конце его последнего ребра. Условимся, что пустой путь может начинаться в любой вершине. Если путь начинается в вершине  $v$  и заканчивается в вершине  $v'$ , мы называем его *путём из  $v$  в  $v'$* . Вершина  $v'$  называется *достижимой* из вершины  $v$ , если существует путь из  $v$  в  $v'$ .

Теперь условие  $|Q \cdot w| = 1$  можно выразить следующим образом: слово  $w$  – это последовательность меток, которые читаются в графе автомата подмножеств  $\mathcal{P}(\mathcal{A})$  вдоль пути, который начинается в  $Q$  и заканчивается одноэлементным подмножеством. Например, метки выделенного жирным шрифтом пути на рис. 11 образуют уже знакомое нам синхронизирующее слово автомата  $\mathcal{C}_4$ . При этом легко видеть, что более короткого пути из  $Q = \{0, 1, 2, 3\}$  в одноэлементное подмножество нет, откуда у  $\mathcal{C}_4$  нет более короткого синхронизирующего слова, как и утверждалось при обсуждении примера 1.1.

Таким образом, вопрос о том, синхронизируем ли данный ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , сводится к следующему вопросу о достижимости в графе автомата подмножеств  $\mathcal{P}(\mathcal{A})$ : существует ли путь от  $Q$  к одноэлементному подмножеству? Понятно, что наличие или отсутствие такого пути не зависит от меток ребер, и потому последний вопрос в действительности касается носителя автомата  $\mathcal{P}(\mathcal{A})$ . Для решения задач достижимости в графах имеются стандартные методы, например, поиск в ширину, см. [55, §22.2]. Мы видим, что применяя поиск в ширину к носителю автомата подмножеств, можно разрешить вопрос о синхронизируемости исходного автомата.

Описанная процедура концептуально очень проста, но становится неэффективной с ростом числа состояний автомата, поскольку число множеств состояний растёт экспоненциально. Однако следующий критерий синхронизируемости (установленный независимо в пионерских работах Лю [112, теорема 15] и Черни [44, теорема 2] и многократно переоткрывавшийся) приводит к намного более эффективному алгоритму.

**Предложение 2.1.** ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  синхронизируем тогда и только тогда, когда для любых  $q, q' \in Q$  существует слово  $w \in \Sigma^*$  такое, что  $q \cdot w = q' \cdot w$ .

*Доказательство.* Конечно, в доказательстве нуждается только достаточность. Возьмём два произвольных состояния  $q, q' \in Q$  и рассмотрим слово  $w_1$  такое, что  $q \cdot w_1 = q' \cdot w_1$ . Тогда  $|Q \cdot w_1| < |Q|$ . Если  $|Q \cdot w_1| = 1$ , то  $w_1$  синхронизирует  $\mathcal{A}$ . Если  $|Q \cdot w_1| > 1$ , возьмём два состояния  $p, p' \in Q \cdot w_1$  и рассмотрим слово  $w_2$  такое, что  $p \cdot w_2 = p' \cdot w_2$ . Тогда  $|Q \cdot w_1 w_2| < |Q \cdot w_1|$ . Если  $|Q \cdot w_1 w_2| = 1$ , то слово  $w_1 w_2$  синхронизирует  $\mathcal{A}$ ; в противном случае мы продолжаем процесс. Ясно, что этот процесс приведет к синхронизирующему слову для  $\mathcal{A}$  за не более чем  $|Q| - 1$  шагов.  $\square$

Если  $Q$  – конечное множество и  $1 \leq k \leq |Q|$ , мы для краткости называем  $k$ -элементные подмножества в  $Q$  его  $k$ -подмножествами. Множество всех  $k$ -подмножеств в  $Q$  таких, что  $1 \leq k \leq m$  для некоторого  $m \leq |Q|$ , обозначим через  $\mathcal{P}^{\leq m}(Q)$ . Если  $\mathcal{A} = \langle Q, \Sigma \rangle$  – ДКА, то  $|P \cdot a| \leq |P|$  для любого подмножества  $P \in \mathcal{P}(Q)$  и любой буквы  $a \in \Sigma$ . Поэтому для любого фиксированного  $m \leq |Q|$  множество  $\mathcal{P}^{\leq m}(Q)$  замкнуто относительно действия букв из  $\Sigma$  и следовательно, определяет подавтомат в автомате подмножеств. Этот подавтомат называется *автоматом  $m$ -подмножеств* ДКА  $\mathcal{A}$  и обозначается  $\mathcal{P}^{\leq m}(\mathcal{A})$ . Рисунок 12 изображает автомат 2-подмножеств автомата  $\mathcal{C}_4$ .

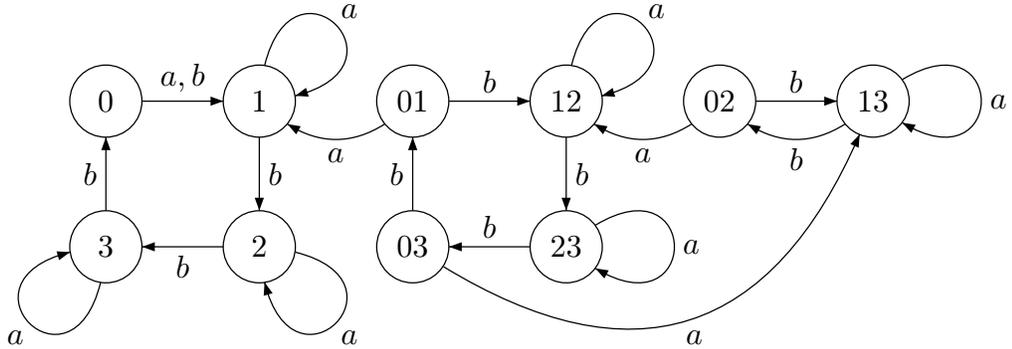


Рис. 12: Автомат 2-подмножеств  $\mathcal{P}^{\leq 2}(\mathcal{C}_4)$

Теперь предложение 2.1 можно переформулировать так:

**Следствие 2.1.** ДКА  $\mathcal{A}$  синхронизируем тогда и только тогда, когда для любого 2-подмножества  $D$  множества его состояний в носителе автомата  $\mathcal{P}^{\leq 2}(\mathcal{A})$  найдётся путь от  $D$  к 1-подмножеству.

Как и выше, вопрос о синхронизируемости данного ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  сводится к вопросу о достижимости в некотором графе. Однако на сей раз мы имеем дело с носителем автомата  $\mathcal{P}^{\leq 2}(\mathcal{A})$ , а этот граф имеет  $\frac{|Q|(|Q|+1)}{2}$  вершин и  $\frac{|Q|(|Q|+1)}{2} \cdot |\Sigma|$  ребер. Известно, что поиск в ширину обрабатывает граф с  $n$  вершинами и  $m$  рёбрами за время  $\Theta(n + m)$ , см., например, [55, §22.2]. Поэтому основанный на следствии 2.1 алгоритм проверяет ДКА  $\mathcal{A}$  на синхронизируемость за время  $O(|Q|^2 \cdot |\Sigma|)$ . Будем называть этот алгоритм *алгоритмом Лю–Черни* в честь его первооткрывателей.

Основываясь на глубоком анализе строения случайных автоматов, М. В. Берлинков предложил идею алгоритма для проверки синхронизируемости данного ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  за время  $O(|Q| \cdot |\Sigma|)$  в среднем, см. [25, §2] и [26, §4]. Алгоритм состоит из набора тестов с ответами ДА/НЕТ. Подробное описание тестов дано в разделе второй части статьи, посвящённом синхронизации случайных автоматов; здесь же приведём три ключевых свойства набора:

- время тестирования ДКА с  $n$  состояниями составляет  $O(n)$ ;
- доля ДКА, не проходящих тесты, среди всех ДКА с  $n$  состояниями есть  $O(\frac{1}{n})$ ;
- ДКА, прошедший все тесты, синхронизируем.

Если какой-то из тестов не пройден, то ДКА может быть синхронизируемым, а может и не быть. В этой ситуации приходится запускать алгоритм Лю–Черни. Поэтому время работы алгоритма Берлинкова в наихудшем случае по-прежнему составляет  $O(|Q|^2 \cdot |\Sigma|)$ . Однако, поскольку доля автоматов с  $n$  состояниями, для которых требуется вызов алгоритма Лю–Черни, есть  $O(\frac{1}{n})$ , среднее время работы алгоритма Берлинкова на ДКА, выбранном равномерно и случайно из всех ДКА с  $n$  состояниями и фиксированным алфавитом, составляет  $O(n)$ . Подчеркнём, что описанный алгоритм не является вероятностным (хотя и опирается на статистические свойства случайных автоматов); он детерминирован и возвращает корректный ответ для любого ДКА.

П. С. Агеев [2] реализовал алгоритм Берлинкова в немного усовершенствованной редакции, см. <https://github.com/birneAgeev/AutomataSynchronizationChecker>. Эксперименты из [2] демонстрируют, что, начиная с ДКА с 31 состоянием, реализация алгоритма Берлинкова опережает реализацию алгоритма Лю–Черни по быстродействию и что с ростом числа состояний преимущество алгоритма Берлинкова возрастает.

Дальнейшие обсуждения в §2 предполагают знакомство читателя с основами теории сложности вычислений, в частности, со стандартными классами сложности **P**, **NP**, **PSPACE** и т.п.<sup>9</sup> В рамках этой теории задача о проверке ДКА на синхронизируемость ставится как следующая задача распознавания:

**DFA-SYNC:** синхронизируемость полного детерминированного автомата.  
**Вход:** Полный детерминированный конечный автомат  $\mathcal{A}$ .  
**Ответ:** ДА, если  $\mathcal{A}$  синхронизируем, НЕТ в противном случае.

Следствие 2.1 сводит задачу DFA-SYNC к известной задаче достижимости в графах:

**PATH:** (ориентированная) достижимость в графе.  
**Вход:** Граф  $\Gamma$  и две его вершины  $s_0$  и  $s_1$ .  
**Ответ:** ДА, если в  $\Gamma$  есть путь из  $s_0$  в  $s_1$ , НЕТ в противном случае.

Задача PATH – одна из «канонических» полных задач для класса сложности **NL**, т.е. класса задач, разрешимых на недетерминированной машине Тьюринга с использованием  $O(\log n)$  дополнительной памяти для входа длиной  $n$ , см. [125, теорема 16.2]. Комбинируя следствие 2.1 и недетерминированный алгоритм для PATH, использующий логарифмическую дополнительную память, легко показать, что задача DFA-SYNC лежит в классе **NL**. В действительности, она, как и PATH, **NL**-полна. Этот результат,

<sup>9</sup>Читатель, не знакомый с этими основами, может «перепрыгнуть» к §3.

полученный в 2010 г. индийским информатиком А. В. Сриджитом, ранее не публиковался. Мы приводим доказательство с любезного разрешения автора.

**Предложение 2.2.** *Задача DFA-SYNC NL-полна.*

*Доказательство.* Ясно, что достаточно L-свести к DFA-SYNC некоторую NL-полную задачу. (L-сведение – это сведение, использующее логарифмическую дополнительную память.) Мы построим L-сведение от сужения PATH на экземпляры вида  $(\Gamma, s_0, s_1)$ , где  $\Gamma$  – граф конфигураций недетерминированной машины Тьюринга, использующей  $O(\log n)$  дополнительной памяти для входа длиной  $n$ ,  $s_0$  отвечает начальной конфигурации машины на данном входе, а  $s_1$  – единственной принимающей конфигурации, в которой машина останавливается. Именно такие экземпляры задачи PATH в действительности используются в доказательстве теоремы 16.2 в [125], поэтому состоящая из них подзадача NL-полна. Произвольную недетерминированную машину Тьюринга можно за счёт введения новых состояний в количестве, логарифмическом от их исходного числа, превратить в машину, у которой из каждой конфигурации есть переход в не более чем две. Поэтому без ограничения общности можно считать, что в графе  $\Gamma$  каждая вершина служит началом не более чем двух рёбер.

Итак, пусть  $(\Gamma = \langle V, E, h, t \rangle, s_0, s_1)$  – экземпляр указанной подзадачи. Добавляя к множеству  $E$  петли в вершинах, в которых начинается меньше двух рёбер, модифицируем граф  $\Gamma$  так, чтобы каждая вершина из  $V$  служила началом ровно двух рёбер. (Скажем, оба ребра, начинающиеся в  $s_1$ , будут петлями.) Понятно, что на наличие/отсутствие пути из  $s_0$  в  $s_1$  такая модификация не влияет. Теперь пометим рёбра графа  $\Gamma$  с помощью двух букв  $a$  и  $b$  так, чтобы для каждой вершины  $v \in V$  одно из рёбер, начинающихся в  $v$ , имело метку  $a$ , а второе – метку  $b$ ; в остальном выбор меток произволен. Получившийся помеченный граф представляет ДКА  $\mathcal{G} := \langle V, \{a, b\}, \zeta \rangle$ , в котором функция перехода  $\zeta$  определяется метками:  $\zeta(v, c) = v'$  для  $c \in \{a, b\}$  и  $v, v' \in V$ , если и только если ребро  $v \rightarrow v'$  несёт метку  $c$ .

Пусть  $\bar{V} := \{\bar{v} : v \in V\}$ . Положим  $Q := V \cup \bar{V}$  и построим ДКА  $\mathcal{A}(\Gamma) := \langle Q, \{0, 1\}, \delta \rangle$ , определив функцию  $\delta$  следующим образом:

$$\begin{aligned} \delta(v, 0) &:= \bar{v}, \quad \delta(v, 1) := \zeta(v, a), \quad \delta(\bar{v}, 0) := \bar{s}_1, \quad \delta(\bar{v}, 1) := \zeta(v, b) \text{ для } v \in V \setminus \{s_1\}, \\ \delta(s_1, 0) &= \delta(s_1, 1) := s_1, \quad \delta(\bar{s}_1, 0) := \bar{s}_1, \quad \delta(\bar{s}_1, 1) := s_0. \end{aligned}$$

Схема автомата  $\mathcal{A}(\Gamma)$  приведена на рис. 13.

Ясно, что построение автомата  $\mathcal{A}(\Gamma)$  по исходной тройке  $(\Gamma, s_0, s_1)$  осуществимо с использованием логарифмической дополнительной памяти. Отметим ещё, что если определить кодирование  $\chi: \{a, b\}^+ \rightarrow \{0, 1\}^+$  правилом  $a\chi := 1$ ,  $b\chi := 01$ , то, как легко проверить,  $\zeta(v, x) = \delta(v, x\chi)$  для любых  $v \in V$  и  $x \in \{a, b\}^*$ .

Утверждается, что в графе  $\Gamma$  имеется путь из  $s_0$  в  $s_1$  тогда и только тогда, когда ДКА  $\mathcal{A}(\Gamma)$  синхронизируем. Действительно, если такой путь есть и метки его рёбер в автомате  $\mathcal{G}$  составляют слово  $x \in \{a, b\}^+$ , непосредственно проверяется, что слово  $001 \cdot x\chi$  синхронизирует  $\mathcal{A}(\Gamma)$ . Обратно, пусть ДКА  $\mathcal{A}(\Gamma)$  синхронизируем. Поскольку  $\delta(s_1, 0) = \delta(s_1, 1) = s_1$ , любое слово из  $\{0, 1\}^*$  фиксирует состояние  $s_1$ , и потому синхронизирующее слово переводит  $\mathcal{A}(\Gamma)$  именно в это состояние. В частности, есть слова  $w \in \{0, 1\}^+$ , такие, что  $\delta(s_0, w) = s_1$  (например, синхронизирующее слово таково). Возьмём кратчайшее слово  $w$  с таким свойством. Путь из  $s_0$  в  $s_1$  в автомате  $\mathcal{A}(\Gamma)$  вдоль  $w$  не посещает состояние  $s_0$  второй раз, а потому этот путь не заходит в состояние  $\bar{s}_1$ , из которого можно перейти только в  $s_0$ . Отсюда в  $w$  не встречается блок  $00$ , поскольку двукратное действие  $0$  переводит в  $\bar{s}_1$  любое состояние, кроме  $s_1$ . Легко видеть, что слова над  $\{0, 1\}$ , в которых отсутствует блок  $00$ , лежат в образе  $\chi$ , откуда  $w = x\chi$  для некоторого  $x \in \{a, b\}^+$ . Путь в  $\mathcal{G}$  вдоль слова  $x$  ведёт из  $s_0$  в  $s_1$ .  $\square$

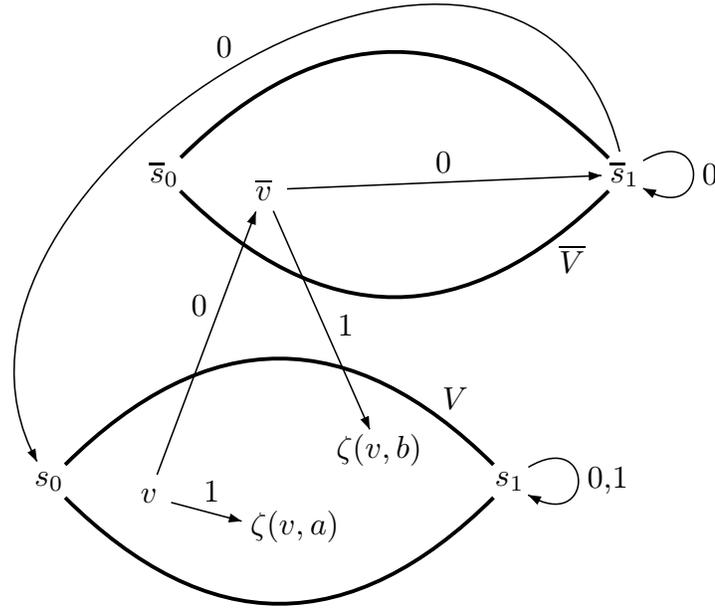


Рис. 13: Схема автомата  $\mathcal{A}(\Gamma)$ ;  $v$  изображает произвольное состояние из  $V \setminus \{s_1\}$

## 2.2 Синхронизируемые автоматы и регулярные языки

(Формальным) языком над данным конечным алфавитом  $\Sigma$  называется любое подмножество в  $\Sigma^*$ . С помощью конечных автоматов выделяется весьма важный тип языков – *регулярные языки*; автоматы при этом выступают в роли *распознавателей*. Чтобы превратить ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  в распознаватель, достаточно выделить в множестве  $Q$  состояние  $q_0$ , называемое *начальным*, и непустое подмножество  $F$ , состояния из которого называются *заключительными*. Распознаватель  $(\mathcal{A}, q_0, F)$  *принимает* слово  $w \in \Sigma^*$ , если  $w$  – последовательность меток вдоль пути в  $\mathcal{A}$ , который начинается в состоянии  $q_0$  и заканчивается в одном из состояний из  $F$ , т.е. если  $q_0 \cdot w \in F$ . Множество всех слов, принимаемых  $(\mathcal{A}, q_0, F)$ , называется *языком, распознаваемым автоматом  $\mathcal{A}$* . Регулярный язык в  $\Sigma^*$  – это язык, распознаваемый каким-то ДКА с входным алфавитом  $\Sigma$ . Классическая теорема Клини [101] характеризует класс регулярных языков над  $\Sigma$  как наименьший класс языков, который:

- содержит пустой язык и все языки вида  $\{a\}$ , где  $a \in \Sigma$ ;
- вместе с любым языком  $L$  содержит его *итерацию*  $L^*$ , т.е. множество всевозможных конечных произведений слов из  $L$  (включая пустое произведение, которое считается равным пустому слову  $\varepsilon$ );
- вместе с любыми двумя языками  $L$  и  $K$  содержит их теоретико-множественное объединение  $L \cup K$  и их *произведение*  $LK$ , т.е. множество всевозможных произведений слов из  $L$  на слова из  $K$ .

Напомним, что для синхронизируемого ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  через  $\text{Sync } \mathcal{A}$  обозначается множество всех его синхронизирующих слов. Конструкция автомата подмножеств из §2.1 позволяет легко доказать следующий факт:

**Предложение 2.3.**  $\text{Sync } \mathcal{A}$  – регулярный язык для синхронизируемого ДКА  $\mathcal{A}$ .

*Доказательство.* Пусть  $\mathcal{A} = \langle Q, \Sigma \rangle$ . Слово из  $\Sigma^*$  синхронизирует  $\mathcal{A}$  тогда и только тогда, когда помеченный им путь в автомате подмножеств  $\mathcal{P}(\mathcal{A})$  ведёт из  $Q$  в некоторое

одноэлементное подмножество. Поэтому язык  $\text{Sync } \mathcal{A}$  распознается ДКА  $\mathcal{P}(\mathcal{A})$ , если в качестве начального состояния выбрать  $Q$ , а в качестве множества заключительных состояний – множество  $\{\{q\} : q \in Q\}$ .  $\square$

Напомним, что  $\text{Sync } \mathcal{A}$  – идеал моноида  $\Sigma^*$  (лемма 1.1). С учётом этого, предложение 2.3 означает, что для любого синхронизируемого ДКА  $\mathcal{A}$  язык  $\text{Sync } \mathcal{A}$  есть непустой регулярный идеал. Нетрудно видеть, что такая формулировка допускает обращение: для произвольного регулярного идеала  $I$  моноида  $\Sigma^*$  найдётся ДКА  $\mathcal{A}$  такой, что  $I = \text{Sync } \mathcal{A}$ . (Можно проверить, что таким ДКА будет ДКА наименьшего размера, распознающий  $I$ .) При этом если  $|\Sigma| \geq 2$ , то автомат  $\mathcal{A}$  можно выбрать сильно связным – это неочевидное уточнение получено Рожериу Рейшем и Эмануэле Родаро [138].

В 2010-х гг. изучение взаимосвязей между синхронизируемыми автоматами и регулярными идеалами моноида  $\Sigma^*$  развилось в довольно активное исследовательское направление в духе так называемой дескриптивной сложности (descriptional complexity) – теории, которая измеряет сложность объектов размерами их описаний. Непустые регулярные идеалы моноида  $\Sigma^*$  – это бесконечные объекты, а синхронизируемые автоматы позволяют описать их в терминах конечных множеств, причём зачастую такое описание оказывается намного более компактным, чем «стандартное» описание с помощью распознавателей. Например, можно проверить, что любой ДКА, распознающий язык  $\text{Sync } \mathcal{C}_4$ , где  $\mathcal{C}_4$  – синхронизируемый ДКА с четырьмя состояниями из примера 1.1, имеет не менее 12 состояний. Среди публикаций в указанном направлении можно отметить [71, 86, 114–116, 133, 140].

Вернёмся теперь к вычислительной сложности распознавания синхронизируемости. Недавно обнаружилось (см. [65]), что эта сложность может существенно меняться при наложении простейших ограничений. В [65] систематически изучается семейство задач, параметризованное регулярными языками:

**DFA-SYNC[ $L$ ]**: синхронизируемость ДКА при регулярном ограничении  $L$ .  
**ВХОД:** Полный детерминированный конечный автомат  $\mathcal{A}$ .  
**ОТВЕТ:** ДА, если  $\mathcal{A}$  синхронизируем словом из  $L$ , НЕТ в противном случае.

Рассмотренная в §2.1 задача DFA-SYNC входит в это семейство: она получается, когда в роли регулярного языка  $L$  выбирается язык всех слов над входным алфавитом автомата, что не накладывает никаких ограничений на строение синхронизирующих слов. На практике, однако, часто возникают ситуации, когда синхронизирующие слова обязаны иметь определённый формат. Так, если устройство, моделируемое автоматом, может функционировать в двух режимах – обычном и отладочном, то инструкции для отладки должны начинаться и заканчиваться специальной командой, которая сначала переключает устройство в отладочный режим, а затем возвращает его к обычной работе. (Пользователь ТРХа наверняка вспомнит здесь про переключение ТРХа в математический режим и обратно с помощью знака \$.) Такое ограничение соответствует регулярному языку вида  $a\Xi^*a$ , где  $a$  – выделенная специальная команда, а  $\Xi$  – набор всех остальных команд устройства. Оказывается, что уже это естественное ограничение сразу делает задачу синхронизируемости вычислительно сложной:

**Предложение 2.4.** *Задача DFA-SYNC[ $a\Xi^*a$ ], где  $a \notin \Xi$ , NP-полна при  $|\Xi| = 1$  и PSPACE-полна при  $|\Xi| > 1$ .*

В обоих случаях ( $|\Xi| = 1$  и  $|\Xi| > 1$ ) трудность задачи DFA-SYNC[ $a\Xi^*a$ ] обеспечивается несложным сведением от задачи FA-INTERSECTION, в которой дано конечное

семейство распознавателей над алфавитом  $\Xi$  и требуется определить, есть ли слово из  $\Xi^*$ , принимаемое всеми этими распознавателями. (Сведёние описано в доказательстве предложения 1 в [65].) Задача FA-INTERSECTION PSPACE-полна при  $|\Xi| > 1$  (см. [105]) и NP-полна при  $|\Xi| = 1$  (см., например, [67]). В [65] показано, что для любого регулярного языка  $L$  задача DFA-SYNC[ $L$ ] лежит в классе PSPACE; отсюда вытекает PSPACE-полнота задачи DFA-SYNC[ $a\Xi^*a$ ] при  $|\Xi| > 1$ . В случае, когда  $\Xi$  состоит из одной буквы, скажем,  $b$ , принадлежность задачи DFA-SYNC[ $a\Xi^*a$ ] классу NP не совсем очевидна, так как у синхронизируемого ДКА  $\mathcal{A} = \langle Q, \{a, b\} \rangle$  может не быть синхронизирующих слов из  $ab^*a$  полиномиальной от  $|Q|$  длины. Однако в [65, предложение 1] проверено, что эта задача лежит в NP.

В [65] приведено много примеров регулярных языков  $L$  над двух- и трёхбуквенными алфавитами, для которых задача DFA-SYNC[ $L$ ] PSPACE-полна. Полная классификация таких языков пока остаётся открытой проблемой. Сложность синхронизации при различных регулярных ограничениях исследуется также в препринтах [89–91].

### 2.3 Сложность вычисления порога синхронизации

Минимум длин синхронизирующих слов синхронизируемого ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется его *порогом синхронизации* и обозначается  $rt(\mathcal{A})$ . Для вычисления этого параметра может использоваться автомат подмножеств, поскольку  $rt(\mathcal{A})$  есть длина кратчайшего пути из  $Q$  в одноэлементное множество в носителе автомата  $\mathcal{P}(\mathcal{A})$ . Конечно, в наихудшем случае для этого потребуются экспоненциальное от  $|Q|$  время. Тем не менее попытки реализовать этот подход были; см., например, [97, 139, 174]. В отличие от ситуации с проверкой синхронизируемости переход к «полиномиальному» подавтомату 2-подмножеств здесь не помогает. Более того, при стандартных допущениях теории сложности вычислений можно показать, что вычисление порога синхронизации – труднорешаемая задача.

Начнём анализ с обсуждения следующей задачи распознавания:

SHORT-SW: синхронизация словом данной длины.  
 Вход: ДКА  $\mathcal{A}$  и натуральное число  $\ell$ .  
 Ответ: ДА, если  $rt(\mathcal{A}) \leq \ell$ , НЕТ в противном случае.

В формулировке задачи SHORT-SW не сказано, в какой форме – унарной или бинарной – записывается число  $\ell$ . Дело в том, что оба варианта данной задачи равноправны. (Это следует из того факта, что если ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  синхронизируем, то  $rt(\mathcal{A}) < |Q|^3$ ; см. подробное обсуждение в §3.2 ниже.)

И. К. Рысцов [150, теорема 3] доказал NP-полноту задачи SHORT-SW. Как это нередко случалось в теории синхронизируемых автоматов, пионерская работа осталась незамеченной, и впоследствии этот важный результат неоднократно переоткрывался, см., например, [63, 81, 158].

**Предложение 2.5.** *Задача SHORT-SW NP-полна.*

*Доказательство.* Чтобы показать, что задача SHORT-SW лежит в классе NP, сначала проверим, синхронизируем ли данный ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ . Как показано в §2.1, это можно сделать за полиномиальное от  $|Q|$  время. Если  $\mathcal{A}$  несинхронизируем, ответ на экземпляр  $(\mathcal{A}, \ell)$  задачи SHORT-SW отрицателен. Если  $\mathcal{A}$  синхронизируем, проверим неравенство  $\ell \geq |Q|^3$ ; если оно выполнено, то ответ на экземпляр  $(\mathcal{A}, \ell)$  положителен. Наконец, если  $\ell < |Q|^3$ , то недетерминированный алгоритм угадывает слово  $w \in \Sigma^*$

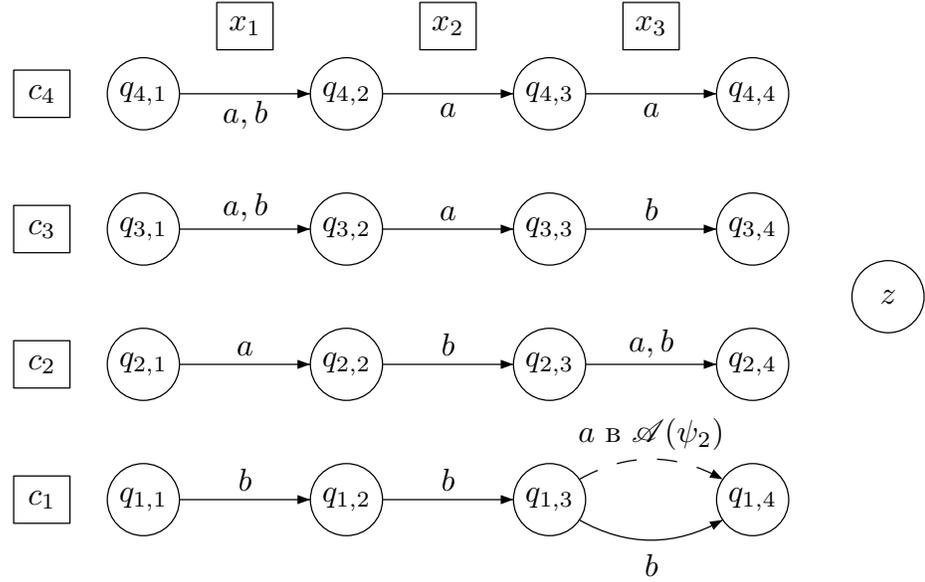


Рис. 14: Автоматы  $\mathcal{A}(\psi_1)$  и  $\mathcal{A}(\psi_2)$

длины  $\leq \ell$ , а затем проверяет, синхронизирует ли  $w$  автомат  $\mathcal{A}$ . Такая проверка производится за время  $\leq \ell|Q|$ .

Для доказательства **NP**-трудности используется сведение от классической **NP**-полной задачи SAT, см. [125, теорема 8.2]. Напомним, что вход SAT – это система *клиз* (дизъюнкций *литералов*, т.е. булевых переменных и их отрицаний), а на выходе требуется определить, существует ли набор значений переменных, обращающий все клизы системы в истинные высказывания. Для произвольного экземпляра  $\psi$  задачи SAT с  $n$  переменными  $x_1, \dots, x_n$  и  $m$  клизами  $c_1, \dots, c_m$  построим ДКА  $\mathcal{A}(\psi) := \langle Q, \{a, b\} \rangle$  следующим образом. Множество  $Q$  состоит из  $(n+1)m$  символов  $q_{i,j}$ , где  $1 \leq i \leq m$ ,  $1 \leq j \leq n+1$ , и особого символа  $z$ . Действие букв определяется правилами:

$$\begin{aligned}
 q_{i,j} \cdot a &:= \begin{cases} z, & \text{если литерал } x_j \text{ входит в клиз } c_i, \\ q_{i,j+1}, & \text{в противном случае} \end{cases} & \text{при } 1 \leq i \leq m, 1 \leq j \leq n; \\
 q_{i,j} \cdot b &:= \begin{cases} z, & \text{если литерал } \neg x_j \text{ входит в клиз } c_i, \\ q_{i,j+1}, & \text{в противном случае} \end{cases} & \text{при } 1 \leq i \leq m, 1 \leq j \leq n; \\
 q_{i,n+1} \cdot a = q_{i,n+1} \cdot b = z \cdot a = z \cdot b &:= z & \text{при } 1 \leq i \leq m.
 \end{aligned}$$

Рисунок 14 изображает два автомата вида  $\mathcal{A}(\psi)$ , построенных для двух систем клизов:

$$\begin{aligned}
 \psi_1 &= \{c_1 := x_1 \vee x_2 \vee x_3, c_2 := \neg x_1 \vee x_2, c_3 := \neg x_2 \vee x_3, c_4 := \neg x_2 \vee \neg x_3\}, \\
 \psi_2 &= \{c_1 := x_1 \vee x_2, c_2 := \neg x_1 \vee x_2, c_3 := \neg x_2 \vee x_3, c_4 := \neg x_2 \vee \neg x_3\}.
 \end{aligned}$$

Каждая «строчка» на рис. 14 помечена соответствующим клизом, а каждый «столбик» – соответствующей переменной. Если из состояния  $q \in Q$  на рис. 14 не исходит ребро с меткой  $c \in \{a, b\}$ , то подразумевается ребро  $q \xrightarrow{c} z$  (все такие рёбра опущены на рисунке, чтобы не загромождать его). Системы  $\psi_1$  и  $\psi_2$  различаются только первым клизом: в  $\psi_1$  этот клиз содержит литерал  $x_3$ , а в первом клизе  $\psi_2$  такого литерала нет. Соответственно, автоматы  $\mathcal{A}(\psi_1)$  и  $\mathcal{A}(\psi_2)$  отличаются только исходящим ребром с меткой  $a$  в состоянии  $q_{1,3}$ : в  $\mathcal{A}(\psi_1)$  это ребро ведёт в  $z$  (и потому не изображено на рис. 14), а в  $\mathcal{A}(\psi_2)$  оно ведёт в состояние  $q_{1,4}$  (и показано пунктиром).

Заметим, что набор значений истинности  $x_1 = x_2 := 0, x_3 := 1$  выполняет все клозы из  $\psi_1$ , в то время как система клозов  $\psi_2$  невыполнима. Нетрудно проверить, что слово  $bba$  синхронизирует  $\mathcal{A}(\psi_1)$ , в то время как  $\mathcal{A}(\psi_2)$  не синхронизируется никаким словом длины 3 (но синхронизируется любым словом над  $\{a, b\}$  длины 4).

В общем случае легко проверить, что ДКА  $\mathcal{A}(\psi)$  синхронизируется любым словом над  $\{a, b\}$  длины  $n + 1$  и синхронизируется словом длины  $n$  тогда и только тогда, когда система  $\psi$  выполнима. Таким образом, сопоставив произвольному экземпляру  $\psi$  задачи SAT экземпляр  $(\mathcal{A}(\psi), n)$  задачи SHORT-SW, где  $n$  – число переменных в  $\psi$ , получим полиномиальное сведение от SAT к SHORT-SW.  $\square$

Введём задачу перечисления, соответствующую SHORT-SW:

#SHORT-SW: перечисление синхронизирующих слов данной длины.

ВХОД: ДКА  $\mathcal{A}$  и натуральное число  $\ell$ .

ОТВЕТ: Число синхронизирующих слов длины  $\ell$  для  $\mathcal{A}$ .

Сведение из доказательства предложения 2.5 устанавливает взаимно однозначное соответствие между выполняющими наборами значений переменных для системы клозов  $\psi$  от  $n$  переменных и синхронизирующими словами длины  $n$  для ДКА  $\mathcal{A}(\psi)$ . В частности, число выполняющих наборов для  $\psi$  равно числу синхронизирующих слов длины  $n$  для  $\mathcal{A}(\psi)$ , т.е. построенное сведение *экономно* (parsimonious) в смысле теории сложности перечислений. Поэтому его можно рассматривать как сведение задачи #SAT (задача перечисления выполняющих наборов для данной системы клозов) к задаче #SHORT-SW. Поскольку задача #SAT #P-полна (см. [125, теорема 18.1]), получаем, что задача #SHORT-SW #P-трудна. С другой стороны, легко видеть, что #SHORT-SW лежит в классе #P. В итоге получается факт, впервые явно отмеченный, по-видимому, Йоргом Ольшевским и Михаэлем Уммельсом [124, замечание 3]:

**Следствие 2.2.** *Задача #SHORT-SW #P-полна.*

В литературе довольно часто можно встретить утверждения типа «задача нахождения длины кратчайшего синхронизирующего слова NP-полна» или даже «задача нахождения кратчайшего синхронизирующего слова NP-полна»; при этом подразумевается результат предложения 2.5. Конечно, такая интерпретация предложения 2.5 некорректна; более того, конструкция из доказательства этого предложения позволяет заключить, что принадлежность упомянутых задач классу NP несовместима со стандартными допущениями теории сложности вычислений. Чтобы продемонстрировать это, введём соответствующую задачу распознавания.

SHORTEST-SW: равенство порога синхронизации данному числу.

ВХОД: ДКА  $\mathcal{A}$  и натуральное число  $\ell$ .

ОТВЕТ: ДА, если  $rt(\mathcal{A}) = \ell$ , НЕТ в противном случае.

Взяв произвольную систему клозов  $\psi$  от  $n$  переменных, сопоставим ей экземпляр  $(\mathcal{A}(\psi), n + 1)$  задачи SHORTEST-SW. Как отмечено в доказательстве предложения 2.5, ДКА  $\mathcal{A}(\psi)$  синхронизируется любым словом над  $\{a, b\}$  длины  $n + 1$  и синхронизируется словом длины  $n$  тогда и только тогда, когда система  $\psi$  выполнима. Поэтому  $rt(\mathcal{A}(\psi)) = n + 1$  тогда и только тогда, когда система  $\psi$  невыполнима. Получаем сведение к SHORTEST-SW от отрицания SAT, т.е. от coNP-полной задачи. Следовательно, задача SHORTEST-SW coNP-трудна. Поэтому принадлежность SHORTEST-SW

классу  $\mathbf{NP}$  имело бы следствием равенство  $\mathbf{NP} = \mathbf{coNP}$ , которое считается маловероятным. Если же принять гипотезу  $\mathbf{NP} \neq \mathbf{coNP}$ , то SHORTEST-SW не лежит в  $\mathbf{NP}$ , что означает, что даже недетерминированный алгоритм не может определить порог синхронизации данного ДКА за полиномиальное время от размера этого ДКА.

Класс сложности, для которого SHORTEST-SW служит полной задачей, нашли Павел Гавриховский [72] и, независимо от него, Ольшевски и Уммельс [124, теорема 1]. Нужным классом оказался класс  $\mathbf{DP}$  (Difference Polynomial-Time), введённый Христосом Пападимитриу и Михалисом Яннакакисом [126], см. также [125, §17.1]. Он состоит из всех задач  $Z$ , допускающих полиномиальное сведение к паре задач  $Z_+$  и  $Z_-$  из  $\mathbf{NP}$ , при котором экземпляры задачи  $Z$  с ответом ДА преобразуются в точности в пары вида (экземпляр  $Z_+$  с ответом ДА, экземпляр  $Z_-$  с ответом НЕТ).  $\mathbf{DP}$  – обширный класс, который включает объединение  $\mathbf{NP} \cup \mathbf{coNP}$ , и предполагается, что включение – строгое. Типичной  $\mathbf{DP}$ -полной задачей является задача SAT-UNSAT, в которой для пары систем кловов  $\psi, \varphi$  нужно узнать, верно ли, что  $\psi$  выполнима, а  $\varphi$  невыполнима. С помощью сведения от SAT-UNSAT в [72] и [124] доказано

**Предложение 2.6.** *Задача SHORTEST-SW  $\mathbf{DP}$ -полна.*

Теперь локализуем задачу **нахождения** порога синхронизации. Она ставится так:

COMPUTE-RT: нахождение порога синхронизации.  
 Вход: Синхронизируемый ДКА  $\mathcal{A}$ .  
 Ответ: Величина  $rt(\mathcal{A})$ .

Отличие COMPUTE-RT от SHORTEST-SW состоит в том, что мы напрямую ищем порог синхронизации, а не проверяем его равенство заранее заданному числу.

Функциональный класс сложности  $\mathbf{FP}^{\mathbf{NP}^{[\log]}}$  состоит из всех функций, вычислимых за полиномиальное время на детерминированной машине Тьюринга, которая может обращаться к оракулу для  $\mathbf{NP}$ -полной задачи, причём число запросов есть логарифмическая функция от размера входных данных, см. [125, §17.1]. Легко понять, что функция  $\mathcal{A} \mapsto rt(\mathcal{A})$  лежит в этом классе. Действительно, уже отмечалось, что если ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  синхронизируем, то  $rt(\mathcal{A}) < |Q|^3$ . Поэтому с помощью двоичного поиска можно найти  $rt(\mathcal{A})$ , запрашивая оракул для  $\mathbf{NP}$ -полной задачи SHORTEST-SW  $O(\log |Q|)$  раз. Как доказали Ольшевски и Уммельс [124, теорема 4], эта очевидная верхняя оценка на сложность задачи COMPUTE-RT точна.

**Предложение 2.7.** *Задача COMPUTE-RT  $\mathbf{FP}^{\mathbf{NP}^{[\log]}}$ -полна.*

Доказательство основано на сведении от задачи MAX-SAT-SIZE,  $\mathbf{FP}^{\mathbf{NP}^{[\log]}}$ -полнота которой установлена в [106]. (В MAX-SAT-SIZE для данной системы кловов требуется найти максимальное число одновременно выполнимых кловов.)

Наконец, обсудим нахождение синхронизирующего слова минимальной длины:

COMPUTE-SW: нахождение синхронизирующего слова минимальной длины.  
 Вход: Синхронизируемый ДКА  $\mathcal{A}$ .  
 Ответ: Синхронизирующее слово минимальной длины для  $rt(\mathcal{A})$ .

Понятно, что найти само синхронизирующее слово минимальной длины не легче, чем найти его длину, т.е. порог синхронизации. Поэтому из предложения 2.7 следует, что задача COMPUTE-SW  $\mathbf{FP}^{\mathbf{NP}^{[\log]}}$ -трудна. С другой стороны, несложно проверить,

что синхронизирующее слово минимальной длины для ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  можно вычислить за полиномиальное от  $|Q|$  время на детерминированной машине Тьюринга, если разрешить запросы к оракулу для следующей **NP**-полной задачи, см. [124, теорема 5]:

**SHORT-SUBSET-SW**: синхронизация подмножества словом данной длины.  
**Вход**: ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , подмножество  $S \subseteq Q$  и число  $\ell$  в унарной записи.  
**Ответ**: ДА, если существует такое слово  $w \in \Sigma^*$ , что  $|S \cdot w| = 1$  и  $|w| \leq \ell$ ,  
**НЕТ** в противном случае.

**NP**-трудность задачи **SHORT-SUBSET-SW** следует из предложения 2.5, поскольку задача **SHORT-SW** является её частным случаем (при  $S = Q$ ), а принадлежность **SHORT-SUBSET-SW** классу **NP** очевидна.<sup>10</sup>

Класс всех функций, вычисляемых за полиномиальное от время на детерминированной машине Тьюринга, использующей оракул для **NP**-полной задачи, обозначается через  $\mathbf{FP}^{\mathbf{NP}}$ ; этот класс важен для теории сложности вычислений как «место прописки» задачи коммивояжера, см. [125, теорема 17.5]. Предполагается, что класс  $\mathbf{FP}^{\mathbf{NP}}$  строго больше класса  $\mathbf{FP}^{\mathbf{NP}[\log]}$ .

Итак, сложность задачи **COMPUTE-SW** ограничена классом  $\mathbf{FP}^{\mathbf{NP}[\log]}$  снизу и классом  $\mathbf{FP}^{\mathbf{NP}}$  сверху. Пока неизвестно, полна ли эта задача для какого-то встречавшегося в литературе функционального класса сложности.

Упомянем ещё работы [66, 186], в которых задача **SHORT-SW** исследуется в рамках популярной сейчас параметрической теории сложности.

## 2.4 Сложность аппроксимации порога синхронизации

Поскольку точное значение порога синхронизации трудновычислимо, естественно задаться вопросом о существовании полиномиальных приближённых алгоритмов. Оказалось, что в предположении, что  $\mathbf{P} \neq \mathbf{NP}$ , для любого полиномиально вычислимого приближения к порогу синхронизации найдутся серии автоматов, на которых эти приближения дают большую относительную погрешность. Этот факт установили Павел Гавриховский и Дамиан Страшак [73]; он перекрывает все предшествующие результаты о трудноаппроксимируемости порога синхронизации, см. [24, 25, 74].

Дадим необходимые для обсуждений определения. *Полиномиальным алгоритмом для аппроксимации порога синхронизации* назовем произвольный алгоритм  $U$ , который для каждого синхронизируемого автомата  $\mathcal{A}$  за полиномиальное от числа его состояний время возвращает натуральное число  $U(\mathcal{A}) \geq \text{rt}(\mathcal{A})$ . Пусть  $f: \mathbb{N} \rightarrow \mathbb{R}$  – некоторая функция. Скажем, что  $U$  *аппроксимирует порог синхронизации с точностью до  $f(n)$* , если для любого  $n \in \mathbb{N}$  и любого синхронизируемого автомата  $\mathcal{A}$  с  $n$  состояниями верно неравенство

$$\frac{U(\mathcal{A})}{\text{rt}(\mathcal{A})} \leq f(n).$$

**Предложение 2.8** ([73], теорема 16). *Пусть  $\epsilon > 0$ . Если  $\mathbf{P} \neq \mathbf{NP}$ , никакой полиномиальный алгоритм не аппроксимирует порог синхронизации с точностью до  $n^{1-\epsilon}$ .*

Доказательство предложения 2.8 задействует мощные методы современной теории сложности вычислений, в частности, полученную Дэвидом Цукерманом [188] дерандо-

<sup>10</sup>Отметим, что в формулировке задачи **SHORT-SUBSET-SW** условие, что число  $\ell$  представлено в унарной записи, существенно! Та же задача с бинарно заданным  $\ell$  будет **PSPACE**-полной, см. [121, 144].

мизацию классического результата Йохана Хостада [88] о трудноаппроксимируемости задачи о максимальной клике.

Результат предложения 2.8 нелучшаем: существуют полиномиальные алгоритмы, которые аппроксимирует порог синхронизации с точностью до  $n$ . Так, Майкл Гербуш и Brent Хиринга [74] построили для каждого фиксированного целого  $k \geq 2$  алгоритм, который, получив на вход синхронизируемый автомат с  $n$  состояниями,  $m$  входными буквами и порогом синхронизации  $\ell$ , находит его синхронизирующее слово длины не более  $\lceil \frac{n-1}{k-1} \rceil \ell$  за время  $O(kmn^k + \frac{n^4}{k})$ . Другими словами, такой алгоритм аппроксимирует порог синхронизации с точностью до  $\lceil \frac{n-1}{k-1} \rceil$ ; в частности, при  $k = 2$  получаем аппроксимацию с точностью до  $n - 1$ .

Алгоритмы из [74] получены модификацией жадного сжимающего алгоритма, который подробно обсуждается в §3.2 ниже, см. там алгоритм 1. Д. С. Ананичев и В. В. Гусев [7] показали, что верхняя оценка точности  $\lceil \frac{n-1}{k-1} \rceil$  не может быть улучшена в рамках очень широкого семейства полиномиальных алгоритмов для аппроксимации порога синхронизации; это семейство включает алгоритмы из [74].

### 3 Гипотеза Черни

#### 3.1 Гипотеза Черни: формулировка и немного истории

Для краткости будем называть ДКА с  $n$  состояниями  $n$ -автоматом. В 1964 г. Черни [44] указал серию синхронизируемых  $n$ -автоматов с порогом синхронизации  $(n-1)^2$ . Серия Черни состоит из автоматов  $\mathcal{C}_n = \langle \{0, 1, \dots, n-1\}, \{a, b\} \rangle$ , где буквы  $a$  и  $b$  действуют следующим образом:

$$i \cdot a := \begin{cases} i, & \text{если } i > 0, \\ 1, & \text{если } i = 0; \end{cases} \quad i \cdot b = i + 1 \pmod{n}.$$

Наш первый пример синхронизируемого автомата (см. пример 1.1 и рис. 1) – это 4-автомат из серии Черни;  $n$ -автомат  $\mathcal{C}_n$  изображен слева на рис. 15.

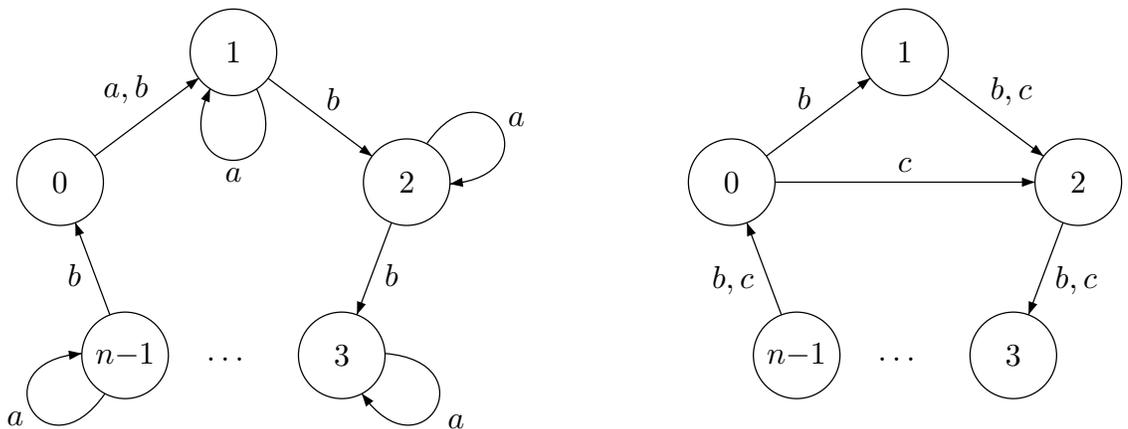


Рис. 15: ДКА  $\mathcal{C}_n$  и ДКА  $\mathcal{W}_n$ , определяемый действиями слов  $b$  и  $c = ab$

Серия  $\{\mathcal{C}_n\}_{n=2,3,\dots}$  многократно переоткрывалась (см., например, [63, 68, 70, 109]). Легко видеть, что слово  $(ab^{n-1})^{n-2}a$  длины  $n(n-2) + 1 = (n-1)^2$  синхронизирует автомат  $\mathcal{C}_n$  к состоянию 1.

**Предложение 3.1** ([44, лемма 1]). *Порог синхронизации автомата  $\mathcal{C}_n$  равен  $(n-1)^2$ .*

Есть несколько изящных доказательств этого результата. Доказательство из [12], представленное ниже, основано на прозрачной идее и увязывает автоматы из серии Черни с экстремальной серией графов из классической работы Хельмута Виландта [187].

*Доказательство предложения 3.1.* Достаточно показать, что если  $w$  – синхронизирующее слово минимальной длины для  $\mathcal{C}_n$ , то  $|w| \geq (n-1)^2$ . Поскольку буква  $b$  действует на  $Q$  как циклическая перестановка, слово  $w$  не может заканчиваться на  $b$ . (В противном случае удаление последней буквы дало бы более короткое синхронизирующее слово.) Таким образом,  $w = w'a$ , причём префикс  $w'$  таков, что  $Q \cdot w' = \{0, 1\}$ .

Поскольку буква  $a$  фиксирует каждое состояние в своем образе  $\{1, 2, \dots, n-1\}$ , за каждым вхождением  $a$  в  $w$ , кроме последнего, следует вхождение  $b$ . (В противном случае в  $w$  встретились бы два вхождения буквы  $a$  подряд и, удалив одно из этих вхождений, мы получили бы более короткое синхронизирующее слово.) Поэтому, если положить  $c := ab$ , то слово  $w'$  можно переписать в слово  $v$  над алфавитом  $\{b, c\}$ . Действия  $b$  и  $c$  индуцируют новый ДКА на множестве состояний  $Q$ ; обозначим этот индуцированный ДКА (изображённый на рис. 15 справа) через  $\mathcal{W}_n$ . Поскольку слова  $w'$  и  $v$  действуют на  $Q$  одинаково, слово  $vc$  синхронизирует автомат  $\mathcal{W}_n$  к состоянию 2.

По лемме 1.1 для произвольного  $u \in \{b, c\}^*$  слово  $uvc$  также синхронизирует автомат  $\mathcal{W}_n$  к состоянию 2. Следовательно, для каждого  $\ell \geq |vc|$  существует путь длины  $\ell$  в  $\mathcal{W}_n$  из любого заданного состояния  $i$  в 2. В частности, полагая  $i = 2$ , мы заключаем, что для каждого  $\ell \geq |vc|$  существует цикл длины  $\ell$  в  $\mathcal{W}_n$ . Носитель автомата  $\mathcal{W}_n$  имеет простые циклы только двух длин:  $n$  и  $n-1$ . Каждый цикл в  $\mathcal{W}_n$  должен состоять из простых циклов этих двух длин, поэтому каждое число  $\ell \geq |vc|$  должно выражаться как комбинация  $n$  и  $n-1$  с неотрицательными целыми коэффициентами. Напомним следующий хорошо известный элементарный результат из теории чисел:

**Лемма 3.1** ([136, теорема 2.1.1]). *Если натуральные числа  $k_1$  и  $k_2$  взаимно просты, то  $k_1k_2 - k_1 - k_2$  – это наибольшее натуральное число, которое не представимо в виде комбинации  $k_1$  и  $k_2$  с неотрицательными целыми коэффициентами.*

Лемма 3.1 влечёт, что  $|vc| > n(n-1) - n - (n-1) = n^2 - 3n + 1$ . Допустим, что  $|vc| = n^2 - 3n + 2$ . Тогда имеется путь такой длины из состояния 1 в состояние 2. В ДКА  $\mathcal{W}_n$  любое ребро с началом 1 имеет концом 2, и потому в этом пути за ним должен следовать цикл длины  $n^2 - 3n + 1$ . По лемме 3.1 циклов такой длины не существует. Следовательно,  $|vc| \geq n^2 - 3n + 3$ .

Действие буквы  $b$  на любом множестве состояний  $S \subseteq Q$  не может изменить мощность  $S$ , а действие буквы  $c$  может уменьшить её не более чем на 1. Поэтому слово  $vc$  должно содержать не менее  $n-1$  вхождений буквы  $c$ . Следовательно, длина слова  $v$  над  $\{b, c\}$  не менее  $n^2 - 3n + 2$ , и  $v$  содержит не менее  $n-2$  вхождений буквы  $c$ . Поскольку каждое вхождение  $c$  в  $v$  соответствует вхождению фактора  $ab$  в  $w'$ , мы заключаем, что длина  $w'$  над  $\{a, b\}$  не меньше  $n^2 - 3n + 2 + n - 2 = n^2 - 2n$ . Таким образом,  $|w| = |w'a| \geq n^2 - 2n + 1 = (n-1)^2$ .  $\square$

Определим функцию Черни  $\mathfrak{C}(n)$  как максимум порогов синхронизации синхронизируемых  $n$ -автоматов. Предложение 3.1 даёт неравенство  $\mathfrak{C}(n) \geq (n-1)^2$ . Гипотеза Черни утверждает, что имеет место равенство  $\mathfrak{C}(n) = (n-1)^2$ :

**Гипотеза Черни.** *Любой синхронизируемый ДКА с  $n$  состояниями синхронизируется словом длины  $(n-1)^2$ .*

В обширной литературе по гипотезе Черни и связанным с ней вопросам обычно ссылаются на работу Черни [44] как на источник гипотезы. Эта основополагающая статья примечательна во многих отношениях, но гипотеза **не была** сформулирована в ней. Понятно, что на самом деле мало кто читал статью [44] в словацком оригинале, но сейчас опубликован в открытом доступе её авторизованный английский перевод, см. [45]. Поэтому заинтересованный читатель может убедиться, что Черни ограничивается наблюдением [44, теорема 3], которое в наших обозначениях записывается как

$$(n - 1)^2 \leq \mathfrak{C}(n) \leq 2^n - n - 1, \quad (3)$$

и заключает статью следующим осторожным выводом:

«Между нижней и верхней оценками имеется значительное расхождение, и их необходимо сделать более точными. Можно ожидать, что это будет возможным, в особенности для верхней оценки.»

Представляется, что гипотеза впервые появилась в печати в 1966 г. в заметке Петра Штарке [163], см. также её английский перевод [166]. Штарке улучшил верхнюю оценку в (3) до  $1 + \frac{n(n-1)(n-2)}{2}$ , что стало первой полиномиальной верхней оценкой для  $\mathfrak{C}(n)$ , и завершил [163] замечанием, которое мы воспроизводим с использованием наших обозначений:

«Данное Черни доказательство утверждения (3) в [44] подсказывает гипотезу о том, что  $\mathfrak{C}(n) = (n - 1)^2$  выполняется для любого  $n \geq 1$ . К сожалению, попытки доказать или опровергнуть эту гипотезу оказались безуспешными.»

К этой цитате можно добавить, что ситуация, констатированная в её последней фразе, сохраняется уже 54 года!<sup>11</sup>

Известно, что Черни обсуждал синхронизируемые автоматы (используя термин ‘directable’) и гипотезу о длине их синхронизирующих слов на нескольких чехословацких конференциях, состоявшихся во второй половине 1960-х гг. Некоторые авторы относят гипотезу Черни к его сообщению на Братиславской конференции по кибернетике в 1969 г. Первая публикация Черни [46] (совместная с Алисой Пирицкой и Бланкой Розенауеровой), в которой явно упоминалась гипотеза, появилась в 1971 г. С учётом этого думается, что имя «Гипотеза Черни–Штарке» было бы более справедливым с исторической точки зрения, однако сейчас уже поздно менять терминологию, которая используется более сорока лет.<sup>12</sup> Возвращаясь к оценке вклада Штарке, упомянем ещё его монографию [164], в §I.9 которой обсуждаются результаты из [44] и [163]. Эта книга, а также её английский перевод [165] сыграли важную роль в привлечении внимания исследователей к синхронизируемым автоматам.

### 3.2 Верхние оценки. Оценка Пэна–Франкля

До недавнего времени лучшей верхней оценкой для функции Черни оставалась оценка  $\mathfrak{C}(n) \leq \frac{n^3-n}{6}$ . Для произвольного синхронизируемого  $n$ -автомата синхронизирующее слово длины не больше  $\frac{n^3-n}{6}$  может быть получено с помощью следующего алгоритма.

<sup>11</sup>Время от времени появляются препринты А. Н. Трахтмана, содержащие рассуждения, которые их автор считает доказательством гипотезы Черни, см. [178–180]. По прошествии некоторого срока в рассуждениях обнаруживаются ошибки, и препринты отзываются. На момент написания данной статьи (август 2020 г.) препринт [180] не отозван, но по консенсусному мнению специалистов новых идей по сравнению с отозванными препринтами он, к сожалению, не содержит.

<sup>12</sup>Я пытался найти самое раннее появление имени «Гипотеза Черни» в печати, но не вполне преуспел в этом. Определённо, оно присутствует в литературе с конца 1970-х гг., см., например, [128].

**GREEDYCOMPRESSION( $\mathcal{A}$ )**

1: $w \leftarrow \varepsilon$	▷ Инициализация текущего слова
2: $P \leftarrow Q$	▷ Инициализация текущего множества
3: <b>while</b> $ P  > 1$ <b>do</b>	
4: <b>if</b> $ P \cdot u  =  P $ для всех $u \in \Sigma^*$ <b>then</b>	
5: <b>return</b> Failure	
6: <b>else</b>	
7:     взять кратчайшее слово $v \in \Sigma^*$ со свойством $ P \cdot v  <  P $	
8: $w \leftarrow wv$	▷ Обновление текущего слова
9: $P \leftarrow P \cdot v$	▷ Обновление текущего множества
10: <b>return</b> $w$	

**Алгоритм 1:** Алгоритм нахождения синхронизирующего слова для ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  методом сжатия

Алгоритм 1 ищет синхронизирующее слово «сверху вниз», придерживаясь жадной стратегии: он пытается сжать текущее множество кратчайшим возможным словом. (Мы говорим, что слово  $u \in \Sigma^*$  *сжимает* подмножество  $S \subseteq Q$  в ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , если  $|S \cdot u| < |S|$ .) Оценим время работы алгоритма и длину возвращаемого им слова.

Если  $|Q| = n$ , то основной цикл алгоритма 1 выполняется не более  $n - 1$  раз, так как после каждого прохождения цикла размер текущего множества  $P$  уменьшается по крайней мере на 1. Слово  $v$  в строке 7 – это последовательность меток кратчайшего пути из 2-подмножества множества  $P$  в 1-подмножество множества  $Q$  в автомате  $\mathcal{P}^{\leq 2}(\mathcal{A})$ , см. следствие 2.1 и обсуждение алгоритма Лю–Черни в §2.1. Поиск в ширину находит такой путь за время  $O(n^2 \cdot |\Sigma|)$ . Таким образом, время работы алгоритма 1 полиномиально зависит от размеров ДКА  $\mathcal{A}$ . Детальный временной анализ алгоритма 1 проделал Дэвид Эппштейн в [63, теорема 5]; он показал, что за счёт предвычисления длин путей в автомате  $\mathcal{P}^{\leq 2}(\mathcal{A})$  алгоритм 1 можно исполнить за время  $O(n^3 + n^2 \cdot |\Sigma|)$ . Статья [63], которая содержит и ряд других важных результатов, получила широкую известность, благодаря чему на алгоритм 1 и его варианты часто ссылаются как на *алгоритм Эппштейна*. Сам Эппштейн в [63] указывает на работу Натараджана [121] как на источник алгоритма, а фактически алгоритм появился на 20 лет раньше, ещё в заметке Штарке [163], с которой ни Натараджан, ни Эппштейн не были знакомы.

Чтобы оценить длину возвращаемого алгоритмом 1 слова  $w$ , оценим длины слов  $v$ , дописываемых к  $w$  при каждом прохождении основного цикла.

Рассмотрим типичный шаг, на котором  $|P| = k > 1$ , и пусть  $v = a_1 \cdots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ . Тогда по выбору  $v$  каждое из множеств

$$P_1 := P, P_2 := P_1 \cdot a_1, \dots, P_\ell := P_{\ell-1} \cdot a_{\ell-1} \quad (4)$$

содержит ровно  $k$  состояний. Поскольку  $|P_\ell \cdot a_\ell| = |P \cdot v| < |P| = |P_\ell|$ , найдутся два различных состояния  $p_\ell, p'_\ell \in P_\ell$ , такие, что  $p_\ell \cdot a_\ell = p'_\ell \cdot a_\ell$ . Положим  $R_\ell := \{p_\ell, p'_\ell\}$  и для всех  $i = 1, \dots, \ell - 1$  определим 2-подмножества  $R_i := \{p_i, p'_i\} \subseteq P_i$ , выбирая элементы  $p_i, p'_i$  так, чтобы  $p_i \cdot a_i = p_{i+1}$ ,  $p'_i \cdot a_i = p'_{i+1}$ , см. рис. 16. Если  $R_i \subseteq P_j$  для каких-то  $j < i$ , то для слова  $a_1 \cdots a_j a_i \cdots a_\ell$  длины  $j + (\ell - i) < \ell$  выполняется неравенство  $|P \cdot a_1 \cdots a_j a_i \cdots a_\ell| < |P|$ , что противоречит выбору  $v$  как кратчайшего слова со свойством  $|P \cdot v| < |P|$ . Поэтому  $R_i \not\subseteq P_j$  при всех  $i, j$  таких, что  $1 \leq j < i \leq \ell$ .

Пусть  $k > 1$ . Последовательность  $k$ -подмножеств  $P_1, P_2, \dots$  некоторого множества назовём *2-обновляемой*, если каждое множество  $P_i$  содержит «новое» 2-подмножество, т.е. такое 2-подмножество  $R_i$ , что  $R_i \not\subseteq P_j$  для всех  $j < i$ . Аргумент, приведённый сейчас, доказывает следующее утверждение.

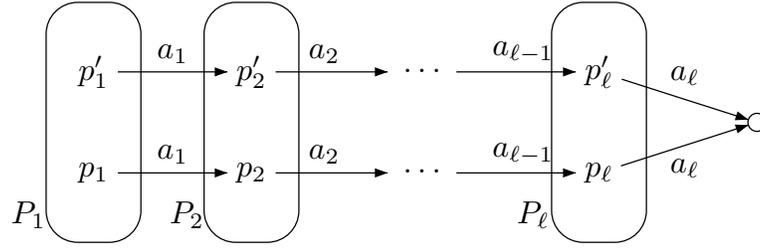


Рис. 16: Комбинаторная конфигурация на типичном шаге алгоритма 1

**Лемма 3.2.** Пусть  $\mathcal{A} = \langle Q, \Sigma \rangle$  – ДКА,  $P$  – подмножество в  $Q$  с  $|P| = k > 1$ , а  $v = a_1 \cdots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , – кратчайшее слово со свойством  $|P \cdot v| < |P|$ . Тогда последовательность  $k$ -подмножеств (4) является 2-обновляемой.

В силу леммы 3.2, чтобы оценить сверху длину слова  $v$  для  $\mathcal{A} = \langle Q, \Sigma \rangle$  с  $|Q| = n$ , достаточно найти максимум длин 2-обновляемых последовательностей  $k$ -подмножеств в  $n$ -множестве как функцию от  $n$  и  $k$ . Это – нетривиальная задача из комбинаторики конечных множеств. Легко построить пример 2-обновляемой последовательности  $k$ -подмножеств длины  $\binom{n-k+2}{2}$ : нужно зафиксировать какие-то  $k - 2$  элемента данного  $n$ -множества и поочередно добавлять к ним 2-подмножества, выбранные из оставшихся  $n - k + 2$  элементов. Но будет ли так построенная последовательность максимальной по длине, далеко не очевидно.

Указанную задачу решил Петер Франкль [69], подтвердивший оптимальность только что описанной конструкции.<sup>13</sup>

**Предложение 3.2.** Максимальная длина 2-обновляемой последовательности  $k$ -подмножеств в  $n$ -множестве равна  $\binom{n-k+2}{2}$ .

Комбинируя лемму 3.2 и предложение 3.2, получаем

**Следствие 3.1.** Пусть  $\mathcal{A} = \langle Q, \Sigma \rangle$  – ДКА,  $P$  – подмножество в  $Q$  с  $|P| = k > 1$ , а  $v \in \Sigma^*$  – кратчайшее слово со свойством  $|P \cdot v| < |P|$ . Тогда  $|v| \leq \binom{n-k+2}{2}$ .

Итак, если  $\ell_k$  – длина слова, которое алгоритм 1 приписывает к текущему слову после шага, на котором текущее множество содержит  $k$  состояний, следствие 3.1 гарантирует, что  $\ell_k \leq \binom{n-k+2}{2}$ . Используя тождество  $\binom{m}{2} + \binom{m}{3} = \binom{m+1}{3}$ , легко вычислить

<sup>13</sup>В [69] рассмотрена и решена более общая задача о максимальной длине (определяемой аналогично)  $m$ -обновляемой последовательности  $k$ -подмножеств в  $n$ -множестве при любом фиксированном  $m \leq k$ .

$$\begin{aligned}
\sum_{k=2}^n \binom{n-k+2}{2} &= \binom{n}{2} + \binom{n-1}{2} + \cdots + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} \\
&= \binom{n}{2} + \binom{n-1}{2} + \cdots + \binom{4}{2} + \binom{3}{2} + \binom{3}{3} \quad (\text{ведь } \binom{2}{2} = 1 = \binom{3}{3}) \\
&= \binom{n}{2} + \binom{n-1}{2} + \cdots + \binom{4}{2} + \binom{4}{3} \\
&= \binom{n}{2} + \binom{n-1}{2} + \cdots + \binom{5}{3} \\
&\dots\dots\dots \\
&= \binom{n}{2} + \binom{n}{3} \\
&= \binom{n+1}{3} = \frac{n^3 - n}{6}.
\end{aligned} \tag{5}$$

Поэтому, суммируя все неравенства  $\ell_k \leq \binom{n-k+2}{2}$  от  $k = 2$  до  $k = n$ , приходим к анонсированной выше оценке

$$\mathfrak{C}(n) \leq \frac{n^3 - n}{6}. \tag{6}$$

В литературе неравенство (6) обычно называют оценкой Пэна–Франкля, ссылаясь на статьи Жана-Эрика Пэна [129] и Франкля [69]. Пэн доказал лемму 3.2 и предположил оценку  $\binom{n-k+2}{2}$  для максимальной длины 2-обновляемой последовательности в своём докладе на коллоквиуме по теории графов и комбинаторике в Марселе в 1981 г.; статья [129] основана на этом докладе. Франкль узнал о предположении Пэна – и немедленно доказал его – во время другого коллоквиума по комбинаторике, проходившего в Билефельде в ноябре 1981 г. Для полноты картины добавим, что впервые оценка (6) возникла в статье Мартина Фишлера и Мейера Танненбаума [68], поданной на симпозиум по теории переключательных схем и автоматов, проходивший в Санта Монике, Калифорния, в октябре 1970 г. В [68] эта оценка выводится из комбинаторной гипотезы, эквивалентной предположению Пэна; Фишлер и Танненбаум приводят несколько частных случаев, в которых им удалось доказать гипотезу, и выражают надежду, что смогут представить полное доказательство на симпозиуме. Однако, по всей видимости, эта надежда не оправдалась – никаких следов доказательства в дальнейших публикациях я не обнаружил. Затем оценка (6) была переоткрыта Зви Кохави и Джозефом Виноградом [103, 104], но её обоснование в этих статьях содержало неисправимые ошибки. В 1987 г. как лемма 3.2, так и предложение 3.2 были ещё раз переоткрыты А. А. Клячко, И. К. Рысцовым и М. А. Спиваком [102], которые были знакомы с работами [68, 103, 104], но не знали про существование статей Пэна и Франкля. Доказательство предложения 3.2 в [102] также покрывает общий случай  $m$ -обновляемых последовательностей  $k$ -подмножеств при любом  $m \leq k$  и основано на той же идее линейаризации, которая использовалась в [69], но изложено на более элементарном языке.

Слово, возвращаемое алгоритмом 1, не всегда будет кратчайшим синхронизирующим словом для данного ДКА. Это неудивительно, поскольку, как уже отмечалось в §2.3, никакой полиномиальный алгоритм, даже недетерминированный, не может вычислить порог синхронизации для произвольного синхронизируемого ДКА.<sup>14</sup> Конкретным примером может служить ДКА  $\mathcal{C}_4$ . На этом автомате алгоритм 1 возвращает слово

<sup>14</sup>Алгоритм 1, строго говоря, не является детерминированным. В общем случае может существовать несколько слов, удовлетворяющих условию в строке 7, и чтобы получить детерминированный вариант алгоритма 1, нужно зафиксировать способ выбора одного из этих слов.

$ab^2abab^3a$  длины 10, складывающееся из меток выделенного жирным шрифтом пути на рис. 17. Как мы знаем, это слово не является кратчайшим словом синхронизирующим

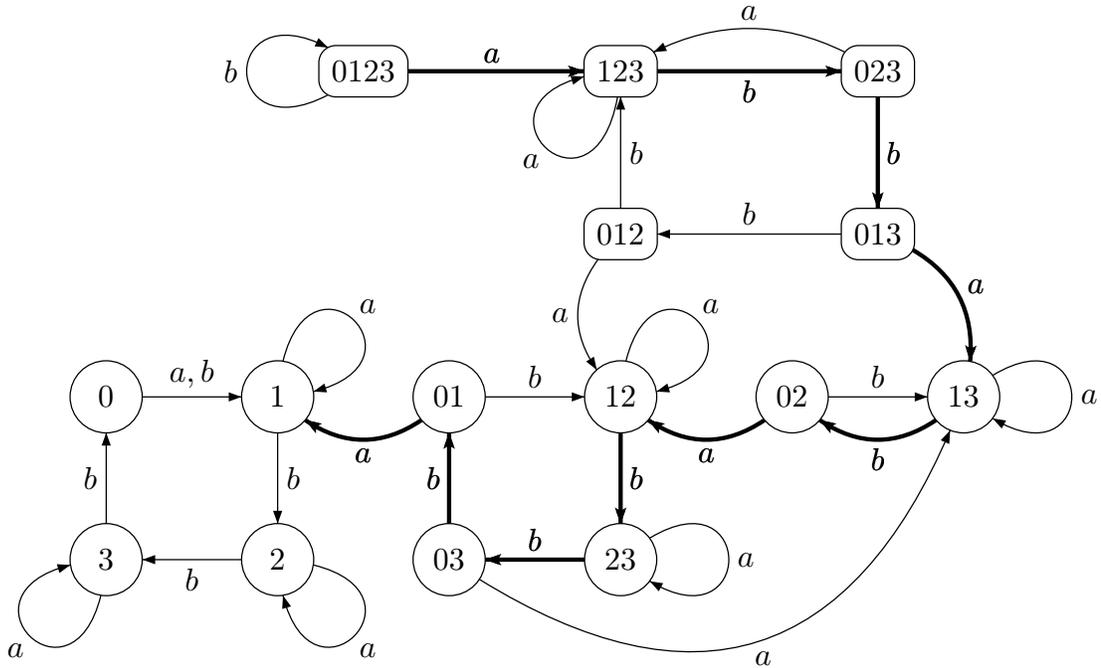


Рис. 17: Результат работы алгоритма 1 на ДКА  $\mathcal{C}_4$

словом для  $\mathcal{C}_4$ . Здесь проявляется одна из основных трудностей проблемы синхронизации – неприменимость стандартных подходов к оптимизации: оптимальное решение **не обязательно** вести себя оптимально на промежуточных этапах. Для  $\mathcal{C}_4$  оптимальным решением является слово  $ab^3ab^3a$ , но его нельзя найти с помощью алгоритма 1, поскольку на втором шаге алгоритм выбирает кратчайшее слово, сжимающее 3-подмножество  $\{1, 2, 3\}$ , т.е. слово  $b^2a$ , в то время как правильным выбором было бы более длинное слово  $b^3a$ , ср. с рис. 11. За свою «поспешность» алгоритм расплачивается на следующем шаге, так как попадает в 2-подмножество  $\{1, 3\}$ , которое не сжимается никаким словом длины меньше 6.

В литературе делались разного рода попытки усовершенствовать алгоритм 1, разрешая ему «заглядывать вперёд» и выбирать на каждом шаге не кратчайшее слово, сжимающее  $k$ -подмножество, а слово, которое оптимизирует какой-нибудь другой разумный параметр, см., например, [74] и обсуждение в [142, §5]. Такие подходы представляют практический интерес, но, конечно, не могут преодолеть теоретический барьер, установленный результатами из §2.3.

Разрыв между порогом синхронизации ДКА и длиной синхронизирующего слова, возвращаемого алгоритмом 1 на этом ДКА, может быть сколь угодно большим. Например, можно подсчитать, что для  $n$ -автомата  $\mathcal{C}_n$  из серии Черни, порог синхронизации которого равен  $(n - 1)^2$  (см. предложение 3.1), алгоритм 1 возвращает слово длины  $\Omega(n^2 \log n)$ . Алгоритм 1 попадает в семейство алгоритмов, аппроксимационные свойства которых исследовались Ананичевым и Гусевым в [7]. Из [7] можно извлечь примеры серий  $n$ -автоматов, на которых относительная погрешность алгоритма 1 составляет  $\Omega(n)$ . Поведение алгоритма 1 в среднем с теоретической точки зрения ещё совсем не изучено; на практике он демонстрирует вполне приемлемые результаты.

Завершая обсуждение оценки Пэна–Франкля (6), отметим ещё, что для  $n = 1, 2, 3$  она даёт такие же значения, как и нижняя оценка  $(n-1)^2$ , поэтому эти три значения (0, 1 и 4 соответственно) точны. Пэн [129] проверил, что для любого синхронизируемого ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  с четырьмя и более состояниями найдется слово  $u \in \Sigma^*$  длины 9, действие которого сжимает множество  $Q$  на три или более состояния:  $|Q \cdot u| \leq |Q| - 3$ . Если начать алгоритм 1 с  $w := u$  и  $P := Q \cdot u$ , то сумма трёх последних слагаемых  $\binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 6 + 3 + 1$  в первой строке выкладки (5) заменится на 9, что уменьшит общую сумму на 1. Поэтому для  $n \geq 4$  оценку (6) можно улучшить до

$$\mathfrak{C}(n) \leq \frac{n^3 - n}{6} - 1. \quad (7)$$

Оценка (6) точна при  $n = 4$ .

### 3.3 Верхние оценки. Недавние продвижения

На протяжении 35 лет оценка Пэна–Франкля оставалась лучшей верхней оценкой для порога синхронизации автоматов с данным числом состояний. В 2011 г. А. Н. Трахтман [177] опубликовал оценку  $\mathfrak{C}(n) \leq \frac{n(7n^2+6n-16)}{48}$ , доказательство которой, однако, содержало неустранимую ошибку. Тем не менее, работа [177] содержала свежую идею, на основе которой Марек Шикуча [172] получил оценку

$$\mathfrak{C}(n) \leq \frac{85059n^3 + 90024n^2 + 196504n - 10648}{511104}. \quad (8)$$

Оценка Шикуча улучшает старший коэффициент  $\frac{1}{6} = 0,166666\dots$  из оценки Пэна–Франкля до  $\frac{85059}{511104} = \frac{28353}{170368} = 0,166422\dots$ . Наименьшее  $n$ , при котором выражение в правой части (8) оказывается меньше, чем  $\frac{n^3 - n}{6}$ , равно 724. Тем не менее, хотя с практической точки зрения полученное улучшение и незначительно, на фоне многолетнего «застоя» результат Шикуча воспринимается как существенное продвижение.

В конце статьи [172] было высказано предположение, что оценка (8) может быть улучшена за счёт более аккуратных вычислений. Это предположение вскоре подтвердилось в заметке Я. Н. Шитова [161], где выведена оценка

$$\mathfrak{C}(n) \leq \left( \frac{7}{48} + \frac{15625}{798768} \right) n^3 + o(n^3) \quad (9)$$

со старшим коэффициентом приблизительно 0,1654.

Для детального обсуждения идеи из [177] и её реализации в [161, 172] потребуются некоторые определения и предварительные результаты.

Пусть дан ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ . Скажем, что слово  $w \in \Sigma^*$  избегает состояние  $q \in Q$ , если  $q \notin Q \cdot w$ . Состояние, для которого существует избегающее слово, называется *избегаемым*. Отметим, что если ДКА  $\mathcal{A}$  синхронизируем и слово  $v$  синхронизирует его к состоянию  $s$ , то  $v$  избегает все состояния, кроме  $s$ . Поэтому в синхронизируемых автоматах все состояния, кроме, быть может, одного, избегаемы. Неизбегаемым в синхронизируемом ДКА может быть только такое состояние  $z$ , что  $z \cdot a = z$  для каждой буквы  $a \in \Sigma$ . Будем называть такие состояния *нулевыми*, а ДКА, в котором есть нулевое состояние, – *автоматом с нулём*. Ясно, что в синхронизируемом ДКА может быть не больше одного нулевого состояния. Более того, в рассмотренных, связанных с гипотезой Черни, можно ограничиться автоматами, в которых нет нулевых состояний. Это следует из двух наблюдений: во-первых, задача установления верхней границы для

порога синхронизации  $n$ -автоматов сводится к двум частным случаям: случаю сильно связанных автоматов и случаю автоматов с нулём, во-вторых, случай автоматов с нулём оказывается несложным.

Сведёние из первого наблюдения – часть фольклора теории синхронизируемых автоматов. Следуя [183], оформим его в удобном для последующих ссылок виде.

**Лемма 3.3** ([183], предложение 2.1). *Пусть  $\mathbf{C}$  – класс ДКА, замкнутый относительно взятия подавтоматов и фактор-автоматов,  $\mathbf{C}_n$  – класс всех  $n$ -автоматов из  $\mathbf{C}$ , а  $f: \mathbb{Z}^+ \rightarrow \mathbb{N}$  – произвольная функция, такая, что*

$$f(n) \geq f(n - m + 1) + f(m) \text{ при всех } n \geq m \geq 1. \quad (10)$$

*Если у каждого сильно связанного синхронизируемого ДКА из  $\mathbf{C}_n$  и каждого синхронизируемого ДКА с нулём из  $\mathbf{C}_n$  порог синхронизации не превышает  $f(n)$ , то это же верно для всех синхронизируемых ДКА из  $\mathbf{C}_n$ .*

*Доказательство.* Пусть  $\mathcal{A} = \langle Q, \Sigma \rangle$  – синхронизируемый ДКА из  $\mathbf{C}_n$ . Рассмотрим множество  $S$  всех состояний, к которым его можно синхронизировать и положим  $m := |S|$ . Пусть  $q \in S$  и  $w \in \Sigma^*$  – слово, которое синхронизирует  $\mathcal{A}$  к  $q$ . Тогда для любой буквы  $a \in \Sigma$  слово  $wa$  синхронизирует  $\mathcal{A}$  к состоянию  $q \cdot a$ , откуда  $q \cdot a \in S$ . Поэтому ограничение функции переходов автомата  $\mathcal{A}$  на множество  $S \times \Sigma$  индуцирует подавтомат  $\mathcal{S}$  с множеством состояний  $S$ . Очевидно, ДКА  $\mathcal{S}$  синхронизируем и сильно связан, и, поскольку класс  $\mathbf{C}$  замкнут относительно взятия подавтоматов, имеем  $\mathcal{S} \in \mathbf{C}_m$ . Следовательно,  $\text{rt}(\mathcal{S}) \leq f(m)$ . Пусть  $v$  – синхронизирующее слово длины  $f(m)$  для  $\mathcal{S}$ .

Рассмотрим теперь разбиение  $\pi$  множества  $Q$  на  $S$  и  $n - m$  одноэлементных классов. Легко видеть, что  $\pi$  – конгруэнция автомата  $\mathcal{A}$ . Фактор-автомат  $\mathcal{A}/\pi$  синхронизируем, и класс  $S$  является его нулевым состоянием. Поскольку класс  $\mathbf{C}$  замкнут относительно взятия фактор-автоматов,  $\mathcal{A}/\pi \in \mathbf{C}_{n-m+1}$ . Следовательно,  $\text{rt}(\mathcal{A}/\pi) \leq f(n - m + 1)$ . Пусть  $u$  – синхронизирующее слово длины  $f(n - m + 1)$  для  $\mathcal{A}/\pi$ . Тогда  $Q \cdot u \subseteq S$  и  $S \cdot v$  – одноэлементное множество. Имеем  $Q \cdot uv \subseteq S \cdot v$ , откуда слово  $uv$  синхронизирует  $\mathcal{A}$ . При этом

$$|uv| = |u| + |v| = f(n - m + 1) + f(m) \leq f(n)$$

согласно (10). Следовательно,  $\text{rt}(\mathcal{A}) \leq f(n)$ .  $\square$

Функция  $(n - 1)^2$  удовлетворяет неравенству (10). Поэтому, применяя лемму 3.3 к классу всех ДКА, получаем, что для доказательства гипотезы Черни её достаточно проверить для сильно связанных синхронизируемых ДКА и для синхронизируемых ДКА с нулём. Давно известно ([149, теорема 1], см. также [143, теорема 6.1]), что для синхронизируемых ДКА с нулём гипотеза Черни верна; более того, для порога синхронизации синхронизируемых ДКА с нулём имеется лучшая, чем  $(n - 1)^2$ , верхняя оценка.

**Предложение 3.3.** *Порог синхронизации синхронизируемого  $n$ -автомата с нулём не превышает  $\frac{n(n-1)}{2}$ , и эта оценка точна.*

*Доказательство.* Пусть  $\mathcal{A} = \langle Q, \Sigma \rangle$  – синхронизируемый  $n$ -автомат с нулевым состоянием  $z$ . Ясно, что любое синхронизирующее слово синхронизирует  $\mathcal{A}$  именно к  $z$ , поэтому из любого состояния  $q \in Q$  есть путь в  $z$ . Для любого подмножества  $S \subseteq Q \setminus \{z\}$  в кратчайшем пути, который начинается в каком-то состоянии из  $S$  и заканчивается в  $z$ , все состояния, кроме первого, лежат в  $Q \setminus S$  и никакое состояние не встречается дважды. Поэтому длина такого пути не превосходит  $|Q \setminus S| = n - |S|$ . Теперь построим синхронизирующее слово  $w$  по следующему алгоритму:

**инициализация:** инициализируем  $w$  пустым словом  $\varepsilon$ ;

**цикл:** до тех пор, пока  $Q \cdot w \neq \{z\}$ , находим кратчайший путь из  $Q \cdot w \setminus \{z\}$  в  $z$  и дописываем к  $w$  слово, составленное из меток этого пути.

После каждого прохождения цикла число состояний в  $Q \cdot w$  становится меньше по крайней мере на 1, поэтому цикл выполняется не более  $n - 1$  раз. Как пояснено выше, длина слова, дописываемого к  $w$  при  $k$ -м прохождении цикла, не превосходит  $n - |Q \cdot w \setminus \{z\}| = n + 1 - |Q \cdot w| \leq n + 1 - (n - k + 1) = k$  для каждого  $k = 1, \dots, n - 1$ . Отсюда

$$|w| \leq 1 + 2 + \dots + (n - 1) = \frac{n(n - 1)}{2}.$$

Итак,  $\text{rt}(\mathcal{A}) \leq \frac{n(n-1)}{2}$ .

Чтобы доказать точность оценки, рассмотрим ДКА  $\mathcal{R}_n := \langle \{0, 1, \dots, n - 1\}, \Sigma \rangle$ , в котором  $\Sigma := \{a_1, \dots, a_{n-1}\}$ . Действие букв понятно из рис. 18: буква  $a_1$  фиксирует все состояния, кроме 1, которое она переводит в 0, а буква  $a_i$  при  $2 \leq i \leq n - 1$  фиксирует все состояния, кроме  $i - 1$  и  $i$ , которые она переставляет. Эта серия автоматов указана И. К. Рысцовым [149, теорема 2], см. также [143, теорема 6.1]. Видно, что  $\mathcal{R}_n$  —  $n$ -ав-

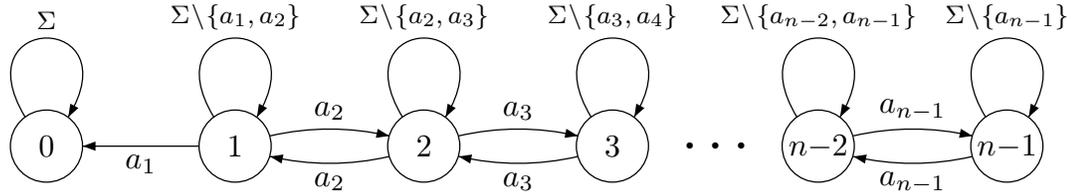


Рис. 18: Автомат  $\mathcal{R}_n$

томат, в котором 0 — нулевое состояние. Поскольку любое состояние можно перевести в 0, автомат  $\mathcal{R}_n$  синхронизируем. Проверим, что  $\text{rt}(\mathcal{R}_n) \geq \frac{n(n-1)}{2}$ .

Пусть  $w$  — синхронизирующее слово для  $\mathcal{R}_n$ . Для  $S = \{s_1, \dots, s_t\} \subseteq Q$ , положим  $f(S) := \sum_{i=1}^t s_i$ . Тогда  $f(\{0\}) = 0$  и  $f(Q) = \frac{n(n-1)}{2}$ . Для любого  $S$  и любой буквы  $a_j$  имеем  $f(S \cdot a_j) \geq f(S) - 1$  поскольку каждая буква либо переставляет два соседних состояния, либо переводит 1 и 0 в 0. С учётом того, что  $Q \cdot w = \{0\}$ , получаем

$$0 = f(\{0\}) = f(Q \cdot w) \geq f(Q) - |w| = \frac{n(n - 1)}{2} - |w|,$$

откуда  $|w| \geq \frac{n(n-1)}{2}$ . □

Размер алфавита в автоматах с нулём из серии  $\mathcal{R}_n$ , демонстрирующей точность оценки из предложения 3.3, растёт вместе с числом состояний. Можно ли достичь того же значения порога синхронизации на сериях автоматов с нулём над **фиксированным** алфавитом? Эта вопрос естествен сам по себе и, кроме того, тесно связан с одной важной задачей теории формальных языков (так называемой *гипотезой Рестиво*, см. обсуждение в [43, §5]). Пока что ответ на него неизвестен; все встречающиеся в литературе «медленно синхронизируемые» серии  $n$ -автоматов с нулём над фиксированным алфавитом имеют порог синхронизации вида  $\frac{1}{4}n^2 + O(n)$ , т.е. примерно вдвое меньший, чем у серии  $\mathcal{R}_n$ , см. [113, 132]. На данный момент дальше всех в этом вопросе продвинулись Д. С. Ананичев и Войтех Ворел [13]: для каждого  $n \equiv 4 \pmod{12}$ , начиная

с  $n = 16$ , они сконструировали синхронизируемый  $n$ -автомат с нулём и двумя входными буквами, порог синхронизации которого равен  $\frac{1}{4}n^2 + 2n - 9$ .

Подытожим: лемма 3.3 и предложение 3.3 показывают, что улучшить оценку Пэна–Франкля достаточно для сильно связных автоматов. До конца этого раздела будем предполагать, что  $\mathcal{A} = \langle Q, \Sigma \rangle$  – сильно связный синхронизируемый  $n$ -автомат. Тогда в  $\mathcal{A}$  любое состояние избегаемо. В [177, лемма 3] утверждается, что для любого состояния в  $\mathcal{A}$  найдётся избегающее слово длины не больше  $n$ . Это утверждение неверно; мы приведём контрпример чуть позже. Дальнейшие рассуждения из [177] в целом корректны. Их идея состоит в том, что можно сжать  $Q$  наполовину, используя избегающие слова следующим образом. Пусть  $v \in \Sigma^*$  таково, что размер подмножества  $S := Q \cdot v$  больше  $\frac{n}{2}$ . Тогда по принципу Дирихле существует состояние  $p \in S$ , для которого есть ровно одно состояние  $q$  со свойством  $p = q \cdot v$ . Если слово  $u$  избегает  $q$ , то по определению  $q \notin Q \cdot u$ , откуда  $p \notin Q \cdot uv \subset Q \cdot v = S$ . Поэтому  $|Q \cdot uv| < |S|$ . (Отметим здесь отличие от алгоритма 1: избегающие слова не дописываются справа, а «предписываются» слева.) Мы приходим к следующему алгоритму.

AVOIDING( $\mathcal{A}$ )

- |  |                                    |
|--|------------------------------------|
| 1: $w \leftarrow \varepsilon$  | ▷ Инициализация текущего слова     |
| 2: $P \leftarrow Q$  | ▷ Инициализация текущего множества |
| 3: <b>while</b> $ P  > \frac{1}{2} Q $ <b>do</b>                                       |                                    |
| 4:   найти $p \in P$ такое, что есть ровно одно $q \in Q$ со свойством $p = q \cdot v$ |                                    |
| 5:   взять кратчайшее слово $u \in \Sigma^*$ , избегающее $q$                          |                                    |
| 6: $w \leftarrow uw$   | ▷ Обновление текущего слова        |
| 7: $P \leftarrow Q \cdot w$  | ▷ Обновление текущего множества    |
| 8: <b>return</b> $w$   |                                    |

**Алгоритм 2:** Алгоритм нахождения слова, сжимающего синхронизируемый автомат  $\mathcal{A} = \langle Q, \Sigma \rangle$  наполовину

Обозначим слово, возвращаемое алгоритмом 2, через  $w_{\frac{1}{2}}$ . По построению для него имеет место  $|Q \cdot w_{\frac{1}{2}}| \leq \frac{n}{2}$ . Чтобы достроить  $w_{\frac{1}{2}}$  до синхронизирующего слова, запустим со слова  $w := w_{\frac{1}{2}}$  и подмножества  $P := Q \cdot w_{\frac{1}{2}}$  алгоритм 1. При этом в силу следствия 3.1 к  $w_{\frac{1}{2}}$  будет дописано слово длины не больше

$$\sum_{k=2}^{\lfloor \frac{n}{2} \rfloor} \binom{n-k+2}{2} = \sum_{k=2}^n \binom{n-k+2}{2} - \sum_{k=\lfloor \frac{n}{2} \rfloor+1}^n \binom{n-k+2}{2} = \binom{n+1}{3} - \binom{\lfloor \frac{n}{2} \rfloor + 2}{3}.$$

(Сумма  $\sum_{k=2}^n \binom{n-k+2}{2}$  вычислена в (5), а сумма  $\sum_{k=\lfloor \frac{n}{2} \rfloor+1}^n \binom{n-k+2}{2}$  вычисляется тем же методом.) Элементарные подсчёты показывают, что

$$\binom{n+1}{3} - \binom{\lfloor \frac{n}{2} \rfloor + 2}{3} = \begin{cases} \frac{n(7n^2 - 6n - 16)}{48} & \text{при чётном } n, \\ \frac{7n^3 - 9n^2 - 31n - 15}{48} & \text{при нечётном } n. \end{cases}$$

Очевидно, что  $7n^3 - 6n^2 - 16n > 7n^3 - 9n^2 - 31n - 15$  при всех натуральных  $n$ , поэтому можно выбрать  $\frac{n(7n^2 - 6n - 16)}{48}$  в качестве годной для любого  $n$  верхней оценки длины слова, дописанного к  $w_{\frac{1}{2}}$  алгоритмом 1. Итак, комбинация алгоритмов 2 и 1 позволяет строить синхронизирующие слова длины не больше

$$|w_{\frac{1}{2}}| + \frac{n(7n^2 - 6n - 16)}{48}. \quad (11)$$

(Теперь читателю должно стать понятней происхождение слагаемого  $\frac{7}{48}$  в коэффициенте при  $n^3$  в оценке (9).)

Ясно, что улучшить оценку Пэна–Франкля с помощью только что описанного построения можно только при наличии хорошей верхней оценки для длины слова  $w_{\frac{1}{2}}$ . Поскольку оно собирается из избегающих слов, вопрос сводится к оценке их длины. Мы уже отмечали, что оценка, заявленная в [177], неверна. Приведём обещанный контрпример; он предложен в [80]. Автомат  $\mathcal{E}_4$ , изображённый на рис. 19, сильно связан, синхронизируем (словом  $ab^2abab^2a$ , например) и имеет 4 состояния. Однако можно проверить, что кратчайшее слово, избегающее состояния 0, – это слово  $abbaba$  длины 6.

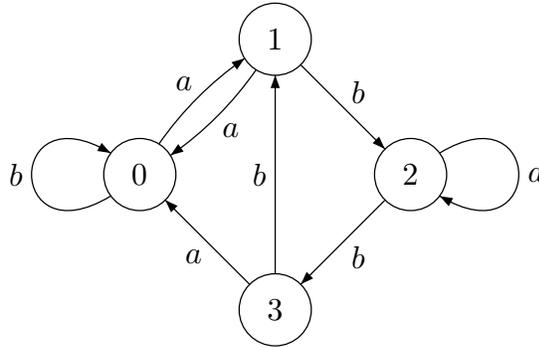


Рис. 19: Автомат  $\mathcal{E}_4$  (контрпример к лемме 3 из [177])

Что же можно сказать о длине избегающих слов? Для удобства обсуждений введём термин *порог избегаемости* и обозначение  $\text{at}(\mathcal{A})$  для максимума длин кратчайших слов, избегающих состояния сильно связанного синхронизируемого ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ :

$$\text{at}(\mathcal{A}) := \max_{q \in Q} \{ |w_q| : w_q \text{ – кратчайшее слово над } \Sigma, \text{ избегающее } q \}.$$

Легко видеть, что  $\text{at}(\mathcal{A}) \leq \text{rt}(\mathcal{A}) + 1$ . Действительно, если  $w$  – кратчайшее синхронизирующее слово для  $\mathcal{A}$ , то  $w$  избегает все состояния, кроме состояния  $s$ , к которому оно синхронизирует  $\mathcal{A}$ . Поскольку  $\mathcal{A}$  сильно связан,  $s$  не является нулевым состоянием, т.е.  $s.a \neq s$  для некоторой буквы  $a \in \Sigma$ . Поэтому слово  $wa$  длины  $\text{rt}(\mathcal{A}) + 1$  избегает  $s$ .

Понятно, что оценка  $\text{at}(\mathcal{A}) \leq \text{rt}(\mathcal{A}) + 1$  не представляет интереса с точки зрения описанного выше подхода. Вопрос о существовании линейной от  $n$  верхней оценки на порог избегаемости в явном виде поставлен в [80]; пока он остаётся открытым. Основываясь на экспериментальных данных, Анджей Кисилевич, Якуб Ковальский и Марек Шикуня выдвинули следующее предположение:

**Гипотеза ККШ** ([98], гипотеза 5). *Порог избегаемости любого сильно связанного синхронизируемого автомата с  $n$  состояниями не превышает  $2n - 2$ .*

Предполагается, что гипотетическая верхняя оценка  $2n - 2$  точна для  $n$ -автоматов с тремя входными буквами при  $n \geq 4$ , см. [98]. В диссертации Ворела [185, теорема 3.26] для любого  $n \geq 5$  указан сильно связанный синхронизируемый  $n$ -автомат с двумя входными буквами и порогом избегаемости  $2n - 3$ .

Заметим, что если линейная оценка на порог избегаемости существует, то длина слова  $w_{\frac{1}{2}}$ , возвращаемого алгоритмом 2, оценивается сверху квадратичной функцией

от  $n$ . Действительно, основной цикл алгоритма 2 выполняется не более  $\lceil \frac{n}{2} \rceil$  раз, так как каждое прохождение цикла уменьшает размер текущего множества по крайней мере на 1. Следовательно,  $w_{\frac{1}{2}}$  собирается из не более чем  $\lceil \frac{n}{2} \rceil$  избегающих слов. При квадратичной верхней оценке на  $|w_{\frac{1}{2}}|$  выражение (11) даёт оценку вида  $\frac{7}{48}n^3 + O(n^2)$  для порога синхронизации сильно связанных синхронизируемых  $n$ -автоматов.

Пока, однако, лучшая из известных верхних оценок на порог избегаемости даётся выражением  $n^2 - 3n + 4$ , см. [172]. Такая оценка не позволяет ограничить длину слова, возвращаемого алгоритмом 2, настолько, чтобы улучшить оценку Пэна–Франкля через выражение (11). Поэтому в [172] слово, сжимающее автомат наполовину, строится более хитрым способом. Он опирается на следующее ключевое соображение:

**Лемма 3.4** ([172], лемма 2). *Пусть  $\mathcal{B} = \langle R, \Sigma \rangle$  – произвольный ДКА,  $S$  – непустое подмножество в  $R$ , а  $T$  – непустое собственное подмножество в  $S$ . Если  $T \not\subseteq S \cdot u$  для некоторого  $u \in \Sigma^*$ , то существует слово  $v \in \Sigma^*$  длины не больше  $|R| - |T|$  такое, что либо  $|S \cdot v| < |S|$ , либо  $T \not\subseteq S \cdot v$ .*

Объясним на идейном уровне, как работает лемма 3.4, отсылая к [172] за (довольно громоздкими) техническими деталями. Пусть, как и выше,  $\mathcal{A} = \langle Q, \Sigma \rangle$  – сильно связный синхронизируемый  $n$ -автомат. Допустим, что уже построен некий фрагмент  $w$  искомого «уполовинивающего» слова и множество  $S := Q \cdot w$  содержит  $k$  состояний, где  $\frac{n}{2} < k < n$ . Если  $u$  – слово, которое синхронизирует  $\mathcal{A}$  к состоянию из  $Q \setminus S$ , то  $q \notin S \cdot u$  для любого  $q \in S$ . Поэтому условия леммы 3.4 выполняются для любой пары множеств  $S$  и  $T := \{q\}$ , где  $q \in S$ . Применяя лемму ко всем таким парам множеств, получаем, что либо (случай 1) существует слово  $v \in \Sigma^*$  длины не больше  $n - 1$  такое, что  $|S \cdot v| < |S|$ , либо (случай 2) для **каждого** состояния  $q \in S$  найдётся слово  $v_q$  длины не больше  $n - 1$ , такое, что  $q \notin S \cdot v_q$ . В случае 1 множество  $S$  сжато словом линейной от  $n$  длины (а не квадратичной, как при применении следствия 3.1). В случае 2 слово  $wv_q$  избегает состояние  $q \in S$ , а слово  $w$  избегает каждое состояние из  $Q \setminus S$ . Поэтому порог избегаемости ДКА  $\mathcal{A}$  не превышает  $|w| + n - 1$ .

Теперь понятно, как лемма 3.4 может помочь построить слово, сжимающее  $\mathcal{A}$  наполовину и более короткое, чем  $w_{\frac{1}{2}}$ . Если в процессе сжатия случай 2 произойдёт рано, когда длина построенного фрагмента  $w$  ещё мала, получается сильная верхняя оценка на  $\text{at}(\mathcal{A})$ , которую можно использовать для дальнейшего сжатия методом алгоритма 2. Если же случай 2 впервые случится поздно, т.е. при  $k$ , близком к  $\frac{n}{2}$ , это значит, что автомат сжат до  $k$  состояний словом длины не больше  $(n - k)(n - 1)$ .

В действительности приведённые соображения ещё недостаточны, чтобы улучшить оценку Пэна–Франкля, – можно подсчитать, что если строить сжимающее наполовину слово, опираясь только на них, получится слово длины  $\frac{n^3}{16} + O(n^2)$ . Нам же нужно слово, в выражении для длины которого коэффициент при  $n^3$  строго меньше чем  $\frac{1}{6} - \frac{7}{48} = \frac{1}{48}$ . (Здесь  $\frac{1}{6}$  и  $\frac{7}{48}$  – коэффициенты при  $n^3$  в оценке Пэна–Франкля и соответственно в оценке длины слова, которое «дожимает» автомат с  $\lfloor \frac{n}{2} \rfloor$  состояний до одного состояния.) К счастью, применяя лемму 3.4 во всей её общности и используя некоторые дополнительные ухищрения, построить такое слово удаётся.

Улучшение оценки (8) до (9), достигнутое в [161], опирается на некоторое уточнение леммы 3.4. Отметим, что явный вид слагаемого  $o(n^3)$  из (9) в [161] не указан; сказано только, что с помощью прямых вычислений можно довести оценку этого слагаемого до  $O(n^2 \log n)$ . Поэтому, хотя оценка (9) асимптотически лучше, чем (8), пока неясно, на автоматах каких размеров начинается её преимущество.

В [172] показано, что слово, синхронизирующее данный синхронизируемый  $n$ -автомат и имеющее длину не больше  $\frac{85059n^3 + 90024n^2 + 196504n - 10648}{511104}$ , можно найти за по-

линомиальное от  $n$  время. По-видимому, то же верно и для синхронизирующих слов, строящихся в [161], хотя в последней работе это не отмечено явно.

Подводя итоги, можно констатировать, что идея сжатия с помощью избегающих слов сдвинула задачу о пороге синхронизации  $n$ -автоматов с мёртвой точки и в то же время породила новые задачи о синхронизируемых автоматах. Однако эта идея недостаточна для доказательства гипотезы Черни или даже для получения оценки вида  $o(n^3)$  на функцию  $\mathfrak{C}(n)$ .

### 3.4 Метод расширения

Сжимающие стратегии, реализованные в алгоритмах 1 и 2, дают лучшие из известных на данный момент верхних оценок для функции Черни в случае произвольных ДКА. Однако наиболее впечатляющие частичные результаты, доказывающие гипотезу Черни для некоторых специальных классов автоматов, были получены с помощью другого алгоритма, строящего синхронизирующее слово снизу вверх. Чтобы описать этот алгоритм, введём одно обозначение.

Для ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , подмножества  $P \subseteq Q$  и слова  $w \in \Sigma^*$  обозначим через  $Pw^{-1}$  полный прообраз  $P$  относительно действия  $w$ :

$$Pw^{-1} := \{q \in Q : q.w \in P\}.$$

Отметим, что  $P(uv)^{-1} = (Pv^{-1})u^{-1}$ . Для  $q \in Q$  договоримся писать  $qw^{-1}$  вместо  $\{q\}w^{-1}$ . Тот факт, что слово  $w \in \Sigma^*$  синхронизирует автомат  $\mathcal{A}$  к состоянию  $s$  в этих обозначениях запишется равенством  $sw^{-1} = Q$ .

GREEDYEXTENSTION( $\mathcal{A}$ )

```

1: if  $|qa^{-1}| = 1$  для всех  $q \in Q$  и  $a \in \Sigma$  then
2:   return Failure
3: else
4:    $w \leftarrow a$ , где  $q$  и  $a$  таковы, что  $|qa^{-1}| > 1$   $\triangleright$  Инициализация текущего слова
5:    $P \leftarrow qa^{-1}$   $\triangleright$  Инициализация текущего множества
6: while  $|P| < |Q|$  do
7:   if  $|Pu^{-1}| \leq |P|$  для всех  $u \in \Sigma^*$  then
8:     return Failure
9:   else
10:    взять кратчайшее слово  $v \in \Sigma^*$  со свойством  $|Pv^{-1}| > |P|$ 
11:     $w \leftarrow vw$   $\triangleright$  Обновление текущего слова
12:     $P \leftarrow Pv^{-1}$   $\triangleright$  Обновление текущего множества
13: return  $w$ 

```

**Алгоритм 3:** Алгоритм нахождения синхронизирующего слова для ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  методом расширения

Назовём собственное подмножество  $P \subset Q$  расширяемым, если существует слово  $v \in \Sigma^*$ , для которого  $|Pv^{-1}| > |P|$ ; такое слово называется расширяющим для подмножества  $P$ . Алгоритм 3 собирает синхронизирующее слово из расширяющих слов в противоположность алгоритму 1, который работает со сжимающими словами. В то время как нахождение сжимающих слов сводится к поиску в автомате 2-подмножеств и потому реализуемо за полиномиальное от размера автомата время, задача о существовании слова, расширяющего данное собственное подмножество, PSPACE-полна даже для синхронизируемых ДКА, см. [27, предложение 5]. Мы обсуждали в §3.3, что

в круге вопросов, связанных с гипотезой Черни, можно ограничиться рассмотрением сильно связанных автоматов. В сильно связанном синхронизируемом ДКА расширяемо любое собственное подмножество, но задача нахождения кратчайшего слова, расширяющего данное подмножество, остаётся вычислительно трудной и для таких автоматов, см. [27, следствие 26]. Поэтому неясно, допускает ли алгоритм 3 полиномиальную от размера автомата имплементацию. Некоторые типы ДКА, для которых такая полиномиальная имплементация возможна, рассмотрены в [28]. Анализ поведения алгоритма 3 в среднем случае представляется важной и интересной исследовательской проблемой, по которой пока нет никаких теоретических результатов, но имеются некоторые экспериментальные данные, см. [142].

В общем случае неизвестна никакая нетривиальная верхняя оценка на длину слов, которые основной цикл алгоритма 3 добавляет к текущему слову. Однако можно выделить некоторые случаи, в которых имеются достаточно сильные оценки на эту длину. Назовём  $n$ -автомат  $\alpha$ -расширяемым, где  $\alpha$  – положительная константа, если в нём каждое собственное неоднородное множество состояний расширяемо словом длины не более  $\alpha n$ . Следующее фольклорное наблюдение объясняет значение этого свойства.

**Предложение 3.4.** *Каждый  $\alpha$ -расширяемый  $n$ -автомат  $\mathcal{A} = \langle Q, \Sigma \rangle$  с  $n > 2$  синхронизируем и его порог синхронизации не превышает  $1 + \alpha n(n - 2)$ . В частности, гипотеза Черни верна для 1-расширяемых автоматов.*

*Доказательство.* При  $n > 2$  у множества  $Q$  есть собственные неоднородные подмножества. Если  $S \subset Q$  – такое подмножество и  $v \in \Sigma^*$  – слово, расширяющее  $S$ , есть состояние  $s \in S$ , для которого  $|sv^{-1}| > 1$ . Если  $v = a_1 \cdots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , то пусть  $i \in \{1, \dots, n\}$  – наибольший номер, для которого  $|s(a_i a_{i+1} \cdots a_\ell)^{-1}| > 1$ . Тогда можно взять букву  $a_i$  и (единственное) состояние из множества  $s(a_{i+1} \cdots a_\ell)^{-1}$  в качестве пары  $(q, a)$ , необходимой для инициализации в строке 4 алгоритма 3.

Алгоритм 3 проходит основной цикл не более  $n - 2$  раз, поскольку цикл начинается при  $|P| \geq 2$ , а каждое прохождение цикла увеличивает  $|P|$  по крайней мере на 1. При этом всякий раз добавляется слово длины не более  $\alpha n$ . Следовательно, длина синхронизирующего слова, возвращаемого алгоритмом, не превышает  $1 + \alpha n(n - 2)$ . При  $\alpha = 1$  получается оценка  $1 + n(n - 2) = (n - 1)^2$  из гипотезы Черни.  $\square$

Подход к гипотезе Черни через 1-расширяемость возник в статье Пэна [128]. Пэн показал, что каждый ДКА с простым числом состояний, в котором одна из букв циклически переставляет состояния, а какая-то другая буква сжимает множество состояний, 1-расширяем. Луи Дюбюк [61] развил этот результат Пэна, доказав 1-расширяемость синхронизируемого автомата, в котором некоторая буква действует как циклическая перестановка множества всех состояний. (Автоматы с этим свойством называются *циклическими*.) Яркко Кари [95] установил 1-расширяемость эйлеровых синхронизируемых автоматов. (Напомним, что мы применяем понятия и термины теории графов к автоматам, подразумевая, что они применяются к их носителям. Граф называется *эйлеровым*, если в нём есть цикл, содержащий каждое ребро ровно один раз.)

Итак, 1-расширяемость позволяет подтвердить гипотезу Черни в нескольких интересных частных случаях. Однако в общем случае доказать эту гипотезу через 1-расширяемость невозможно, поскольку существуют сильно связанные синхронизируемые автоматы, не являющиеся 1-расширяемыми. Первым таким примером был 6-автомат  $\mathcal{H}_6$ , открытый Кари [94]; он изображен на рис. 20, из которого понятно действие букв. Этот автомат синхронизируем, его кратчайшее синхронизирующее слово имеет длину 25. Кари нашёл  $\mathcal{H}_6$  как контрпример к обобщению гипотезы Черни, предложенному в

диссертации Пэна [127], но этот автомат обладает и другими примечательными свойствами. В частности, можно проверить, что в  $\mathcal{K}_6$  никакое множество из более чем

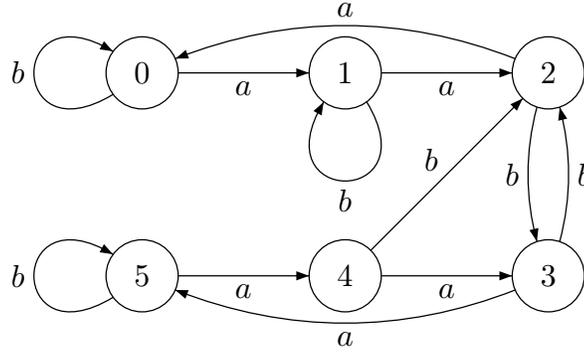


Рис. 20: Автомат  $\mathcal{K}_6$

четырёх состояний не будет полным прообразом множества  $\{2, 3, 4, 5\}$  относительно действия слова длины 6 (как требуется в определении 1-расширяемости) или даже 7.

2-расширяемость (и следовательно, в силу предложения 3.4, квадратичная верхняя оценка на порог синхронизации) была доказана для ряда классов синхронизируемых ДКА, см. [79, 145, 146, 148]. Мари-Пьер Беал, М. В. Берлинков и Доминик Перрэн [18, 19] установили, что это свойство в слегка ослабленной, но достаточной для квадратичной верхней оценки на порог синхронизации форме выполнено для синхронизируемых однокластерных автоматов. ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *однокластерным*, если есть такие  $q \in Q$  и  $a \in \Sigma$ , что из любого состояния найдётся путь в  $q$ , рёбра которого помечены только буквой  $a$ . (Например, автоматы  $\mathcal{C}_n$  и  $\mathcal{W}_n$ , изображенные на рис. 15, однокластерные, а автомат Кари  $\mathcal{K}_6$  нет. Однокластерными являются и такие важные для приложений ДКА, как декодеры конечных максимальных префиксных кодов, см. §1.4.) Для буквы  $a$  из определения однокластерности в  $\mathcal{A}$  есть ровно один простой цикл, рёбра которого помечены  $a$ ; назовём его *a-циклом*. Если обозначить через  $C$  множество вершин  $a$ -цикла, то несложно понять, что  $Q \cdot a^{|Q|-|C|} = C$ . Это позволяет модифицировать алгоритм 3 следующим образом.

RELATIVEEXTENSTION( $\mathcal{A}, C, a$ )

- |  |                                    |
|--|------------------------------------|
| 1: $w \leftarrow \varepsilon$  | ▷ Инициализация текущего слова     |
| 2: $P \leftarrow \{q\}$ , где $q \in C$  | ▷ Инициализация текущего множества |
| 3: <b>while</b> $ P  <  C $ <b>do</b>  |                                    |
| 4: <b>if</b> $ Pu^{-1} \cap C  \leq  P $ для всех $u \in \Sigma^*$ <b>then</b>       |                                    |
| 5: <b>return</b> Failure   |                                    |
| 6: <b>else</b>   |                                    |
| 7:     взять кратчайшее слово $v \in \Sigma^*$ со свойством $ Pv^{-1} \cap C  >  P $ |                                    |
| 8: $w \leftarrow vw$   | ▷ Обновление текущего слова        |
| 9: $P \leftarrow Pv^{-1} \cap C$   | ▷ Обновление текущего множества    |
| 10: <b>return</b> $a^{ Q - C }w$   |                                    |

**Алгоритм 4:** Модифицированный алгоритм нахождения синхронизирующего слова для однокластерного ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  с  $a$ -циклом  $C$  методом расширения

В [18, 19] показано, что длина слова, добавляемого при каждом прохождении основного цикла алгоритма 4, не превышает  $2n$ , где  $n := |Q|$ , что влечёт верхнюю оценку

вида  $2n^2 + o(n^2)$  на  $\text{rt}(\mathcal{A})$ . Аналогичный результат получили Артуро Карпи и Флавио Д'Алессандро [40, 41]. Стейнберг [168, 169] обобщил описанный выше подход и немного улучшил явное выражение для члена  $o(n^2)$  в оценках из [18, 19, 40, 41]. А именно, Стейнберг проверил, что  $\text{rt}(\mathcal{A}) \geq 2n^2 - 9n + 14$ , где, как и выше,  $\mathcal{A}$  – однокластерный синхронизируемый  $n$ -автомат. В [169] Стейнберг также доказал гипотезу Черни для однокластерных ДКА, у которых размер  $a$ -цикла – простое число. В [42] Карпи и Д'Алессандро дали верхнюю оценку с лучшей на сегодняшний день асимптотикой, установив, что порог синхронизации однокластерного синхронизируемого  $n$ -автомата не превышает  $2n^2 - 4n + 1 - 2(n - 1) \ln \frac{n}{2}$ .

Старший коэффициент 2 в первоначальных оценках из [18, 19, 40, 41] устоял и при последующих улучшениях, полученных в рамках метода расширения. Эту закономерность объясняет результат Берлинкова [22]: он указал серию синхронизируемых однокластерных ДКА  $\mathcal{B}_n = \langle \{0, 1, \dots, n - 1\}, \{a, b\} \rangle$ , в которой для каждого  $\alpha < 2$  есть автоматы, не являющиеся  $\alpha$ -расширяемыми. Автомат  $\mathcal{B}_n$  изображен на рис. 21. Дей-

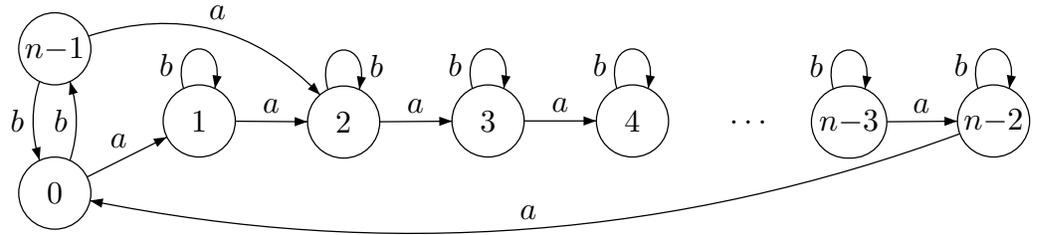


Рис. 21: Автомат  $\mathcal{B}_n$

ствие букв в  $\mathcal{B}_n$  определено так:

$$i \cdot a := \begin{cases} i + 1 \pmod{n - 1}, & \text{если } i < n - 1, \\ 2, & \text{если } i = n - 1; \end{cases} \quad i \cdot b = \begin{cases} n - 1, & \text{если } i = 0, \\ i, & \text{если } 0 < i < n - 1, \\ 0, & \text{если } i = n - 1. \end{cases}$$

Можно проверить, что в  $\mathcal{B}_n$  кратчайшим расширяющим словом для подмножества  $\{0, n - 1\}$  является слово  $a^{n-2}ba^{n-2}$  длины  $2n - 3$ . Поэтому при любом  $n > \frac{3}{2-\alpha}$  автомат  $\mathcal{B}_n$  не будет  $\alpha$ -расширяемым.

2-расширяемость была установлена и для некоторых других типов синхронизируемых автоматов, см. [42, 145, 146, 148]. Однако надежды на то, что квадратичную от  $n$  оценку на порог синхронизации произвольных синхронизируемых  $n$ -автоматов можно получить, доказав их  $\alpha$ -расширяемость для какого-то  $\alpha$  и применив предложение 3.4, не оправдались. Киселевич и Шикун [100] построили серию синхронизируемых ДКА  $\mathcal{KS}_{2m-1} = \langle \{1, 2, \dots, 2m - 1\}, \{a, b\} \rangle$ , в которой для каждого  $\alpha$  найдётся автомат, не являющийся  $\alpha$ -расширяемым. Автомат  $\mathcal{KS}_{2m-1}$  изображен на рис. 22. Действие букв в нём определено так:

$$i \cdot a := \begin{cases} 1, & \text{если } i = m, \\ m + 1, & \text{если } i = 2m - 1, \\ i + 1 & \text{в остальных случаях;} \end{cases} \quad i \cdot b = \begin{cases} i, & \text{если } 1 \leq i \leq m - 1, \\ 2m - 1, & \text{если } i = m, \\ i - m & \text{в остальных случаях.} \end{cases}$$

Можно проверить, что в  $\mathcal{KS}_{2m-1}$  длина кратчайшего слова, расширяющего подмножество  $\{m + 1, \dots, 2m - 1\}$  (т.е. «верхнюю строку» на рис. 22), равна  $2 + m \lceil \frac{m-2}{2} \rceil$ . Последнее выражение растёт быстрее любой линейной функции от числа  $2m - 1$ , поэтому для любого  $\alpha$  можно подобрать  $m$ , начиная с которого автоматы  $\mathcal{KS}_{2m-1}$  не будут  $\alpha$ -расширяемыми.

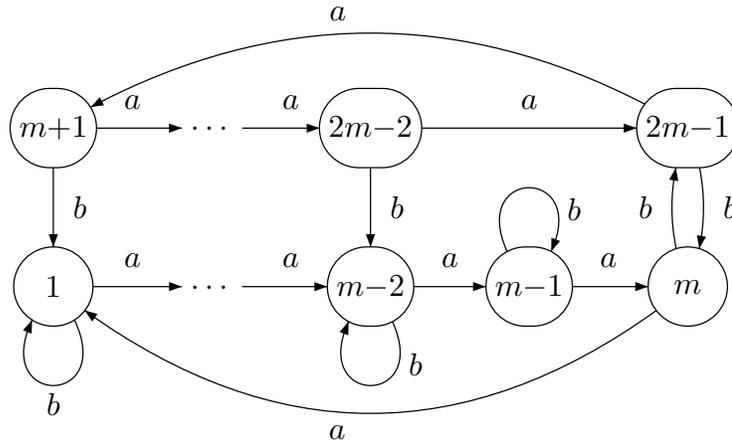


Рис. 22: Автомат  $\mathcal{KS}_{2m-1}$

Завершая обсуждение метода расширения, упомянем ещё недавнюю статью [85], результаты которой в некотором смысле уменьшают «зазор» между сжимающим алгоритмом 1 и расширяющим алгоритмом 3. А именно, в [85] продемонстрировано, что для большинства типов автоматов, рассмотренных в этом разделе, т.е. для ДКА, для которых хорошо работает метод расширения, сжимающий алгоритм работает лучше, чем в общем случае. Так, для синхронизируемых циклических ДКА алгоритм 1 возвращает синхронизирующее слово длины не больше  $n^2 \ln n$ , см. [85, теорема 13].

### 3.5 Сводка частичных результатов

Этот раздел носит справочный характер. В нём собраны основные результаты, устанавливающие справедливость ограничения гипотезы Черни на различные классы ДКА. В дополнение к этому перечислены классы автоматов, для которых гипотеза Черни не доказана, но известна квадратичная от числа состояний верхняя оценка на порог синхронизации. Некоторые из обсуждаемых в этом разделе результатов уже упоминалось выше; мы сочли уместным привести их здесь для полноты сводного списка. Сведения, относящиеся к каждому классу автоматов, организованы в следующем формате:

- *определение* класса,
- *верхняя оценка* для порога синхронизации синхронизируемых автоматов с  $n$  состояниями в данном классе,
- *метод* получения верхней оценки,
- *нижняя оценка* для функции Черни (максимума порогов синхронизации синхронизируемых автоматов с  $n$  состояниями) в данном классе,
- *комментарии* (при необходимости).

В случаях, когда верхняя оценка равна нижней, т.е. ограничение функции Черни на соответствующий класс ДКА известно, к метке класса добавлен верхний индекс †.

#### А. Типы ДКА с порогом синхронизации $(n - 1)^2$ или близким к $(n - 1)^2$

A1†. Циклические автоматы

- *Определение*: ДКА называется *циклическим*, если одна из его букв действует как циклическая перестановка множества всех состояний.

- *Верхняя оценка:*  $(n - 1)^2$ , см. [61].
- *Метод:* 1-расширяемость, см. §3.4.
- *Нижняя оценка:*  $(n - 1)^2$ , поскольку автоматы  $\mathcal{C}_n$  из серии Черни циклические.

## A2. Однокластерные автоматы с циклом простой длины

• *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *однокластерным*, если есть такие  $q \in Q$  и  $a \in \Sigma$ , что из любого состояния найдётся путь в  $q$ , рёбра которого помечены только буквой  $a$ . Для всякой такой буквы  $a$  в  $\mathcal{A}$  есть ровно один простой цикл, рёбра которого помечены  $a$ ; к рассматриваемому здесь типу относятся однокластерные ДКА, в которых для какой-то буквы этот цикл содержит простое число состояний.

- *Верхняя оценка:*  $(n - 1)^2$ , см. [169].
- *Метод:* 2-расширяемость, см. §3.4, с некоторыми дополнительными техническими ухищрениями.

• *Нижняя оценка:* если  $n$  – простое число, то  $(n - 1)^2$ , поскольку тогда автомат  $\mathcal{C}_n$  из серии Черни относится к рассматриваемому типу; для составных  $n$  точная нижняя оценка неизвестна.

## A3<sup>†</sup>. Ориентируемые автоматы

• *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *ориентируемым*,<sup>15</sup> если состояния из  $Q$  можно расположить в циклическом порядке<sup>16</sup>  $q_0, q_1, \dots, q_{n-1}$ , где  $n = |Q|$ , так, чтобы действие букв из  $\Sigma$  сохраняло этот циклический порядок. Чтобы строго определить последнее условие, скажем, что последовательность  $p_0, p_1, \dots, p_{m-1}$  (не обязательно различных) состояний из  $Q$  *правильно ориентирована*, если оставив в ней ровно одно состояние из каждого блока  $p_i = p_{i+1 \pmod{m}} = \dots = p_{i+k \pmod{m}}$  одинаковых соседних состояний, получим подпоследовательность некоторой циклической перестановки последовательности  $q_0, q_1, \dots, q_{n-1}$ . Буква  $a \in \Sigma$  *сохраняет циклический порядок*, если последовательность образов  $q_0 \cdot a, q_1 \cdot a, \dots, q_{n-1} \cdot a$  правильно ориентирована.

- *Верхняя оценка:*  $(n - 1)^2$ , см. [63].

• *Метод:* основан на следующем свойстве ориентируемых автоматов: для любого слова  $w \in \Sigma^*$  и любого интервала  $I \subseteq Q$  полный прообраз  $Iw^{-1}$  является интервалом,<sup>17</sup> см. [63, лемма 1]. Если  $w := a_1 \dots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , – синхронизирующее слово, возвращаемое алгоритмом З, а  $q \in Q$  – состояние, с которого алгоритм З начинает работу, то все подмножества  $q(a_i a_{i+1} \dots a_\ell)^{-1}$  различны и являются неоднородными интервалами. (Иллюстрацией может служить рисунок 11, где видно, что подъём от состояния  $q := 1$  до множества  $Q := \{0, 1, 2, 3\}$  происходит исключительно через интервалы.) Нетрудно подсчитать, что в  $n$ -элементном циклически упорядоченном множестве имеется  $(n - 1)^2$  неоднородных интервалов, откуда  $\ell \leq (n - 1)^2$ .

- *Нижняя оценка:*  $(n - 1)^2$ , поскольку автоматы  $\mathcal{C}_n$  из серии Черни ориентируемые.

<sup>15</sup>Понятие восходит к статье Дэвида Эппштейна [63], где такие автоматы именовались монотонными. Мы следуем терминологии из [9], а термин «монотонный» резервируем для более узкого класса ДКА.

<sup>16</sup>Хотя циклический порядок на конечном множестве – интуитивно ясное понятие, для полноты напомним формальное определение. *Циклический порядок* на множестве  $Q$  – это такое тернарное отношение  $\Upsilon \subseteq Q \times Q \times Q$ , что для любых  $p, q, r \in Q$ :

- если  $(p, q, r) \in \Upsilon$ , то  $(q, r, p) \in \Upsilon$ ;
- если  $(p, q, r) \in \Upsilon$ , то  $(r, q, p) \notin \Upsilon$ ;
- если  $(p, q, r) \in \Upsilon$  и  $(p, r, u) \in \Upsilon$ , то  $(p, q, u) \in \Upsilon$ ;
- если  $p, q$  и  $r$  различны, то либо  $(p, q, r) \in \Upsilon$ , либо  $(r, q, p) \in \Upsilon$ .

<sup>17</sup>Непустое подмножество  $I$  циклически упорядоченного множества  $Q$  называется *интервалом*, если вместе с любыми различными  $p, r \in I$  в  $I$  лежат все такие  $q \in Q$ , что последовательность  $p, q, r$  правильно ориентирована. В силу этого определения одноэлементные подмножества – интервалы.

- *Комментарии:* В [9, теорема 2.2] верхняя оценка из [63] распространена на более широкий класс синхронизируемых полуориентируемых автоматов. ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  *полуориентируем*, если состояния из  $Q$  можно расположить в циклическом порядке  $q_0, q_1, \dots, q_{n-1}$ , где  $n = |Q|$ , и для каждой буквы  $a \in \Sigma$  одна из последовательностей  $q_0 \cdot a, q_1 \cdot a, \dots, q_{n-1} \cdot a$  и  $q_{n-1} \cdot a, q_{n-2} \cdot a, \dots, q_0 \cdot a$  правильно ориентирована. Ещё одно, более громоздко формулируемое обобщение обсуждается в пункте А4.

#### А4<sup>†</sup>. Автоматы, сохраняющие интервалы графов

- *Определение:* Пусть  $\Delta$  – граф, множество вершин которого есть множество  $Q$  состояний ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ . (Предостережение: граф  $\Delta$ , вообще говоря, отличен от носителя ДКА  $\mathcal{A}$ , хотя и имеет с последним такое же множество вершин.) Для  $p, r \in Q$  назовём путь из  $p$  в  $r$  *сингулярным*, если ни  $p$ , ни  $r$  не встречаются на этом пути как промежуточные вершины. Определим  $[p, r]$  как пустое множество, если в  $\Delta$  нет пути из  $p$  в  $r$ , и как множество

$$\{q \in Q : q \text{ встречается на сингулярном пути из } p \text{ в } r\},$$

если в  $\Delta$  есть путь из  $p$  в  $r$ . Граф  $\Delta$  назовём *плотным*, если для всех  $p, q, r \in Q$  из того, что  $[p, r], [r, p], [p, q], [q, r] \neq \emptyset$  следует, что либо  $q \in [p, r]$ , либо  $q \in [r, p]$ . Скажем, что  $\mathcal{A}$  *сохраняет интервалы графа*  $\Delta$ , если для любых  $p, r \in Q$  и любой буквы  $a \in \Sigma$  выполнены следующие условия:

- если  $[p, r] \neq \emptyset$ , то  $[p \cdot a, r \cdot a] \neq \emptyset$ ,
- если  $[p, r], [r, p] \neq \emptyset$ , то  $[p, r] \cdot a \subseteq [p \cdot a, r \cdot a]$ ,
- если  $p \cdot a = r \cdot a$ , то хотя бы одно из множеств  $[p, r] \cdot a$  или  $[r, p] \cdot a$  одноэлементно.

К рассматриваемому здесь типу относятся сильно связанные автоматы, сохраняющие интервалы некоторого слабо связанного плотного графа на множестве своих состояний.

- *Верхняя оценка:*  $(n - 1)^2$ , см. [82, следствие 2.1].
- *Метод:* индукция по числу состояний и разбор случаев.
- *Нижняя оценка:*  $(n - 1)^2$ , поскольку автоматы  $\mathcal{C}_n$  из серии Черни попадают в данный класс (с циклом  $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow n - 1 \rightarrow 0$  в роли графа  $\Delta$ ).

#### А5<sup>†</sup>. Автоматы с 2-пересечениями

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *автоматом с 2-пересечениями*, если для любой буквы  $a \in \Sigma$  выполнено одно из следующих условий:

- все состояния, не фиксируемые буквой  $a$ , за исключением самое большее двух, фиксируются всеми буквами из  $\Sigma \setminus \{a\}$ , а каждое из исключительных состояний не фиксируется ровно одной буквой из  $\Sigma \setminus \{a\}$ ;
- все состояния, не фиксируемые буквой  $a$ , за исключением одного, фиксируются всеми буквами из  $\Sigma \setminus \{a\}$ , а исключительное состояние не фиксируется ровно двумя буквами из  $\Sigma \setminus \{a\}$ .

- *Верхняя оценка:*  $(n - 1)^2$ , см. [83, теорема 2.1].
- *Метод:* Сведение к ДКА типов А1 и А6 и разбор остающихся случаев.
- *Нижняя оценка:*  $(n - 1)^2$ , поскольку автоматы  $\mathcal{C}_n$  из серии Черни попадают в данный класс.

#### А6. Эйлеровы автоматы

ДКА называется *эйлеровым*, если в его носителе есть цикл, содержащий каждое ребро ровно один раз.

- *Верхняя оценка:*  $n^2 - 3n + 3$ , см. [95, теорема 3].
- *Метод:* 1-расширяемость, см. §3.4, с дополнительным улучшением: для эйлеровых автоматов длина слова, добавляемого при каждом прохождении основного цикла алгоритма 3, не превышает  $n - 1$ . Поэтому алгоритм возвращает синхронизирующее слово длины не больше  $1 + (n - 2)(n - 1) = n^2 - 3n + 3$ .

- *Нижняя оценка:*  $\lfloor \frac{n^2-3}{2} \rfloor$ , см. [173].

- *Комментарии:* Верхняя оценка  $n^2 - 3n + 3$  распространена на класс псевдо-эйлеровых автоматов в [168, теорема 3]. ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *псевдо-эйлеровым*, если можно так приписать буквам из  $\Sigma$  положительные веса, чтобы для каждого состояния  $q \in Q$  как сумма весов меток рёбер с началом  $q$ , так и сумма весов меток рёбер с концом  $q$  равнялись 1. (Для эйлеровых автоматов последнее условие выполнено, если каждой букве приписать вес  $\frac{1}{|\Sigma|}$ .)

Нижняя оценка  $\lfloor \frac{n^2-3}{2} \rfloor$  из [173] реализуется на серии эйлеровых автоматов с четырьмя входными буквами. Автоматы из этой же серии показывают, что верхняя оценка  $n - 1$  на длину слова, добавляемого при прохождении основного цикла алгоритма 3 на эйлеровых автоматах, вообще говоря, не улучшаема. В классе эйлеровых ДКА с двумя входными буквами П. В. Мартюгин (неопубликовано) указал серию синхронизируемых  $n$ -автоматов ( $n$  нечётно) с предположительным порогом синхронизации  $\frac{n^2-5}{2}$ . Это предположение подтверждено полным перебором эйлеровых  $n$ -автоматов с двумя входными буквами при  $n = 5, 7, 9, 11$ , причём при каждом из этих  $n$  автомат из серии Мартюгина оказался единственным с порогом синхронизации  $\frac{n^2-5}{2}$ . В общем случае предположение, однако, не доказано, и наибольшей доказанной нижней границей для максимума порогов синхронизации синхронизируемых эйлеровых  $n$ -автоматов с двумя входными буквами ( $n$  нечётно) пока остаётся оценка  $\frac{n^2-3n+4}{2}$  из [84].

#### A7. Автоматы с буквой малого ранга

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *автоматом с буквой малого ранга*, если есть такая буква  $a \in \Sigma$ , что  $|Q \cdot a| \leq \sqrt[3]{6|Q|} - 6$ .

- *Верхняя оценка:*  $(n - 1)^2$ , см. [28, следствие 13].

- *Метод:* следствие теоремы 6 из [28], полученной с помощью усовершенствованного метода расширения.

- *Нижняя оценка:* неизвестна.

#### A8. Автоматы без инволюций

- *Определение:* ДКА  $\mathcal{A}$  называется *автоматом без инволюций*, если в  $\mathcal{A}$  нет состояния  $q$  и слова  $w$  таких, что  $q \cdot w \neq q = q \cdot w^2$ . В алгебраических терминах это означает, что моноид переходов автомата  $\mathcal{A}$  не содержит подгрупп чётного порядка.

- *Верхняя оценка:*  $(n - 1)^2$ , см. [181, теорема 6]

- *Метод:* модификация подхода из [175], см. пункт B2.

- *Нижняя оценка:* Известна только линейная нижняя оценка  $n + \lfloor \frac{n}{2} \rfloor - 2$ , которая следует из [14]. В случае сильно связанных синхронизируемых автоматов без инволюций пока не известны примеры с порогом синхронизации больше  $n - 1$ .

- *Комментарии:* В доказательстве из [181] есть не вполне ясные места.

### B. Типы ДКА с порогом синхронизации от $n$ до $\frac{n(n-1)}{2}$

#### B1<sup>†</sup>. Автоматы с нулём

- *Определение:* ДКА называется *автоматом с нулём*, если в нём есть состояние, фиксируемое всеми буквами.

- *Верхняя оценка:*  $\frac{n(n-1)}{2}$ , [149, теорема 1], см. также [143, теорема 6.1].
- *Метод:* см. доказательство предложения §3.3.
- *Нижняя оценка:*  $\frac{n(n-1)}{2}$  для ДКА с неограниченным алфавитом, см. доказательство предложения §3.3. Для автоматов с двумя входными буквами наилучшая на сегодня нижняя граница для максимума порогов синхронизации синхронизируемых  $n$ -автоматов с нулём составляет  $\frac{1}{4}n^2 + 2n - 9$ ; она достигается при  $n \equiv 4 \pmod{12}$ , начиная с  $n = 16$ , на серии ДКА из [13].

## В2. Аперриодические автоматы

• *Определение:* ДКА  $\mathcal{A}$  называется *аперриодическим*, если в  $\mathcal{A}$  нет состояния  $q$  и слова  $w$  таких, что  $q \cdot w \neq q = q \cdot w^k$  для некоторого  $k$ . В алгебраических терминах это означает, что моноид переходов автомата  $\mathcal{A}$  не содержит неоднородных подгрупп.

- *Верхняя оценка:*  $\frac{n(n-1)}{2}$ , см. [175, теорема 10].
- *Метод:* Класс аперриодических автоматов замкнут относительно взятия подавтоматов и фактор-автоматов, а функция  $\frac{n(n-1)}{2}$  удовлетворяет неравенству (10). По лемме 3.3 верхнюю оценку  $\frac{n(n-1)}{2}$  достаточно проверить для сильно связных синхронизируемых аперриодических ДКА и для синхронизируемых аперриодических ДКА с нулём. Во втором случае она верна для всех ДКА с нулём, см. В1. Сильно связный аперриодический ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  всегда синхронизируем [175, теорема 9] и допускает нетривиальное отношение частичного порядка  $\leq$ , устойчивое относительно действия букв: если  $p \leq r$  для каких-то  $p, r \in Q$ , то  $p \cdot a \leq r \cdot a$  для любой буквы  $a \in \Sigma$ . Поэтому если  $p \cdot w = r \cdot w$  для некоторого слова  $w \in \Sigma^*$ , то это слово переводит в состояние  $p \cdot w = r \cdot w$  любое состояние  $q \in Q$ , такое, что  $p \leq q \leq r$ . Это позволяет собрать синхронизирующее слово длины  $\leq \frac{n(n-1)}{2}$  из слов длины  $\leq n - 1$ , переводящих максимальные относительно порядка  $\leq$  состояния в минимальные или наоборот. (В силу сильной связности автомата  $\mathcal{A}$  для любых двух его состояний есть слово длины  $\leq n - 1$ , переводящее одно в другое.) Ясно, что либо число максимальных, либо число максимальных состояний не превосходит  $\frac{n}{2}$ . Для аккуратной реализации описанного подхода нужно ещё применить индукцию по числу состояний.

• *Нижняя оценка:* Известна только линейная нижняя оценка  $n + \lfloor \frac{n}{2} \rfloor - 2$ , которая следует из [14]. В случае сильно связных аперриодических автоматов не известны примеры с порогом синхронизации больше  $n - 1$ , и предполагается, что порог синхронизации сильно связных аперриодических  $n$ -автоматов не превосходит  $n - 1$ . Это предположение подтверждено полным перебором сильно связных аперриодических  $n$ -автоматов с двумя входными буквами при  $n \leq 11$  и с тремя входными буквами при  $n \leq 7$ , см. [171, §7.4.4].

• *Комментарии:* В [183] верхняя оценка на порог синхронизации сильно связных аперриодических  $n$ -автоматов улучшена до  $\lfloor \frac{n(n+1)}{6} \rfloor$ .

## В3<sup>†</sup>. Автоматы с моноидом переходов из **EDS**

• *Определение:* Класс **DS** состоит из всех конечных моноидов  $M$  таких, что для всех  $x, y, z \in M$  выполнена импликация:<sup>18</sup>

$$\text{если } MxM = MyM = MzM = Mx^2M, \text{ то } MxM = MyzM.$$

Элемент  $e$  моноида  $M$  называется *идемпотентом*, если  $e^2 = e$ . Класс **EDS** состоит из всех конечных моноидов  $M$  со следующим свойством: подмоноид, порождённый всеми

<sup>18</sup>Для читателей, знакомых с основами теории полугрупп, напомним эквивалентное, но более стандартное определение: конечный моноид  $M$  принадлежит классу **DS** тогда и только тогда, когда каждый регулярный  $\mathcal{D}$ -класс моноида  $M$  является подполугруппой в  $M$ .

идемпотентами из  $M$ , лежит в **EDS**. Здесь рассматриваются ДКА, моноиды переходов которых принадлежат **EDS**.

- *Верхняя оценка:*  $\frac{n(n-1)}{2}$ , см. [5, следствие 4.3].

- *Метод:* используется информация о линейных представлениях моноидов из **EDS**, вытекающая из классической теории полугрупповых представлений (теория Манна–Понизовского).

- *Нижняя оценка:*  $\frac{n(n-1)}{2}$  для ДКА с неограниченным алфавитом, поскольку автоматы  $\mathcal{R}_n$  из серии Рысцова (см. доказательство предложения §3.3) попадают в данный класс. Для синхронизируемых ДКА над фиксированным алфавитом из этого класса, а также для сильно связных синхронизируемых ДКА из него о нижних границах значений для максимума порогов синхронизации практически ничего не известно.

#### V4. Декодеры конечных максимальных префиксных кодов

- *Определение:* Множество  $X \subset \Sigma^+$  называется *префиксным кодом над  $\Sigma$* , если слова из  $X$  не являются префиксами друг друга. Префиксный код называется *максимальным*, если он не содержится ни в каком другом префиксном коде над  $\Sigma$ . *Декодер конечного максимального префиксного кода  $X \subset \Sigma^+$*  – это ДКА  $\mathcal{A}_X = \langle Q, \Sigma \rangle$ , где  $Q$  – множество всех префиксов слов из  $X$ , а функция перехода определена так:

$$q \cdot a = \begin{cases} qa, & \text{если } qa \text{ – префикс слова из } X, \\ \varepsilon, & \text{если } qa \in X. \end{cases}$$

- *Верхняя оценка:*  $2 + (n + h - 1) \frac{h^3 - h}{6}$ , где  $h = \lceil \log_{|\Sigma|} n \rceil$ , см. [28, следствие 17].

- *Метод:* усовершенствованный метод расширения.

- *Нижняя оценка:*  $2n - 5$  при чётном  $n$  и  $2n - 7$  при нечётном  $n$ , если  $|\Sigma| = 2$ , см. [33].

Для  $|\Sigma| \geq 3$  в [28, теорема 19] получена нижняя оценка  $2 \lceil \frac{n}{|\Sigma|+1} \rceil$ .

#### V5. Полумонотонные автоматы

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *полумонотонным*, если на  $Q$  есть линейный порядок  $\leq$  и для любой буквы  $a \in \Sigma$  либо для всех  $p, r \in Q$  из  $p \leq r$  следует  $p \cdot a \leq r \cdot a$ , либо для всех  $p, r \in Q$  из  $p \leq r$  следует  $p \cdot a \geq r \cdot a$ . (Другими словами, функция  $q \mapsto q \cdot a$  либо монотонно неубывающая, либо монотонно невозрастающая.)

- *Верхняя оценка:*  $\frac{n(n-1)}{2}$  в общем случае;  $2n - 3$ , если у  $\mathcal{A}$  имеется такая буква  $c$ , что функция  $q \mapsto q \cdot c$  есть монотонно убывающая биекция, см. [9, предложение 3.2].

- *Метод:* Общий случай основан на трёх соображениях. 1) В полумонотонном автомате для любого слова  $w \in \Sigma^*$  и любого интервала  $I$  линейно упорядоченного множества  $\langle Q, \leq \rangle$  полный прообраз  $Iw^{-1}$  является интервалом;<sup>19</sup> 2) Если  $w := a_1 \dots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , – синхронизирующее слово, возвращаемое алгоритмом 3, а  $q \in Q$  – состояние, с которого алгоритм 3 начинает работу, то все подмножества  $q(a_i a_{i+1} \dots a_\ell)^{-1}$  различны и являются неоднородными интервалами; 3) в  $n$ -элементном линейно упорядоченном множестве имеется  $\frac{n(n-1)}{2}$  неоднородных интервалов. Отсюда  $\ell \leq \frac{n(n-1)}{2}$ .

При наличии буквы, действующей как монотонно убывающая биекция, используется сведение к монотонным ДКА, см. пункт C1.

- *Нижняя оценка:*  $2n - 3$  при  $n \equiv 3 \pmod{4}$ , см. [9, предложение 3.1].

#### V6<sup>†</sup>. 0-монотонные автоматы

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  с нулём 0 называется *0-монотонным*, если на  $Q \setminus \{0\}$  есть линейный порядок  $\leq$  и для любой буквы  $a \in \Sigma$  и любых  $p, r \in Q \setminus \{0\}$  из  $p \leq r$  и  $p \cdot a, r \cdot a \neq 0$  следует  $p \cdot a \leq r \cdot a$ .

<sup>19</sup>Непустое подмножество  $I$  линейно упорядоченного множества  $\langle Q, \leq \rangle$  называется *интервалом*, если вместе с любыми  $p, r \in I$ , такими, что  $p \leq r$ , в  $I$  лежат все такие  $q \in Q$ , что  $p \leq q \leq r$ .

- *Верхняя оценка:*  $n + \lfloor \frac{n}{2} \rfloor - 2$ , см. [14, теорема 1].
- *Метод:* сведение к некоторым свойствам монотонных ДКА, см. пункт С1.
- *Нижняя оценка:*  $n + \lfloor \frac{n}{2} \rfloor - 2$ , см. [14, теорема 2].

### С. Типы ДКА с порогом синхронизации $n - 1$

#### С1<sup>†</sup>. Монотонные автоматы

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *монотонным*, если на  $Q$  есть линейный порядок  $\leq$  и для любой буквы  $a \in \Sigma$  и для всех  $p, r \in Q$  из  $p \leq r$  следует  $p \cdot a \leq r \cdot a$ .
- *Верхняя оценка:*  $n - 1$ , см. [10].
- *Метод:* индукция по числу состояний.
- *Нижняя оценка:*  $n - 1$ , примером может служить серия монотонных автоматов  $\mathcal{M}_n := \langle \{0, 1, 2, \dots, n - 1\}, \{a\} \rangle$ , в которых действие буквы определено правилом:

$$i \cdot a := \begin{cases} i - 1, & \text{если } i > 0, \\ 0, & \text{если } i = 0. \end{cases}$$

- *Комментарии:* Насколько можно судить по английской аннотации опубликованной на китайском языке статьи Жен-Хе Куи, Йонг Хе и Ши-Юань Сана [56], в ней верхняя оценка  $n - 1$  распространена на такие ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , что на  $Q$  есть частичный порядок  $\leq$ , относительно которого в  $Q$  есть наибольший и наименьший элементы, и для любой буквы  $a \in \Sigma$  и для всех  $p, r \in Q$  из  $p \leq r$  следует  $p \cdot a \leq r \cdot a$ .

#### С2<sup>†</sup>. Обобщённо монотонные автоматы

- *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется  $\rho$ -*монотонным*, где  $\rho$  – конгруэнция на  $\mathcal{A}$ , если на  $Q$  есть частичный порядок  $\leq$ , такой, что два состояния сравнимы относительно  $\leq$  тогда и только тогда, когда они лежат в одном  $\rho$ -классе, и для любой буквы  $a \in \Sigma$  и для всех  $p, r \in Q$  из  $p \leq r$  следует  $p \cdot a \leq r \cdot a$ . ДКА  $\mathcal{A}$  называется *обобщённо монотонным*, если он допускает цепочку конгруэнций

$$\rho_0 \subset \rho_1 \subset \dots \subset \rho_\ell,$$

такую, что  $\rho_0$  – отношение равенства,  $\rho_\ell$  – универсальное отношение на  $Q$  и для каждого  $i = 1, \dots, \ell$  фактор-автомат  $\mathcal{A}/\rho_{i-1}$  является  $\rho_i/\rho_{i-1}$ -монотонным.

- *Верхняя оценка:*  $n - 1$ , см. [11, теорема 1.2].
- *Метод:* индукция по числу состояний.
- *Нижняя оценка:*  $n - 1$ , поскольку автоматы  $\mathcal{M}_n$  из С1 попадают в данный класс.

#### С3<sup>†</sup>. Автоматы с моноидом переходов из **DS**

- *Определение:* Класс моноидов **DS** определён в пункте В3 выше. Здесь рассматриваются ДКА, моноиды переходов которых принадлежат **DS**.
- *Верхняя оценка:*  $n - 1$ , см. [5, теорема 2.6].
- *Метод:* используется информация о линейных представлениях моноидов из **DS** из статьи [3].
- *Нижняя оценка:*  $n - 1$ , поскольку автоматы  $\mathcal{M}_n$  из С1 попадают в данный класс.
- *Комментарии:* Данный класс автоматов содержит ряд классов, для которых верхняя оценка  $n - 1$  была ранее установлена И. К. Рысцовым, см. [143, 147, 152], а также класс так называемых слабо ациклических автоматов, для которого ту же оценку доказал А. Рыжиков [153]. В терминах моноидов переходов в [143, 147, 152] изучались автоматы с коммутативными или близкими к коммутативным моноидами переходов,

а в [153] – автоматы, моноиды переходов которых  $\mathcal{R}$ -тривиальны.<sup>20</sup> Нетрудно видеть, что и коммутативные, и  $\mathcal{R}$ -тривиальные конечные моноиды принадлежат **DS**.

**C4<sup>†</sup>**. Автоматы с двумя идемпотентными буквами.

• *Определение*: ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *автоматом с идемпотентными буквами*, если  $q \cdot a = q \cdot a^2$  для любых  $q \in Q$  и  $a \in \Sigma$ . Здесь рассматриваются такие автоматы при  $|\Sigma| = 2$ .

• *Верхняя оценка*:  $n - 1$ , см. [184, предложение 5].

• *Метод*: Класс автоматов с двумя идемпотентными буквами замкнут относительно взятия подавтоматов и фактор-автоматов, а функция  $n - 1$  удовлетворяет неравенству (10). По лемме 3.3 верхнюю оценку  $n - 1$  достаточно проверить для сильно связного случая и случая ДКА с нулём. Показывается, что сильно связный синхронизируемый автомат с двумя идемпотентными буквами имеет не более двух состояний. Для синхронизируемых автоматов с двумя идемпотентными буквами и с нулём используется индукция по числу состояний.

• *Нижняя оценка*:  $n - 1$ , примером служит серия ДКА  $\mathcal{F}_n := \langle \{1, 2, \dots, n\}, \{a, b\} \rangle$ , в которых действие букв определено правилом:

$$i \cdot a = \begin{cases} i, & \text{если } i \text{ нечётно или } i = n, \\ i+1, & \text{если } i \text{ чётно и } i < n; \end{cases} \quad i \cdot b = \begin{cases} i, & \text{если } i \text{ чётно или } i = n, \\ i+1, & \text{если } i \text{ нечётно и } i < n. \end{cases}$$

• *Комментарии*: При  $|\Sigma| > 2$  порог синхронизации  $n$ -автомата с идемпотентными буквами может достигать  $\lfloor \frac{n^2}{2} \rfloor - 2n + 2$ , см. [184, следствие 3] или [60, §3].

**C5**. Медиумы.

• *Определение*: Для ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  буквы  $a, b \in \Sigma$  *квазиобратны* друг другу, если  $p \cdot a = r$  тогда и только тогда, когда  $r \cdot b = p$  для любых различных состояний  $p, r \in Q$ . Слово из  $\Sigma^*$  называется *согласованным*, если среди его букв нет квазиобратных друг другу. Слово  $w \in \Sigma^*$  называется *пассивным*, если каждая буква встречается в  $w$  столько же раз, сколько и буква, квазиобратная к ней. Слово  $a_1 \cdots a_\ell$ , где  $a_1, \dots, a_\ell \in \Sigma$ , называется *пошагово эффективным для состояния*  $q \in Q$ , если  $q \cdot a_1 \neq q$  и  $q \cdot a_1 \cdots a_i \neq q \cdot a_1 \cdots a_{i-1}$  для всех  $i = 2, \dots, \ell$ . *Медиумом* называется ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$ , удовлетворяющий следующим аксиомам:

- каждая буквы из  $\Sigma$  имеет единственную квазиобратную,
- для любых различных состояний  $p, r \in Q$  существует согласованное слово  $w \in \Sigma^*$ , такое, что  $p \cdot w = r$ ,
- если слово  $w \in \Sigma^*$  пошагово эффективно для состояния  $q \in Q$ , то  $q \cdot w = q$  тогда и только тогда, когда  $w$  пассивно,
- если  $p \cdot u = r \cdot v$ , слово  $u$  пошагово эффективно для  $p$ , слово  $v$  пошагово эффективно для  $r$  и оба слова  $u, v$  согласованы, то и слово  $uv$  согласовано.

• *Верхняя оценка*:  $n - 1$ , см. [64, теорема 1].

• *Метод*: задача сводится к поиску кратчайшего пути во вспомогательном графе без циклов с множеством вершин  $Q$ .

• *Нижняя оценка*: В [64] утверждается, что верхняя оценка  $n - 1$  точна, однако пример медиума с порогом синхронизации  $n - 1$  приведён только для  $n = 6$ .

<sup>20</sup>Моноид  $M$  называется  $\mathcal{R}$ -тривиальным, если для всех  $x, y \in M$  из  $xM = yM$  следует, что  $x = y$ .

## Д. Типы ДКА с квадратичной оценкой на порог синхронизации

### Д1. Однокластерные автоматы.

- *Определение:* см. пункт А2.
- *Верхняя оценка:*  $2n^2 - 4n + 1 - 2(n - 1) \ln \frac{n}{2}$ , см. [42, следствие 1].
- *Метод:* 2-расширяемость, см. §3.4, с некоторыми дополнительными техническими ухищрениями.

• *Нижняя оценка:*  $(n - 1)^2$ , поскольку автоматы  $\mathcal{C}_n$  из серии Черни однокластерные.

• *Комментарии:* М. В. Берлинков [23] называет *квази-однокластерным степени  $d$*  ДКА, у которого есть такая буква, что сумма длин всех, кроме одного, простых циклов, помеченных этой буквой, не превышает  $d$ . (Однокластерные автоматы суть в точности квази-однокластерные автоматы степени 0; таким образом, параметр  $d$  измеряет «отклонение» от однокластерности.) В [23, следствие 11] показано, что порог синхронизации квази-однокластерного  $n$ -автомата степени  $d$  не превосходит  $2^d(n - d + 1)(2n - d - 2)$ .

### Д2. Автоматы с полным моноидом переходов.

• *Определение:* ДКА называется *автоматом с полным моноидом переходов*, если его моноид переходов состоит из всех преобразований множества состояний.

• *Верхняя оценка:*  $2n^2 - 6n + 5$ , см. [79, теорема 7].

• *Метод:* 2-расширяемость, см. §3.4.

• *Нижняя оценка:* В [79, теорема 4] приведена серия  $n$ -автоматов с  $n + 1$  входной буквой, с полным моноидом переходов и порогом синхронизации  $\frac{n(n-1)}{2}$ . В [60, §5.1] приведена сходная серия  $n$ -автоматов с  $n$  входными буквами и порогом синхронизации  $\frac{n(n+1)}{2}$ ; можно проверить, что и автоматы этой серии имеют полный моноид переходов. Для ДКА с полным моноидом переходов над фиксированным алфавитом нетривиальные нижние оценки неизвестны.

### Д3. Автоматы с простыми идемпотентами.

• *Определение:* ДКА называется *автоматом с простыми идемпотентами*, если каждая его буква либо действует как перестановка множества состояний, либо фиксирует все состояния, кроме одного.

• *Верхняя оценка:*  $2(n - 1)^2$ , см. [148, теорема 4].

• *Метод:* 2-расширяемость, см. §3.4.

• *Нижняя оценка:*  $(n - 1)^2$ , ибо автоматы  $\mathcal{C}_n$  из серии Черни лежат в этом классе.

### Д4. Квази-эйлеровы автоматы.

• *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *квази-эйлеровым степени  $d$* , если выполнены следующие условия:

- существует такое подмножество  $E_d \subset Q$  из  $|Q| - d$  состояний, что ровно одно состояние  $s \in E_d$  может служить концом рёбер, начинающихся в  $Q \setminus E_d$ ;
- можно так приписать буквам из  $\Sigma$  положительные веса, чтобы для каждого состояния  $p \in Q$  сумма весов меток рёбер с началом  $q$  равнялась 1 и для каждого состояния  $r \in E_d \setminus \{s\}$  сумма весов меток рёбер с концом  $r$  равнялась 1.

• *Верхняя оценка:*  $2^d(n - d + 1)(n - 1)$ , см. [23, теорема 8].

• *Метод:* усовершенствованный метод расширения.

• *Нижняя оценка:*  $(n - 1)^2$ , так как автоматы  $\mathcal{C}_n$  из серии Черни являются квази-эйлеровыми степени 1. (В качестве множества  $E_1$  годится множество  $\{1, 2, \dots, n - 1\}$ , а в качестве специального состояния  $s$  – состояние 1, см. рис. 15. В качестве весов букв  $a$  и  $b$  годятся любые два положительных числа, дающих в сумме 1.)

## D5. Регулярные автоматы.

• *Определение:* ДКА  $\mathcal{A} = \langle Q, \Sigma \rangle$  называется *регулярным*, если существует семейство слов  $W \subset \Sigma^*$ , содержащее пустое слово и удовлетворяющее следующим условиям:

- длина любого слова из  $W$  меньше числа состояний автомата;
- существует такое число  $m \geq 1$ , что для любых двух состояний  $p, r \in Q$  в  $W$  есть ровно  $m$  слов, переводящих  $p$  в  $r$ .

• *Верхняя оценка:*  $2(n-1)^2$ , см. [146, теорема 7] или [145, теорема 7].

• *Метод:* 2-расширяемость, см. §3.4.

• *Нижняя оценка:*  $(n-1)^2$ , так как автоматы  $\mathcal{C}_n$  из серии Черни регулярны. (В роли семейства  $W$  при  $m = 1$  годится множество слов  $\{\varepsilon, b, b^2, \dots, b^{n-1}\}$ .)

• *Комментарии:* Сходные определения вводили Карпи и Д'Алессандро [40, 42]. В [40] они назвали  $n$ -автомат  $\mathcal{A} = \langle Q, \Sigma \rangle$  *сильно транзитивным*, если для него существует семейство из  $n$  слов  $\{w_1, \dots, w_n\} \subset \Sigma^*$ , такое, что для любых  $p, r \in Q$  есть ровно одно слово  $w_i$ , для которого  $p \cdot w_i = r$ . Если  $L := \max\{|w_1|, \dots, |w_n|\}$ , то (см. [40, теорема 2])

$$\text{rt}(\mathcal{A}) \leq (n-2)(n+L-1) + 1.$$

В [42]  $n$ -автомат  $\mathcal{A} = \langle Q, \Sigma \rangle$  назван *локально сильно транзитивным*, если для него существует семейство из  $k$  слов  $\{w_1, \dots, w_k\} \subset \Sigma^*$  и множество  $\{q_1, \dots, q_k\} \subset Q$  из  $k$  различных состояний, такие, что  $\{q \cdot w_1, \dots, q \cdot w_k\} = \{q_1, \dots, q_k\}$  для любого  $q \in Q$ . Если  $L := \max\{|w_1|, \dots, |w_k|\}$ , а  $\ell := \min\{|w_1|, \dots, |w_k|\}$ , то (см. [42, предложение 5])

$$\text{rt}(\mathcal{A}) \leq (k-1)(n+L+1) - 2k \ln \frac{k+1}{2} + \ell.$$

## 3.6 Экспериментальные результаты

Значение экспериментальных исследований в изучении автоматов заметно возросло за последние 20–25 лет с распространением всё более мощных и легко доступных компьютеров. Теория автоматов стала во многом следовать методике естественных наук, где первоначальным источником знания служат эксперименты, осмысление результатов которых позволяет строить новые гипотезы и нащупывать пути для их доказательства (или опровержения). Этот процесс затронул и круг вопросов, связанных с гипотезой Черни. Обширные компьютерные эксперименты [12, 57, 59, 98, 99, 171, 174] подтвердили справедливость гипотезы для всех ДКА с 7 или менее состояниями и любым размером входного алфавита, для всех ДКА с тремя входными буквами и 8 состояниями и для всех ДКА с двумя входными буквами и 12 или менее состояниями. Чтобы дать представление о масштабах необходимых вычислений, напомним, что число  $n$ -автоматов с  $m$  входными буквами равно  $n^{nm}$ , поскольку задание такого автомата равносильно выбору  $m$  отображений среди всех  $n^n$  отображений  $n$ -элементного множества в себя. Функция  $n^{nm}$  очень быстро растёт с ростом  $n$  и  $m$ ; например, число ДКА с тремя входными буквами и 8 состояниями равно  $8^{24} = 2^{72} \approx 4,7223665 \cdot 10^{21}$ . Для сокращения перебора используется ряд нетривиальных приёмов; обзор используемой здесь техники см. в диссертации Шикеры [171, глава 6]. Так, в случае 8-автоматов с тремя входными буквами число автоматов, которые реально пришлось проанализировать, удалось сократить до 20 933 723 139.

Помимо подтверждения гипотезы Черни для ДКА небольших размеров, переборные эксперименты выявили непредвиденные особенности в распределении возможных

порогов синхронизации синхронизируемых автоматов с данным числом состояний. Например, среди синхронизируемых ДКА с 6 состояниями есть автоматы<sup>21</sup> с порогом синхронизации 25, но нет ни одного с порогом синхронизации 24. Зазор между максимальным и вторым по величине порогами синхронизации синхронизируемых автоматов с  $n \geq 6$  впервые обнаружил А. Н. Трахтман [174]. Эксперименты в [174] ограничивались  $n \leq 10$ , но дальнейшие эксперименты подтвердили существование такого зазора также при  $n = 11$  и  $n = 12$ . В [12, гипотеза 1(d)] высказано предположение, что при  $n > 6$  второй по величине возможный порог синхронизации синхронизируемых автоматов с  $n$  состояниями равен  $n^2 - 3n + 4$  и что при  $n > 7$  существует единственный (с точностью до изоморфизма и опускания несущественных букв)  $n$ -автомат, на котором эта величина достигается. А именно, речь идет о серии автоматов  $\mathcal{D}_n := \langle \{0, 1, 2, \dots, n-1\}, \{a, b\} \rangle$ , в которых действие букв определено правилами:

$$i.a := \begin{cases} i+1, & \text{если } i < n-2, \\ 0, & \text{если } i = n-2, \\ 1, & \text{если } i = n-1; \end{cases} \quad i.b := i+1 \pmod{n}.$$

Типичный автомат из серии  $\mathcal{D}_n$  изображён на рис. 23.

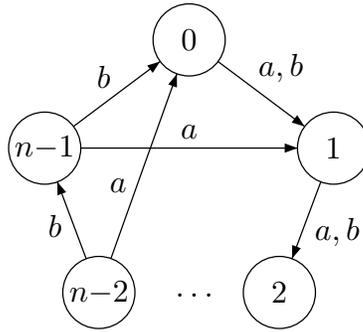


Рис. 23: Автомат  $\mathcal{D}_n$

В переборных экспериментах, описанных в [12], было обнаружено, что при  $n > 8$  имеется второй зазор в распределении возможных порогов синхронизации синхронизируемых автоматов с  $n$  состояниями и двумя входными буквами. А именно, среди таких автоматов не появляются ДКА с порогом синхронизации, меньшим чем  $n^2 - 3n + 2$  и бóльшим чем  $n^2 - 4n + 7$ , если  $n$  нечётно или  $n^2 - 4n + 6$ , если  $n$  чётно. Третий зазор похожего сорта был зарегистрирован при  $n > 10$  в [99, 171]. Детальный анализ зазоров в распределении порогов синхронизации предпринят в [62].

Феномен «разрывности» порогов синхронизации любопытен сам по себе, но мы остановились на нём главным образом потому, что его обнаружение помогло нащупать интересные связи, до того остававшиеся незамеченными. (Здесь как раз сработала «методика естественных наук», упомянутая в начале раздела.) А именно, в [12] было подмечено, что наблюдаемое в экспериментах поведение количества синхронизируемых автоматов с фиксированным числом состояний как функции от порога синхронизации весьма сходно с поведением другой величины, обстоятельно исследовавшейся в дискретной математике, – количества примитивных графов с фиксированным чис-

<sup>21</sup>С точностью до изоморфизма и опускания несущественных букв имеется два синхронизируемых 6-автомата с порогом синхронизации 25: автомат  $\mathcal{C}_6$  из серии Черни и автомат Кари  $\mathcal{K}_6$ , см. рис. 20. Буква *несущественна*, если её удаление из алфавита автомата не меняет порог синхронизации.

лом вершин как функции от значения экспоненты.<sup>22</sup> Анализ причин такого сходства позволил установить ряд новых обоюдодоползных соответствий между синхронизируемыми ДКА и примитивными неотрицательными матрицами; эти взаимосвязи подробно обсуждаются во второй части статьи.

Возвращаясь к обсуждению экспериментальных результатов, связанных с гипотезой Черни, упомянем ещё обширные эксперименты, проводившиеся над случайными автоматами, см. [53, 97, 98, 162]. Во всех этих экспериментах использовалась простейшая модель случайного ДКА с  $n$  состояниями и  $m$  буквами, в которой такой ДКА есть просто  $m$ -ка отображений, выбранных равномерно и случайно из всех  $n^n$  преобразований множества состояний. В контексте гипотезы Черни эксперименты со случайными ДКА показали, что всякий раз, когда случайно сгенерированный автомат был синхронизируем, его порог синхронизации оказывался намного меньше, чем граница из гипотезы Черни. Например, максимальное значение порога синхронизации, наблюдавшееся в [97] для синхронизируемых 100-автоматов с двумя входными буквами, составило 41 при том, что в ходе эксперимента было сгенерировано около миллиона таких автоматов. Напомним для сравнения, что гипотетическая верхняя граница для порога синхронизации синхронизируемых 100-автоматов составляет  $99^2 = 9801$ , а доказанная верхняя граница –  $(100^3 - 100)/6 = 166\,650$ . Таким образом, эксперименты показали, что даже если гипотеза Черни в общем случае не выполнена, она с большим запасом верна для «почти всех» синхронизируемых автоматов. Это экспериментальное наблюдение нашло теоретическое обоснование в важных работах Сирила Нико [123] и М. В. Берлинкова и Марека Шикеры [28], которые в деталях обсуждаются во второй части статьи.

## Заключение

Первая часть статьи фокусировалась на алгоритмических и теоретико-сложностных аспектах синхронизации конечных детерминированных автоматов и на гипотезе Черни. Другие разделы теории синхронизируемых автоматов освещаются во второй части.

Вторая часть состоит из семи параграфов, нумерация которых продолжается. В §§4 и 5 подробно обсуждаются два крупнейших достижения последних лет в теории синхронизируемых автоматов. Это найденное А. Н. Трахтманом [176] доказательство гипотезы Роя Л. Адлера, Л. Уэйна Гудвина и Бенджамина Вайса [1], известной под названием «Гипотеза о раскраске дорог», и доказательство гипотезы Питера Камерона [38] о вероятности синхронизации случайного детерминированного автомата, полученное М. В. Берлинковым [26]. §6 посвящён глубоким связям между синхронизируемыми ДКА и примитивными неотрицательными матрицами. В трёх следующих параграфах излагаются основные результаты о синхронизации других типов автоматов. В §7 рассматриваются частичные детерминированные автоматы, в §8 – недетерминированные автоматы, а в §9 – различные другие автоматные модели. Заключительный §10 содержит сводку основных открытых на сегодняшний день задач теории синхронизируемых автоматов.

---

<sup>22</sup>Сильно связный граф называется *примитивным*, если наибольший общий делитель длин всех его циклов равен 1. Граф  $\Gamma$  обладает указанным свойством тогда и только тогда, когда его матрица смежности  $M(\Gamma)$  примитивна в смысле теории неотрицательных матриц (теории Перрона–Фробениуса), т.е. когда у  $M(\Gamma)$  есть положительное собственное значение, строго большее модуля любого из её остальных собственных значений. *Экспонентой*  $M(\Gamma)$  называется наименьшее число  $d$ , для которого все элементы матрицы  $M(\Gamma)^d$  положительны. Экспоненты графов служат объектом интенсивного изучения на протяжении последних 70 лет, начиная с классической статьи Виландта [187]; обзор накопленных в этой области фактов см., например, в монографии [37].

## Список литературы

- [1] R. L. Adler, L. W. Goodwyn, and B. Weiss. Equivalence of topological Markov shifts. *Israel J. Math*, 27(1):49–63, 1977.
- [2] P. Ageev. Implementation of the algorithm for testing an automaton for synchronization in linear expected time. *J. Autom. Lang. Comb.*, 24(2-4):139–152, 2019.
- [3] J. Almeida, S. Margolis, B. Steinberg, and M. Volkov. Representation theory of finite semigroups, semigroup radicals and formal language theory. *Trans. Amer. Math. Soc.*, 361:1429–1461, 2009.
- [4] J. Almeida and E. Rodaro. Semisimple synchronizing automata and the Wedderburn–Artin theory. *Int. J. Found. Comput. Sci.*, 27(2):127–146, 2016.
- [5] J. Almeida and B. Steinberg. Matrix mortality and the Černý-Pin conjecture. In V. Diekert and D. Nowotka, editors, *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, volume 5583 of *Lecture Notes in Computer Science*, pages 67–80. Springer, 2009.
- [6] J. Almeida and M. Volkov. Profinite identities for finite semigroups whose subgroups belong to a given pseudovariety. *J. Algebra Appl.*, 2(2):137–163, 2003.
- [7] D. S. Ananichev and V. V. Gusev. Approximation of reset thresholds with greedy algorithms. *Fundam. Inform.*, 145(3):221–227, 2016.
- [8] D. S. Ananichev, I. V. Petrov, and M. V. Volkov. Collapsing words: a progress report. *Int. J. Found. Comput. Sci.*, 17(3):507–518, 2006.
- [9] D. S. Ananichev and M. V. Volkov. Some results on Černý type problems for transformation semigroups. In I. M. Araújo, M. J. J. Branco, V. H. Fernandes, and G. M. S. Gomes, editors, *Semigroups and Languages*, pages 23–42. World Scientific, 2004.
- [10] D. S. Ananichev and M. V. Volkov. Synchronizing monotonic automata. *Theor. Comput. Sci.*, 327(3):225–239, 2004.
- [11] D. S. Ananichev and M. V. Volkov. Synchronizing generalized monotonic automata. *Theor. Comput. Sci.*, 330(1):3–13, 2005.
- [12] D. S. Ananichev, M. V. Volkov, and V. V. Gusev. Primitive digraphs with large exponents and slowly synchronizing automata. *J. Math. Sci.*, 192(3):263–278, 2013. Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 402, 2012, pp. 9–39.
- [13] D. S. Ananichev and V. Vorel. A new lower bound for reset threshold of binary synchronizing automata with sink. *J. Autom. Lang. Comb.*, 24(2-4):153–164, 2019.
- [14] Д. С. Ананичев. Порог аннуляции для частично монотонных автоматов. *Изв. вузов. Матем.*, 2010(1):3–13, 2010. (English transl.: The Annulation Threshold for Partially Monotonic Automata, *Russian Math. (Iz. VUZ)*, 54:1 (2010), 1–9).
- [15] J. Araújo, P. Cameron, and B. Steinberg. Between primitive and 2-transitive: synchronization and its friends. *EMS Surveys in Mathematical Sciences*, 4(2):101–184, 2017.
- [16] F. Arnold and B. Steinberg. Synchronizing groups and automata. *Theor. Comput. Sci.*, 359(1-3):101–110, 2006.
- [17] W. R. Ashby. *An introduction to cybernetics*. Chapman & Hall, 1956.
- [18] M.-P. Béal, M. V. Berlinkov, and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. *Internat. J. Found. Comp. Sci.*, 22(2):277–288, 2011.
- [19] M.-P. Béal and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. In V. Diekert and D. Nowotka, editors, *Developments in Language Theory*, volume 5583 of *Lecture Notes in Comput. Sci.*, pages 81–90. Springer, 2009.

- [20] Y. Benenson, R. Adar, T. Paz-Elizur, Z. Livneh, and E. Shapiro. DNA molecule provides a computing machine with both data and fuel. *Proc. National Acad. Sci. USA*, 100:2191–2196, 2003.
- [21] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414(1):430–434, 2001.
- [22] M. V. Berlinkov. On a conjecture by Carpi and D’Alessandro. *Int. J. Found. Comput. Sci.*, 22(7):1565–1576, 2011.
- [23] M. V. Berlinkov. Synchronizing quasi-Eulerian and quasi-one-cluster automata. *Int. J. Found. Comput. Sci.*, 24(6):729–746, 2013.
- [24] M. V. Berlinkov. Approximating the minimum length of synchronizing words is hard. *Theory Comput. Syst.*, 54(2):211–223, 2014.
- [25] M. V. Berlinkov. On two algorithmic problems about synchronizing automata. In A. M. Shur and M. V. Volkov, editors, *Developments in Language Theory – 18th International Conference, DLT 2014, Ekaterinburg, Russia, August 26–29, 2014. Proceedings*, volume 8633 of *Lecture Notes in Comput. Sci.*, pages 61–67. Springer, 2014.
- [26] M. V. Berlinkov. On the probability of being synchronizable. In S. Govindarajan and A. Maheshwari, editors, *Algorithms and Discrete Applied Mathematics – Second International Conference, CALDAM 2016, Thiruvananthapuram, India, February 18–20, 2016, Proceedings*, volume 9602 of *Lecture Notes in Comput. Sci.*, pages 73–84. Springer, 2016.
- [27] M. V. Berlinkov, R. Ferens, and M. Szykula. Complexity of preimage problems for deterministic finite automata. In I. Potapov, P. G. Spirakis, and J. Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27–31, 2018, Liverpool, UK*, volume 117 of *LIPICs*, pages 32:1–32:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [28] M. V. Berlinkov and M. Szykuła. Algebraic synchronization criterion and computing reset words. *Inf. Sci.*, 369:718–730, 2016.
- [29] J. Berstel, D. Perrin, and C. Reutenauer. *Codes and automata*. Number 129 in *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2009.
- [30] M. T. Biskup. *Error Resilience in Compressed Data – Selected Topics*. Ph.D. Thesis, Institute of Informatics, University of Warsaw, 2008.
- [31] M. T. Biskup. Guaranteed synchronization of Huffman codes. In *2008 Data Compression Conference (DCC 2008), 25–27 March 2008, Snowbird, UT, USA*, pages 462–471. IEEE Computer Society, 2008.
- [32] M. T. Biskup and W. Plandowski. Guaranteed synchronization of Huffman codes with known position of decoder. In J. A. Storer and M. W. Marcellin, editors, *2009 Data Compression Conference (DCC 2009), 16–18 March 2009, Snowbird, UT, USA*, pages 33–42. IEEE Computer Society, 2009.
- [33] M. T. Biskup and W. Plandowski. Shortest synchronizing strings for Huffman codes. *Theor. Comput. Sci.*, 410(38–40):3925–3941, 2009.
- [34] S. Bogdanović, B. Imreh, M. Ćirić, and T. Petković. Directable automata and their generalizations: a survey. *Novi Sad J. Math.*, 29(2):29–69, 1999.
- [35] P. Bonizzoni and N. Jonoska. Existence of constants in regular splicing languages. *Inf. Comput.*, 242:340–353, 2015.
- [36] V. Boppana, S. Rajan, K. Takayama, and M. Fujita. Model checking based on sequential ATPG. In *Computer Aided Verification*, volume 1622 of *Lecture Notes in Comput. Sci.*, pages 418–430. Springer, 1999.
- [37] R. Brualdi and H. Ryser. *Combinatorial Matrix Theory*. Cambridge University Press, 1991.
- [38] P. J. Cameron. Dixon’s theorem and random synchronization. *Discrete Math.*, 313(11):1233–1236, 2013.

- [39] R. M. Capocelli, L. Gargano, and U. Vaccaro. On the characterization of statistically synchronizable variable-length codes. *IEEE Trans. Inform. Theory*, 34(4):817–825, 1988.
- [40] A. Carpi and F. D’Alessandro. Strongly transitive automata and the Černý conjecture. *Acta Informatica*, 46(8):591–607, 2009.
- [41] A. Carpi and F. D’Alessandro. The synchronization problem for locally strongly transitive automata. In R. Kráľovic and D. Niwinski, editors, *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings*, volume 5734 of *Lecture Notes in Computer Science*, pages 211–222. Springer, 2009.
- [42] A. Carpi and F. D’Alessandro. Independent sets of words and the synchronization problem. *Adv. Appl. Math.*, 50(3):339–355, 2013.
- [43] A. Carpi and F. D’Alessandro. Locally strongly transitive automata in the Černý conjecture and related problems. *J. Autom. Lang. Comb.*, 24(2-4):165–184, 2019.
- [44] J. Černý. Poznámka k homogénnym experimentom s konečnými automatami. *Matematicko-fyzikálny Časopis Slovenskej Akadémie Vied*, 14(3):208–216, 1964. (in Slovak).
- [45] J. Černý. A note on homogeneous experiments with finite automata. *J. Autom. Lang. Comb.*, 24(2-4):123–132, 2019. (English translation of [44]).
- [46] J. Černý, A. Pirická, and B. Rosenauerová. On directable automata. *Kybernetika*, 7(4):289–298, 1971.
- [47] Y.-B. Chen and D. J. Ierardi. The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14:367–397, 1995.
- [48] A. Cherubini. Synchronizing and collapsing words. *Milan J. Math.*, 75(1):305–321, 2007.
- [49] A. Cherubini, A. Frigeri, and Z. Liu. Composing short 3-compressing words on a 2-letter alphabet. *Discret. Math. Theor. Comput. Sci.*, 19(1), 2017.
- [50] A. Cherubini and A. Kisielewicz. Collapsing words, permutation conditions and coherent colorings of trees. *Theor. Comput. Sci.*, 410(21-23):2135–2147, 2009.
- [51] A. Cherubini and A. Kisielewicz. Recognizing 3-collapsing words over a binary alphabet. *Theor. Comput. Sci.*, 629:64–79, 2016.
- [52] A. Cherubini, A. Kisielewicz, and B. Piochi. On the length of shortest 2-collapsing words. *Discret. Math. Theor. Comput. Sci.*, 11(1):33–44, 2009.
- [53] K. Chmiel and A. Roman. COMPAS—A computing package for synchronization. In M. Domaratzki and K. Salomaa, editors, *Implementation and Application of Automata – 15th International Conference, CIAA 2010, Winnipeg, MB, Canada, August 12–15, 2010. Revised Selected Papers*, volume 6482 of *Lecture Notes in Comput. Sci.*, pages 79–86. Springer, 2010.
- [54] H. Cho, S.-W. Jeong, F. Somenzi, and C. Pixley. Synchronizing sequences and symbolic traversal techniques in test generation. *J. Electronic Testing*, 4:19–31, 1993.
- [55] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press and McGraw-Hill, 2001.
- [56] Z.-H. Cui, Y. He, and S.-Y. Sun. Synchronizing bounded partially ordered automata. *Chinese J. Computers*, 42(3):610–623, 2019. in Chinese.
- [57] M. de Bondt, H. Don, and H. Zantema. DFAs and PFAs with long shortest synchronizing word length. In É. Charlier, J. Leroy, and M. Rigo, editors, *Developments in Language Theory – 21st International Conference, DLT 2017, Liège, Belgium, August 7–11, 2017, Proceedings*, volume 10396 of *Lecture Notes in Comput. Sci.*, pages 122–133. Springer, 2017.
- [58] F. M. Dekking. The spectrum of dynamical systems arising from substitutions of constant length. *Z. Wahrsch. Verw. Gebiete*, 41:221–239, 1978.

- [59] H. Don and H. Zantema. Finding DFAs with maximal shortest synchronizing word length. In F. Drewes, C. Martín-Vide, and B. Truthe, editors, *Language and Automata Theory and Applications – 11th International Conference, LATA 2017, Umeå, Sweden, March 6–9, 2017, Proceedings*, volume 10168 of *Lecture Notes in Computer Science*, pages 249–260, 2017.
- [60] H. Don and H. Zantema. Counting symbol switches in synchronizing automata. *J. Autom. Lang. Comb.*, 24(2-4):253–286, 2019.
- [61] L. Dubuc. Sur le automates circulaires et la conjecture de Černý. *RAIRO Inform. Théor. App.*, 32:21–34, 1998. (in French).
- [62] M. Dzyga, R. Ferens, V. V. Gusev, and M. Szykuła. Attainable values of reset thresholds. In K. G. Larsen, H. L. Bodlaender, and J.-F. Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21–25, 2017 – Aalborg, Denmark*, volume 83 of *LIPICs*, pages 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
- [63] D. Eppstein. Reset sequences for monotonic automata. *SIAM J. Comput.*, 19:500–510, 1990.
- [64] D. Eppstein and J. Falmagne. Algorithms for media. *Discret. Appl. Math.*, 156(8):1308–1320, 2008.
- [65] H. Fernau, V. V. Gusev, S. Hoffmann, M. Holzer, M. V. Volkov, and P. Wolf. Computational complexity of synchronization under regular constraints. In P. Rossmanith, P. Heggernes, and J. Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPICs*, pages 63:1–63:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [66] H. Fernau, P. Heggernes, and Y. Villanger. A multi-parameter analysis of hard problems on deterministic finite automata. *J. Comput. Syst. Sci.*, 81(4):747–765, 2015.
- [67] H. Fernau and A. Krebs. Problems on finite automata and the exponential time hypothesis. *Algorithms*, 10(1):24, 2017.
- [68] M. A. Fischler and M. Tannenbaum. Synchronizing and representation problems for sequential machines with masked outputs. In *Proc. 11th Annual Symp. Switching and Automata Theory*, pages 97–103. IEEE Press, 1970.
- [69] P. Frankl. An extremal problem for two families of sets. *European J. Combinatorics*, 3:125–127, 1982.
- [70] D. Frettlöh and B. Sing. Computing modular coincidences for substitution tilings and point sets. *Discrete Comput. Geom.*, 37:381–407, 2007.
- [71] A. Frigeri and E. Rodaro. Missing factors of ideals and synchronizing automata. *J. Autom. Lang. Comb.*, 24(2-4):309–320, 2019.
- [72] P. Gawrychowski. Complexity of shortest synchronizing word. Private communication, 2008.
- [73] P. Gawrychowski and D. Straszak. Strong inapproximability of the shortest reset word. In G. F. Italiano, G. Pighizzini, and D. Sannella, editors, *Mathematical Foundations of Computer Science 2015 – 40th International Symposium, MFCS 2015, Milan, Italy, August 24–28, 2015, Proceedings, Part I*, volume 9234 of *Lecture Notes in Comput. Sci.*, pages 243–255. Springer, 2015.
- [74] M. Gerbush and B. Heeringa. Approximating minimum reset sequences. In M. Domaratzki and K. Salomaa, editors, *Implementation and Application of Automata*, volume 6482 of *Lecture Notes in Comput. Sci.*, pages 154–162. Springer, 2011.
- [75] A. Gill. State-identification experiments in finite automata. *Inform. Control*, 4(2-3):132–154, 1961.
- [76] S. Ginsburg. On the length of the smallest uniform experiment which distinguishes the terminal states of a machine. *J. Assoc. Comput. Mach.*, 5:266–280, 1958.
- [77] S. Ginsburg. *An Introduction to Mathematical Machine Theory*. Addison-Wesley, 1962.

- [78] K. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.
- [79] F. Gonze, V. V. Gusev, R. M. Jungers, B. Gerencsér, and M. V. Volkov. On the interplay between Černý and Babai’s conjectures. *Int. J. Found. Comput. Sci.*, 30(1):93–114, 2019.
- [80] F. Gonze, R. M. Jungers, and A. Trakhtman. A note on a recent attempt to improve the Pin–Frankl bound. *Discrete Math. & Theoret. Comput. Sci.*, 17(1):307–308, 2015.
- [81] P. Goralčík and V. Koubek. Rank problems for composite transformations. *Internat. J. Algebra Comput.*, 5:309–316, 1995.
- [82] M. Grech and A. Kisielewicz. The černý conjecture for automata respecting intervals of a directed graph. *Discret. Math. Theor. Comput. Sci.*, 15(3):61–72, 2013.
- [83] M. Grech and A. Kisielewicz. Synchronizing sequences for road colored digraphs. *Discret. Appl. Math.*, 285:128–140, 2020.
- [84] V. V. Gusev. Lower bounds for the length of reset words in Eulerian automata. In G. Delzanno and I. Potapov, editors, *Reachability Problems*, volume 6945 of *Lecture Notes in Comput. Sci.*, pages 180–190. Springer, 2011.
- [85] V. V. Gusev, R. M. Jungers, and D. Průša. Dynamics of the independence number and automata synchronization. In M. Hoshi and S. Seki, editors, *Developments in Language Theory - 22nd International Conference, DLT 2018, Tokyo, Japan, September 10-14, 2018, Proceedings*, volume 11088 of *Lecture Notes in Computer Science*, pages 379–391. Springer, 2018.
- [86] V. V. Gusev, M. I. Maslennikova, and E. V. Pribavkina. Principal ideal languages and synchronizing automata. *Fundam. Inform.*, 132(1):95–108, 2014.
- [87] V. V. Gusev and E. V. Pribavkina. On codeword lengths guaranteeing synchronization. In R. Mercas and D. Reidenbach, editors, *Combinatorics on Words - 12th International Conference, WORDS 2019, Loughborough, UK, September 9-13, 2019, Proceedings*, volume 11682 of *Lecture Notes in Computer Science*, pages 207–216. Springer, 2019.
- [88] J. Håstad. Clique is hard to approximate within  $n^{1-\epsilon}$ . *Electronic Colloquium on Computational Complexity (ECCC)*, 4(38), 1997.
- [89] S. Hoffmann. Computational complexity of synchronization under regular commutative constraints. *CoRR*, abs/2005.04042, 2020.
- [90] S. Hoffmann. Constraint synchronization with two or three state partial constraint automata. *CoRR*, abs/2005.05907, 2020.
- [91] S. Hoffmann. On a class of constrained synchronization problems in NP. *CoRR*, abs/2006.01903, 2020.
- [92] Hyunwoo Cho, Seh-Woong Jeong, F. Somenzi, and C. Pixley. Multiple observation time single reference test generation using synchronizing sequences. In *1993 European Conference on Design Automation with the European Event in ASIC Design*, pages 494–498, 1993.
- [93] H. Jürgensen. Synchronization. *Inf. Comput.*, 206(9-10):1033–1044, 2008.
- [94] J. Kari. A counter example to a conjecture concerning synchronizing words in finite automata. *Bull. European Assoc. Theor. Comput. Sci.*, 73:146, 2001.
- [95] J. Kari. Synchronizing finite automata on Eulerian digraphs. *Theoret. Comput. Sci.*, 295:223–232, 2003.
- [96] J. Kari and M. Volkov. Černý’s conjecture and the road coloring problem. In J.-E. Pin, editor, *Handbook of Automata Theory, Volume I*, chapter 15, pages 525–565. EMS Publishing House. (in print).
- [97] A. Kisielewicz, J. Kowalski, and M. Szykuła. Computing the shortest reset words of synchronizing automata. *J. Comb. Optim.*, 29(1):88–124, 2015.

- [98] A. Kisielewicz, J. Kowalski, and M. Szykuła. Experiments with synchronizing automata. In Y.-S. Han and K. Salomaa, editors, *Implementation and Application of Automata – 21st International Conference, CIAA 2016, Seoul, South Korea, July 19–22, 2016, Proceedings*, volume 9705 of *Lecture Notes in Comput. Sci.*, pages 176–188. Springer, 2016.
- [99] A. Kisielewicz and M. Szykuła. Generating small automata and the Černý conjecture. In S. Konstantinidis, editor, *Implementation and Application of Automata – 18th International Conference, CIAA 2013, Halifax, NS, Canada, July 16–19, 2013. Proceedings*, volume 7982 of *Lecture Notes in Comput. Sci.*, pages 340–348. Springer, 2013.
- [100] A. Kisielewicz and M. Szykuła. Synchronizing automata with extremal properties. In G. F. Italiano, G. Pighizzini, and D. Sannella, editors, *Mathematical Foundations of Computer Science 2015 – 40th International Symposium, MFCS 2015, Milan, Italy, August 24–28, 2015, Proceedings, Part I*, volume 9234 of *Lecture Notes in Comput. Sci.*, pages 331–343. Springer, 2015.
- [101] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–42. Princeton University Press, 1956.
- [102] A. A. Klyachko, I. K. Rystsov, and M. A. Spivak. An extremal combinatorial problem associated with the bound of the length of a synchronizing word in an automaton. *Cybernetics and System Analysis*, 23(2):165–171, 1987. Translated from *Kibernetika*, No. 2, 1987, pp. 16–20, 25.
- [103] Z. Kohavi and J. Winograd. Bounds on the length of synchronizing sequences and the order of information losslessness. In Z. Kohavi and A. Paz, editors, *Theory of Machines and Computations*, pages 197–206. Academic Press, 1971.
- [104] Z. Kohavi and J. Winograd. Establishing certain bounds concerning finite automata. *J. Comput. System Sci.*, 7(3):288–299, 1973.
- [105] D. Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 254–266. IEEE Computer Society, 1977.
- [106] M. W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36(3):490–509, 1988.
- [107] A. E. Laemmel. A general class of discrete codes and certain of their properties. Research Report R-459-55, PIB-389, Microwave Research Inst., Polytechnic Inst. Brooklyn, NY, 1956.
- [108] A. E. Laemmel. Study on application of coding theory. Technical Report PIBMRI-895.5-63, Dept. Electrophysics, Microwave Research Inst., Polytechnic Inst. Brooklyn, NY, 1963.
- [109] A. E. Laemmel and B. Rudner. Study of the application of coding theory. Technical Report PIBEP-69-034, Dept. Electrophysics, Polytechnic Inst. Brooklyn, Farmingdale, NY, 1969.
- [110] В. И. Левенштейн. Самонастраивающиеся автоматы для декодирования сообщений. *Докл. АН СССР*, 141(6):1320–1323, 1961.
- [111] C. L. Liu. Determination of the final state of an automaton whose initial state is unknown. *IEEE Transactions on Electronic Computers*, EC-12(5):918–920, 1963.
- [112] C. L. Liu. Some memory aspects of finite automata. Technical Report 411, Research Lab. Electronics, Massachusetts Inst. Technology, Cambridge, MA, 1963.
- [113] P. V. Martugin. A series of slowly synchronizing automata with zero state over a small alphabet. *Information and Computer*, 19:517–536, 2009.
- [114] M. I. Maslennikova. Reset complexity of ideal languages over a binary alphabet. *Int. J. Found. Comput. Sci.*, 30(6-7):1177–1196, 2019.
- [115] M. I. Maslennikova and E. Rodaro. Representation of (left) ideal regular languages by synchronizing automata. In L. D. Beklemishev and D. V. Musatov, editors, *Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*, volume 9139 of *Lecture Notes in Computer Science*, pages 325–338. Springer, 2015.

- [116] M. I. Maslennikova and E. Rodaro. Trim strongly connected synchronizing automata and ideal languages. *Fundam. Inform.*, 162(2-3):183–203, 2018.
- [117] A. Mateescu and A. Salomaa. Many-valued truth functions, Černý’s conjecture, and road coloring. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science, Entering the 21th Century*, pages 693–707. World Scientific, 2001.
- [118] Ю. Т. Медведев. О классе событий, допускающих представление в конечном автомате. In A. А. Ляпунов, editor, *Автоматы. Сб. статей под редакцией К. Э. Шеннона и Дж. Маккарти*, pages 385–401. ИЛ, 1956.
- [119] E. Moore. On certain crinkly curves. *Trans. Amer. Math. Soc.*, 1:72–90, 1900. Correction: *Trans. Amer. Math. Soc.* 1 (1900), 507.
- [120] E. F. Moore. Gedanken-experiments on sequential machines. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 129–153. Princeton University Press, 1956.
- [121] B. K. Natarajan. An algorithmic approach to the automated design of parts orienters. In *Proc. 27th Annual Symp. Foundations Comput. Sci.*, pages 132–142. IEEE Press, 1986.
- [122] B. K. Natarajan. Some paradigms for the automated design of parts feeders. *Internat. J. Robotics Research*, 8(6):89–109, 1989.
- [123] C. Nicaud. The černý conjecture holds with high probability. *J. Autom. Lang. Comb.*, 24(2-4):343–365, 2019.
- [124] J. Olschewski and M. Ummels. The complexity of finding reset words in finite automata. In P. Hliněný and A. Kučera, editors, *Mathematical Foundations of Computer Science*, volume 6281 of *Lecture Notes in Comput. Sci.*, pages 568–579. Springer, 2010.
- [125] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [126] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *J. Comput. System Sci.*, 28(2):244–259, 1984.
- [127] J.-E. Pin. *Le problème de la synchronisation et la conjecture de Černý*. Thèse de 3ème cycle, Université Paris VI, 1978. (in French).
- [128] J.-E. Pin. Sur un cas particulier de la conjecture de Černý. In *Proc. 5th Colloq. on Automata, Languages, and Programming (ICALP)*, volume 62 of *Lecture Notes in Comput. Sci.*, pages 345–352. Springer, 1978. (in French).
- [129] J.-E. Pin. On two combinatorial problems arising from automata theory. *Ann. Disc. Math.*, 17:535–548, 1983.
- [130] C. Pixley, S. Jeong, and G. D. Hachtel. Exact calculation of synchronizing sequences based on binary decision diagrams. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 13(8):1024–1034, 1994.
- [131] R. Pöschel, M. Sapir, N. Sauer, M. Stone, and M. Volkov. Identities in full transformation semigroups. *Algebra Universalis*, 31(4):580–588, 1994.
- [132] E. V. Pribavkina. Slowly synchronizing automata with zero and noncomplete sets. *Math. Notes*, 90:411–417, 2011.
- [133] E. V. Pribavkina and E. Rodaro. Synchronizing automata with finitely many minimal synchronizing words. *Inf. Comput.*, 209(3):568–579, 2011.
- [134] N. Pytheas Fogg. *Substitutions in dynamics, arithmetics and combinatorics*, volume 1794 of *Lecture Notes in Mathematics*. Springer, 2002. Edited by V. Berthé, S. Ferenczi, C. Mauduit and A. Siegel.
- [135] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. Develop.*, 3(2):114–125, 1959.
- [136] J. L. Ramírez Alfonsín. *The diophantine Frobenius problem*. Oxford University Press, 2005.
- [137] A. S. Rao and K. Y. Goldberg. Manipulating algebraic parts in the plane. *IEEE Trans. Robotics Autom.*, 11(4):598–602, 1995.

- [138] R. Reis and E. Rodaro. Ideal regular languages and strongly connected synchronizing automata. *Theor. Comput. Sci.*, 653:97–107, 2016.
- [139] J.-K. Rho, F. Somenzi, and C. Pixley. Minimum length synchronizing sequences of finite state machine. In *Proc. 30th Design Automation Conf.*, pages 463–468. ACM, 1993.
- [140] E. Rodaro. Strongly connected synchronizing automata and the language of minimal reset words. *Adv. Appl. Math.*, 99:158–173, 2018.
- [141] E. Rodaro. A bound for the length of the shortest reset words for semisimple synchronizing automata via the packing number. *J. Algebr. Comb.*, 50:237–253, 2019.
- [142] A. Roman and M. Szykuła. Forward and backward synchronizing algorithms. *Expert Syst. Appl.*, 42(24):9512–9527, 2015.
- [143] I. Rystsov. Reset words for commutative and solvable automata. *Theor. Comput. Sci.*, 172(1-2):273–279, 1997.
- [144] I. K. Rystsov. Polynomial complete problems in automata theory. *Inf. Process. Lett.*, 16(3):147–151, 1983.
- [145] I. K. Rystsov. Almost optimal bound of recurrent word length for regular automata. *Cybernetics and System Analysis*, 31:669–674, 1995. Translated from *Kibernetika i Sistemnyj Analiz*, No. 5, 1995, pp. 40–48.
- [146] I. K. Rystsov. Quasioptimal bound for the length of reset words for regular automata. *Acta Cybernetica*, 12:145–152, 1995.
- [147] I. K. Rystsov. Exact linear bound for the length of reset words in commutative automata. *Publ. Math. Debrecen*, 46(3-4):405–409, 1996.
- [148] I. K. Rystsov. Reset words for automata with simple idempotents. *Cybernetics and System Analysis*, 36:339–344, 2000. Translated from *Kibernetika i Sistemnyj Analiz*, No. 3, 2000, pp. 32–39.
- [149] И. К. Рысцов. Оценка длины ядерного слова для конечного автомата. In М. А. Спивак, editor, *Автоматы. Межвуз. науч. сб.*, volume 2, pages 43–48. Саратов. гос. ун-т, 1977.
- [150] И. К. Рысцов. О минимизации синхронизирующих слов для конечных автоматов. In А. А. Летичевский, editor, *Теоретические вопросы проектирования вычислительных систем*, pages 75–82. АН УССР, Науч. совет по пробл. «Кибернетика», Ин-т кибернетики, 1980.
- [151] И. К. Рысцов. О ранге конечного автомата. *Киб. и систем. анализ*, 28(3):3–10, 187, 1992. (English transl.: On the rank of a finite automaton, *Cybernet. Systems Anal.* 28 (1992), no. 3, 323–328).
- [152] И. К. Рысцов. О ранге конечного автомата. *Возвратные слова для разрешимых автоматов*, 30(5):21–26, 1994.
- [153] A. Ryzhikov. Synchronization problems in automata without non-trivial cycles. *Theor. Comput. Sci.*, 787:77–88, 2019.
- [154] A. Ryzhikov and M. Szykula. Finding short synchronizing words for prefix codes. In I. Potapov, P. G. Spirakis, and J. Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPICs*, pages 21:1–21:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [155] A. Salomaa. Compositions over a finite domain: From completeness to synchronizable automata. In A. Salomaa, D. Wood, and S. Yu, editors, *A Half-Century of Automata Theory: Celebration and Inspiration*, pages 131–143. World Scientific, 2000.
- [156] A. Salomaa. Generation of constants and synchronization of finite automata. *J. UCS*, 8(2):332–347, 2002.

- [157] A. Salomaa. Synchronization of finite automata: Contributions to an old problem. In T. Æ. Mogensen, D. A. Schmidt, and I. H. Sudborough, editors, *The Essence of Computation, Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones [on occasion of his 60th birthday]*, volume 2566 of *Lecture Notes in Computer Science*, pages 37–59. Springer, 2002.
- [158] A. Salomaa. Composition sequences for functions over a finite domain. *Theoret. Comput. Sci.*, 292:263–281, 2003.
- [159] S. Sandberg. Homing and synchronizing sequences. In M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, editors, *Model-Based Testing of Reactive Systems*, volume 3472 of *Lecture Notes in Comput. Sci.*, pages 5–33. Springer, 2005.
- [160] M.-P. Schützenberger. On an application of semi groups methods to some problems in coding. *IRE Transactions on Information Theory*, 2(3):47–60, 1956.
- [161] Y. Shitov. An improvement to a recent upper bound for synchronizing words of finite automata. *J. Autom. Lang. Comb.*, 24(2-4):367–373, 2019.
- [162] E. Skvortsov and E. Tipikin. Experimental study of the shortest reset word of random automata. In B. Bouchou-Markhoff, P. Caron, J.-M. Champarnaud, and D. Maurel, editors, *Implementation and Application of Automata – 16th International Conference, CIAA 2011, Blois, France, July 13–16, 2011. Proceedings*, volume 6807 of *Lecture Notes in Comput. Sci.*, pages 290–298. Springer, 2011.
- [163] P. H. Starke. Eine Bemerkung über homogene Experimente. *Elektronische Informationsverarbeitung und Kybernetik*, 2:257–259, 1966. (in German).
- [164] P. H. Starke. *Abstrakte Automaten*. Deutscher Verlag der Wissenschaften, 1969. (in German).
- [165] P. H. Starke. *Abstract automata*. North-Holland and American Elsevier, 1972. (English translation of [164]).
- [166] P. H. Starke. A remark about homogeneous experiments. *J. Autom. Lang. Comb.*, 24(2-4):133–137, 2019. (English translation of [163]).
- [167] B. Steinberg. Černý’s conjecture and group representation theory. *J. Algebr. Comb.*, 34:83–109, 2010.
- [168] B. Steinberg. The averaging trick and the Černý conjecture. *Int. J. Found. Comput. Sci.*, 22(7):1697–1706, 2011.
- [169] B. Steinberg. The Černý conjecture for one-cluster automata with prime length cycle. *Theoret. Comput. Sci.*, 412:5487–5491, 2011.
- [170] M. Suomalainen, A. Q. Nilles, and S. M. LaValle. Virtual reality for robots. *CoRR*, abs/1909.07096, 2019.
- [171] M. Szykuła. *Algorithms for synchronizing automata*. Ph.D. Thesis, Institute of Computer Science, University of Wrocław, 2014.
- [172] M. Szykuła. Improving the upper bound on the length of the shortest reset word. In R. Niedermeier and B. Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28–March 3, 2018, Caen, France*, volume 96 of *LIPICs*, pages 56:1–56:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [173] M. Szykuła and V. Vorel. An extremal series of eulerian synchronizing automata. In S. Brlek and C. Reutenauer, editors, *Developments in Language Theory – 20th International Conference, DLT 2016, Montréal, Canada, July 25–28, 2016, Proceedings*, volume 9840 of *Lecture Notes in Comput. Sci.*, pages 380–392. Springer, 2016.
- [174] A. Trahtman. An efficient algorithm finds noticeable trends and examples concerning the Černý conjecture. In R. Kráľovič and P. Urzyczyn, editors, *31st Int. Symp. Math. Foundations of Comput. Sci.*, volume 4162 of *Lecture Notes in Comput. Sci.*, pages 789–800. Springer, 2006.
- [175] A. Trahtman. The Černý conjecture for aperiodic automata. *Discrete Math. & Theoret. Comput. Sci.*, 9(2):3–10, 2007.

- [176] A. Trahtman. The Road Coloring Problem. *Israel J. Math.*, 172(1):51–60, 2009.
- [177] A. Trahtman. Modifying the upper bound on the length of minimal synchronizing word. In O. Owe, M. Steffen, and J. Telle, editors, *Fundamentals of Computation Theory*, volume 6914 of *Lecture Notes in Comput. Sci.*, pages 173–180. Springer, 2011.
- [178] A. Trahtman. The Černý conjecture. *CoRR*, abs/1202.4626, 2012. (withdrawn).
- [179] A. Trahtman. The length of a minimal synchronizing word and the Černý conjecture. *CoRR*, abs/1405.2435, 2014.
- [180] A. Trahtman. Cerny-Starke conjecture from the sixties of XX century. *CoRR*, abs/2003.06177, 2020.
- [181] A. Trkhtman. Some aspects of synchronization of DFA. *J. Comput. Sci. Technol.*, 23(5):719–727, 2008.
- [182] M. V. Volkov. Synchronizing automata and the Černý conjecture. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *Language and Automata Theory and Applications*, volume 5196 of *Lecture Notes in Comput. Sci.*, pages 11–27. Springer, 2008.
- [183] M. V. Volkov. Synchronizing automata preserving a chain of partial orders. *Theor. Comput. Sci.*, 410(37):3513–3519, 2009.
- [184] M. V. Volkov. Slowly synchronizing automata with idempotent letters of low rank. *J. Autom. Lang. Comb.*, 24(2-4):375–386, 2019.
- [185] V. Vorel. *Synchronization and Discontinuous Input Processing in Transition Systems*. Doctoral thesis, Charles University, Prague, 2018.
- [186] V. Vorel and A. Roman. Parameterized complexity of synchronization and road coloring. *Discrete Math. & Theoret. Comput. Sci.*, 17(1):283–306, 2015.
- [187] H. Wielandt. Unzerlegbare, nicht negative Matrizen. *Math. Z.*, 52:642–648, 1950. (in German).
- [188] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(1):103–128, 2007.

**М. В. Волков (M. V. Volkov)**

Уральский федеральный университет

*E-mail:* m.v.volkov@urfu.ru