Библиотека
«Математическое просвещение»
Выпуск 10

А. Б. Сосинский

Узлы и косы

Рисунки М. Ю. Панова

Издательство Московского центра
непрерывного математического образования
Москва • 2001

УДК 515.16
С66
ББК 22.15

Аннотация

Красивые и наглядные понятия узла и косы сейчас в центре внимания современной математики и физики. В брошюре обсуждаются их простейшие геометрические и алгебраические свойства и их компьютерная обработка.

Текст брошюры представляет собой дополненную обработку записи лекции, прочитанной автором 7 октября 2000 года на Малом мехмате для школьников 9–11 классов.

Брошюра рассчитана на широкий круг читателей, интересующихся математикой: школьников старших классов, студентов младших курсов, учителей...

РФФИ

Работа автора над брошюрой частично поддержана Российским фондом фундаментальных исследований (РФФИ), грант № 98–01–00555.

ISBN 5–900916–76–6

Сосинский Алексей Брониславович

Узлы и косы

(Серия: «Библиотека „Математическое просвещение“»)
М.: МЦНМО, 2001. — 24 с.: ил.

Главный редактор серии *В. М. Тихомиров*.

Редакторы *А. А. Ермаченко*, *Е. Н. Осьмова*. Техн. редактор *М. Ю. Панов*.

Лицензия ИД № 01335 от 24/III 2000 года. Подписано к печати 25/I 2001 года. Формат бумаги 60×88 $^1/_{16}$. Офсетная бумага № 1. Офсетная печать. Физ. печ. л. 1,50. Усл. печ. л. 1,50. Уч.-изд. л. 1,38. Тираж 3000 экз. Заказ 4248.

Издательство Московского центра непрерывного математического образования.
121002, Москва, Г-2, Бол. Власьевский пер., 11. Тел. 241–05–00.

Отпечатано в Производственно-издательском комбинате ВИНИТИ.
140010, г. Люберцы Московской обл., Октябрьский пр-т, 403. Тел. 554–21–86.

Узлы и косы — предметы простые и наглядные. Вы, конечно, встречались с ними в повседневной жизни, но, может быть, не подозревали, что это ещё и математические объекты; более того, в последние 20 лет математики и физики с огромным интересом и удивительной интенсивностью стали заниматься соответствующими теориями, особенно теорией узлов. Достаточно сказать, что за это время четыре медали Филдса*) были получены именно за работы, связанные с этой теорией. А именно, лауреатами медали Филдса в разное время стали Владимир Дринфельд из Харькова, работающий в Чикаго, Максим Концевич из Москвы, работающий в Париже, Воган Джонс из Новой Зеландии, работающий в Калифорнии, и Эдвард Виттен, физик-теоретик, работающий в Принстоне.

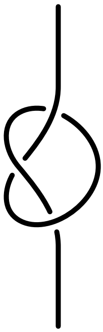


Рис. 1

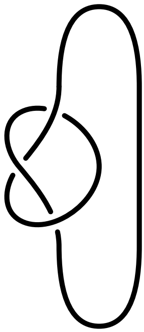


Рис. 2

* * *

Чем отличается математический узел от узлов, которые завязывают на галстуках или на шнурках ботинок? Естественно, в математике *узел* — это некая абстракция: рассматривается не верёвка и не шнур, а бесконечно тонкая, гибкая и растяжимая нить. Кроме того, рассматривая математический узел, нужно либо как-то зафиксировать его концы (обычно говорят, что один конец уходит в бесконечность «вверх», а другой — в бесконечность «вниз», рис. 1), либо просто соединить их (рис. 2). В последнем случае модель узла — замкнутая несамопересекающаяся кривая в пространстве. Будем предполагать, что эта кривая является ломаной, т. е. состоит из отрезков (впрочем, на рисунках мы почти всегда будем изображать узлы в виде гладких кривых, считая отдельные звенья ломаной очень маленькими).

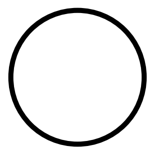


Рис. 3

Самый простой узел — *тривиальный* (рис. 3). Узел называется *нетривиальным*, если он не эквивалентен тривиальному, т. е. его нельзя «пошевелить» (возможно растягивая, но не разрывая верёвку) так, чтобы он превратился в тривиальный.

*) Медаль Филдса — это как Нобелевская премия, только по математике. Математикам не дают Нобелевских премий (это противоречит завещанию Нобеля), и самой высокой наградой в математике считается медаль Филдса.

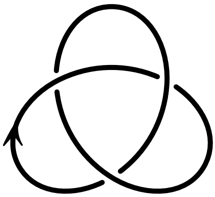


Рис. 4

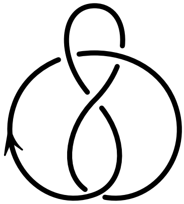


Рис. 5

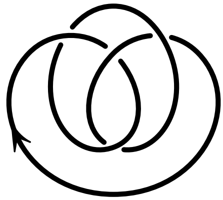


Рис. 6

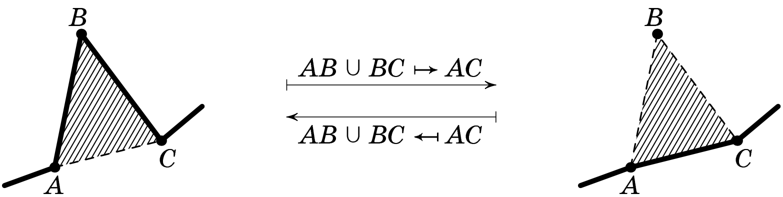


Рис. 7

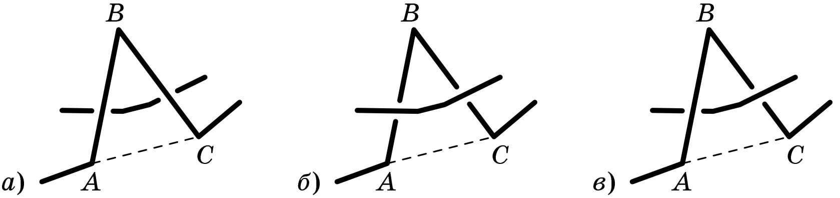


Рис. 8

Вот несколько примеров нетривиальных узлов: узел на рис. 4 называется *трилистником*, узел на рис. 5 — *восьмёркой*. (Обычно узлы рассматривают с *ориентацией*, т. е. считают, что задано направление обхода кривой, это направление изображается стрелкой.) Подумайте, как назвать узел, изображённый на рис. 6.

Дадим математически строгое определение эквивалентности узлов. Напомним, что узел — это ломаная. С этой ломаной можно производить следующие *элементарные операции* (рис. 7):

(1) два последовательных звена AB и BC ломаной заменить звеном AC;

(2) звено AC заменить двузвенной ломаной $AB \cup BC$.

Обе операции разрешены, только если треугольник ABC не пересекается (в пространстве) ни с какими другими кусками нашего узла. Например, в ситуациях, показанных на рис. 8, *а*, *б*, эти операции производить можно, а в ситуации, показанной на рис. 8, *в*, — нельзя.

Теперь назовём два узла *эквивалентными*, если их можно элементарными операциями превратить в совершенно одинаковые (совмещаемые сдвигом) узлы. Под словом узел мы часто будем понимать не только конкретную кривую, но и весь класс эквивалентности этой кривой.

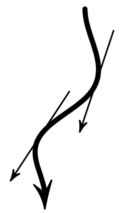


Рис. 9

* * *

Математическая *коса* состоит из n нитей (т. е. кривых в пространстве), которые начинаются в n точках горизонтальной прямой и заканчиваются в n точках другой горизонтальной прямой, расположенной ниже. При этом нити должны быть нисходящими, т. е. касательный вектор в любой точке кривой должен всё время «смотреть вниз» (рис. 9), ему запрещается быть горизонтальным и тем более «смотреть вверх».

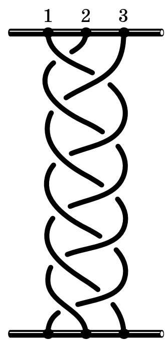


Рис. 10

На рис. 10 изображена коса из трёх нитей — коса *девичья*. На рис. 11 приведён пример косы из четырёх нитей.

Эквивалентность кос определяется точно так же, как эквивалентность узлов, но с единственным ограничением: разрешены лишь те элементарные преобразования, в результате которых нити ко́сы остаются нисходящими (например, преобразование, изображённое на рис. 12, недопустимо).

Итак, как и узлы, косы не просто конкретные объекты, а классы эквивалентности этих объектов.

* * *

Существует ли какая-нибудь связь между узлами и косами?

Вот способ превратить косу в узел: надо *замкнуть* её, т. е. соединить верхние концы нитей с нижними (не запутывая между собой соединяющие нити). Например, в результате замыкания косы из

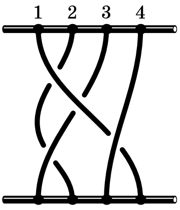


Рис. 11

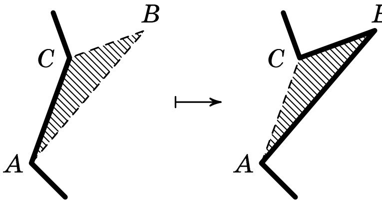


Рис. 12

двух нитей, изображённой на рис. 13, получается узел, эквивалентный трилистнику. Но не всегда при замыкании косы образуется узел: после замыкания косы на рис. 14 получаются две замкнутые кривые, которые между собой зацеплены (одна как бы обматывается вокруг другой).

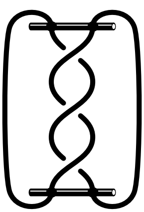


Рис. 13

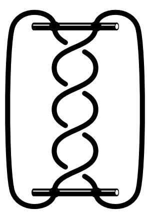


Рис. 14

Другой естественный вопрос заключается в следующем: а всякий ли узел можно получить, замкнув некоторую косу? Ответ даёт

Теорема Александера*). Любой узел — это замкнутая коса.

Мы приведём здесь лишь идею доказательства этой теоремы.

*) Джеймс Уэнделл Александер (1888—1971) — знаменитый американский тополог.

Узел трилистник обладает замечательным свойством: если смотреть из точки O на любой его участок, направление движения по этому участку всегда видится слева направо (рис. 15), т. е. узел как бы обматывается вокруг этой точки. Тем же свойством обладает узел,

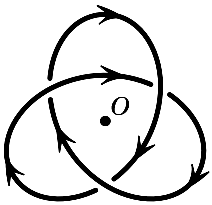


Рис. 15

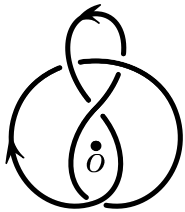


Рис. 16

изображённый на рис. 6. А вот восьмёрка этим свойством не обладает: можно проверить, что какую бы точку O мы ни взяли, всегда найдутся и участки, которые обходятся слева направо, и участки, которые обходятся справа налево (рис. 16). Значит, этот узел не обматывается.

Если узел обматывается вокруг некоторой точки, то очень легко построить косу, замыканием которой он является: разрежем наш узел по лучу, проведённому из этой точки, и развернём его (можно представить, что наш узел заключён в подкову, а мы как бы выпрямляем эту подкову, превращая её в прямоугольник, рис. 17). Легко видеть, что замыканием полученной косы является исходный узел.

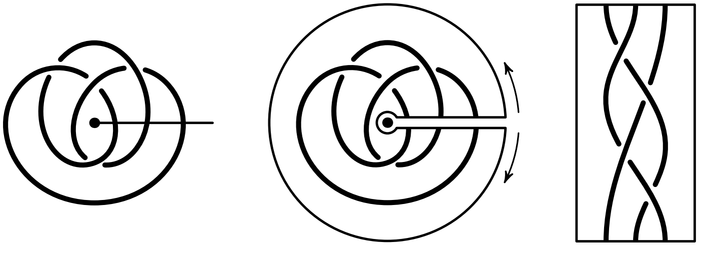


Рис. 17

Как же быть, если узел не обматывается? Оказывается, довольно несложно превратить его в узел, который обматывается. Мы не будем этого подробно доказывать, но продемонстрируем основную идею.

Возьмём участок, который обходится «не в ту сторону», и перекинем его через точку O так, чтобы он обходился в нужную сторону (рис. 18)*). Так можно поочерёдно избавиться от всех «неправильных» участков (на самом деле это не совсем просто, и в строгом доказательстве есть некоторые тонкие моменты).

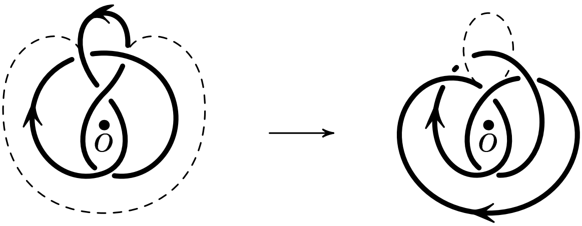


Рис. 18

Алгебра кос и узлов

Оказывается, можно рассматривать узлы и косы как алгебраические объекты: их можно у м н о ж а т ь, и это умножение обладает многими свойствами обычного умножения чисел.

Начнём с кос. Возьмём две косы a и b с одинаковым числом нитей и соединим нижние концы нитей первой косы с верхними концами нитей второй косы (рис. 19); полученную косу, сжатую в два раза в вертикальном направлении, называют *произведением* этих двух кос и обозначают ab.

Сразу ясно, что это умножение *ассоциативно*, т. е. для любых трёх кос a, b и c косы $(a \cdot b) \cdot c$ и $a \cdot (b \cdot c)$ эквивалентны.

Тривиальная коса (рис. 20) играет роль *единицы* (и поэтому обозначается $\mathbb{1}$): для любой косы a

$$a \cdot \mathbb{1} = a = \mathbb{1} \cdot a.$$

В самом деле, «подклеив» к любой косе тривиальную, мы можем так «пошевелить» новую косу, что получится снова та же самая коса.

Пусть имеется коса b. Как построить *обратную* косу, т. е. такую косу b^{-1}, при умножении которой на b получается $\mathbb{1}$? Очень просто:

*) Заметим, что мы попутно доказали эквивалентность узлов, изображённых на рис. 5 и 6. Следовательно, узел на рис. 6 тоже следует называть восьмёркой.

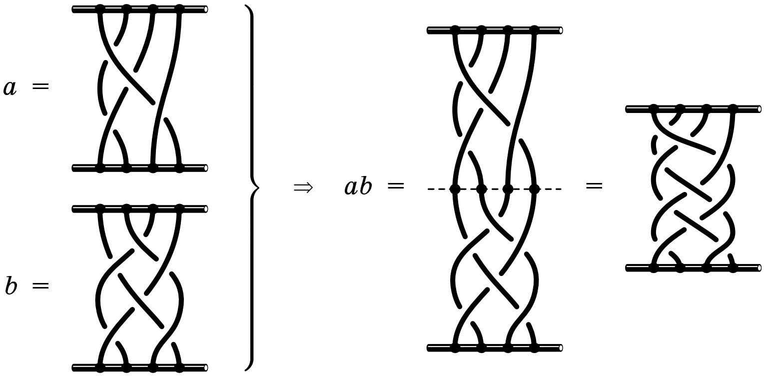


Рис. 19

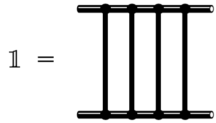


Рис. 20

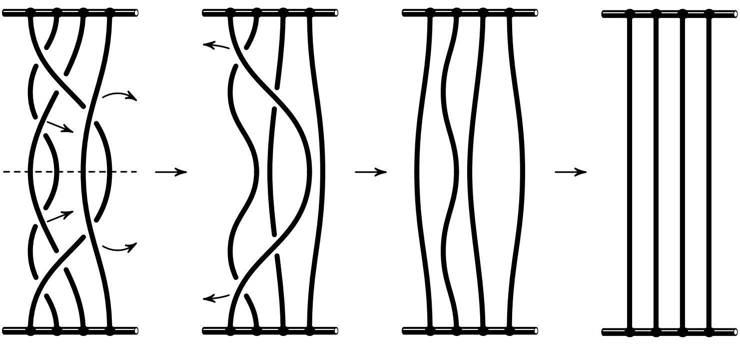


Рис. 21

нужно зеркально отразить косу b относительно горизонтальной плоскости. Легко понять, что косу, «склеенную» из двух симметричных, можно «расплести» (рис. 21).

Мы привыкли, что произведение двух чисел не зависит от порядка сомножителей. Увы, для кос это не так: умножение кос *некоммутативно*. Придумать две косы a и b, произведение которых зависит

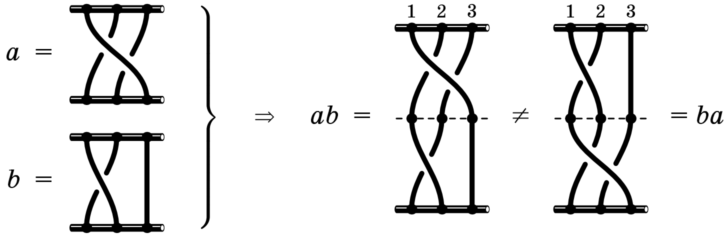


Рис. 22

от порядка, несложно (рис. 22). Косы ab и ba действительно различны (неэквивалентны): первая нить косы ab приходит в крайнее правое положение, а первая нить косы ba — в крайнее левое.

Подведём итог в виде теоремы.

Теорема о косах. Умножение кос обладает следующими свойствами:

1° (ассоциативность) для любых кос a, b и c

$$(a \cdot b) \cdot c = a \cdot (b \cdot c);$$

2° (наличие единицы) существует такая коса $\mathbb{1}$, что для любой косы a

$$a \cdot \mathbb{1} = a = \mathbb{1} \cdot a;$$

3° (наличие обратного элемента) для любой косы b найдётся такая коса b^{-1}, что

$$bb^{-1} = \mathbb{1} = b^{-1}b.$$

Иными словами, косы образуют *группу*. Эта группа некоммутативна.

* * *

Оказывается, в «алгебраизации» теории кос можно пойти ещё дальше, выхолостив всю геометрию и заменив её алгебраическими преобразованиями буквенных выражений.

Обозначим через b_i косу из n нитей, i-я нить которой проходит «под» $(i+1)$-й, а остальные нити вертикальны (рис. 23). Символов b_1, b_2, ..., b_{n-1} достаточно, чтобы кодировать любую косу из n нитей: «разрежем» косу горизонтальными линиями так, чтобы между двумя

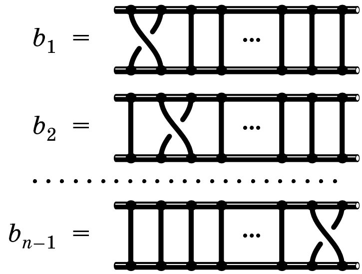


Рис. 23

соседними линиями оказался ровно один «перекрёсток», т. е. коса b_i или b_i^{-1} *), и выпишем подряд все полученные буквы (рис. 24).

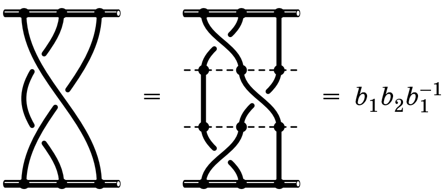


Рис. 24

Умножение кос в этих обозначениях записывается очень просто, например,

$$b_1 b_3^{-1} b_2 \cdot b_3^{-1} b_2^2 b_1^{-1} = b_1 b_3^{-1} b_2 b_3^{-1} b_2^2 b_1^{-1}$$

(b_2^2 означает $b_2 b_2$), т. е. достаточно «стереть» знак умножения.

Некоторые записи можно *сократить*: например, $b_1 b_3 b_3^{-1} b_2^{-1} b_2^2$ и $b_1 b_2$ — одна и та же коса, поскольку

$$b_i b_i^{-1} = \mathbb{1} = b_i^{-1} b_i \quad \text{при } 1 \leqslant i \leqslant n-1. \tag{1}$$

Между символами b_i есть и другие замечательные соотношения. Например,

$$b_i b_j = b_j b_i, \quad \text{при } |i-j| \geqslant 2,\ 1 \leqslant i,\ j \leqslant n-1. \tag{2}$$

---

*) Легко видеть, что в каждом классе эквивалентности кос найдётся коса, для которой это сделать можно.

Обратите внимание, что это соотношение выполнено не для всех i, j, иначе умножение было бы коммутативным. Но если $i+1 < j$ (случай $j+1 < i$ полностью аналогичен), можно получить из косы $b_i b_j$ косу $b_j b_i$, «сдвинув» точку пересечения i-й и $(i+1)$-й нитей вниз, а j-й и $(j+1)$-й — вверх (рис. 25). Таким образом, эти косы эквивалентны.

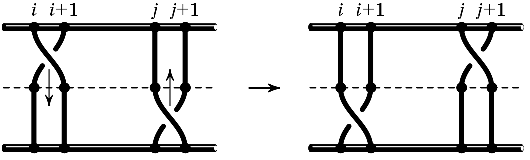


Рис. 25

Кроме того, имеет место формула

$$b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1}, \quad \text{при } 1 \leqslant i \leqslant n-2 \tag{3}$$

(косу $b_i b_{i+1} b_i$ также можно превратить в $b_{i+1} b_i b_{i+1}$ последовательными преобразованиями, рис. 26).

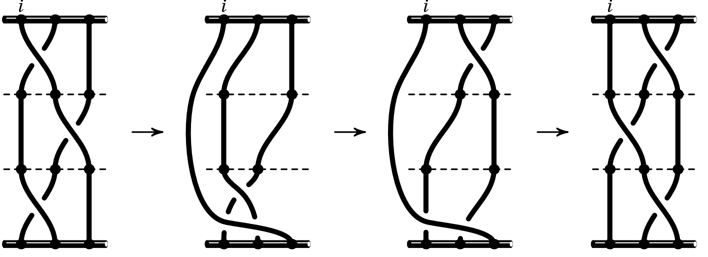


Рис. 26

Теорема Артина*). В группе кос все равенства вытекают из соотношений (1), (2), (3).

Иными словами, любое верное равенство можно получить, комбинируя эти соотношения. Например, равенство

$$b_1 b_2 b_1 b_3 b_5 = b_2 b_1 b_2 b_5 b_3$$

можно получить, перемножив равенства

$$b_1 b_2 b_1 = b_2 b_1 b_2 \qquad \text{и} \qquad b_3 b_5 = b_5 b_3$$

---

*) Эмиль Артин (1898—1962) — известный немецкий алгебраист, родоначальник теории кос.

(при этом из-за некоммутативности умножать нужно обязательно с одной и той же стороны). При помощи нескольких подобных операций можно получить любое равенство в группе кос.

Более того (и это замечательное достижение самых последних лет), существуют очень быстрые алгоритмы, основанные на теореме Артина, которые позволяют узнать, являются ли два элемента группы кос эквивалентными. Можно, конечно, эту задачу решать геометрически — нарисовав две сложные косы, пытаться превратить одну в другую с помощью элементарных преобразований. Если это не удаётся быстро сделать практически, то это ещё не значит, что эти косы разные. Оказывается, что этой сложной геометрией заниматься не нужно: можно закодировать эти косы и к полученным словам, состоящим из букв b_1, b_2, ..., b_{n-1}, применить чисто алгебраический алгоритм, который и выяснит, совпадают ли эти косы.

* * *

Соотношение (3) называют *уравнением Артина* или *уравнением кос*, а в последнее время употребляется ещё и термин *уравнение Янга—Бакстера*. Янг — физик китайского происхождения, лауреат Нобелевской премии — вывел это уравнение, исследуя теорию элементарных частиц на прямой. Его уравнение описывает взаимодействие трёх частиц, которые меняются местами. Бакстер — австралийский математик, специалист по статистической физике — он рассматривал модели сплошной среды, изучал процесс превращения воды в лёд, и в своей работе он вывел уравнение, которое формально выглядит точно так же. Никакой связи между Янгом и Бакстером не было, они занимались совершенно разными задачами, но это уравнение было получено ими примерно в одно и то же время, и его стали называть уравнением Янга—Бакстера. Потом выяснилось, что это уравнение совпадает с уравнением кос, и вообще теория кос тесно связана с физикой — квантовой, статистической... В последние годы физики интересуются теорией кос не меньше, чем математики. Недаром одну из филдсовских медалей (которые присуждаются только за математические достижения) получил физик Виттен.

* * *

А можно ли умножать узлы? Если считать узлы кривыми, концы которых уходят в бесконечность, то умножение узлов определяется естественным образом: произведение узлов a и b — это просто нить, на которой завязан сначала узел a, затем узел b (рис. 27). Это умножение, как и умножение кос, ассоциативно: для любых узлов a, b и c

$$(a \cdot b) \cdot c = a \cdot (b \cdot c).$$

Ясно, что тривиальный узел (т. е. просто вертикальная прямая) является единичным элементом (рис. 28). Но, в отличие от случая кос, ни один нетривиальный узел не имеет обратного: если мы завяжем на нити два узла, а затем потянем за концы, узлы н е р а з в я ж у т с я (доказательство этого факта мы приводить не будем).

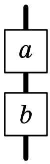


Рис. 27

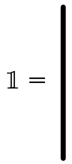


Рис. 28

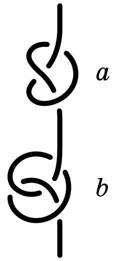


Рис. 29

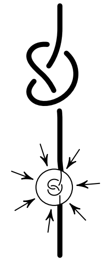


Рис. 30

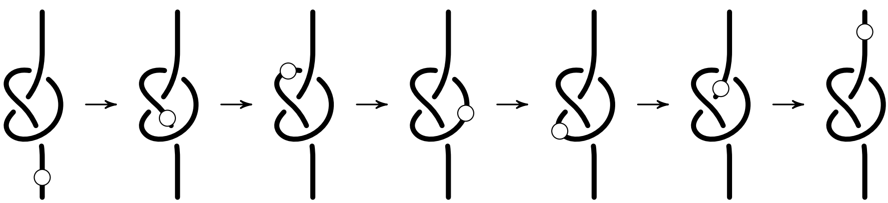


Рис. 31

Покажем, что два узла, завязанных на одной верёвке, можно переставить. Действительно, пусть на нити завязан сначала узел a, затем узел b (рис. 29). Сперва, не трогая узел a, «затянем» узел b в маленький узелок (рис. 30). Потом заключим этот узелок в маленький стеклянный шарик и будем двигать его вверх по нити (рис. 31). В итоге этот шарик окажется наверху, и его можно превратить опять в узел b. Таким образом, умножение узлов коммутативно: $a \cdot b = b \cdot a$.

Итак, верна

Теорема об узлах. Узлы образуют ассоциативную и коммутативную систему относительно умножения. В этой системе есть единичный элемент, но нет обратных элементов.

Компьютер развязывает узел

Ещё совсем недавно считалось, что компьютер — это вычислительная машина, хорошо и быстро умеющая считать, но сильно уступающая человеку в интуиции, в творческих возможностях, в пространственном воображении. Но это мнение сегодня пересматривается. У моего небольшого ноутбука (компьютера размером с маленькую конторскую папку) с пространственным воображением, видимо, всё в порядке — во всяком случае, ему часто удаётся развязывать узлы гораздо быстрее меня!

Чтобы объяснить как мой ноутбук это делает, нам потребуется немного теории (придуманной, кстати, в докомпьютерную эру живыми людьми).

Первый шаг в этой теории состоит в сведении (сложной) пространственной задачи развязывания узла к (более простой) задаче применения простых операций к кривым на плоскости. Эти операции придумал в 1920-е годы немецкий математик Рейдемейстер*), они изображены на рис. 32.

Операции, показанные на рис. 32, выполняются так. Пусть дана плоская диаграмма (проекция) узла, который мы намереваемся распутать, скажем, диаграмма узла на рис. 33, *а*. Проведём на диаграмме пунктирную окружность так, чтобы внутри неё оказалась одна из конфигураций, выделенных пунктирными окружностями на рис. 32, и заменим её внутри окружности на парную конфигурацию. Так, на рис. 33, *а* мы провели окружность, охватывающую «петельку» и, применив операцию Ω_1, заменили её на дугу без петли, получив диаграмму 33, *б*. Затем на этом рисунке мы провели окружность, охватывающую конфигурацию из трёх дуг и, применив операцию Ω_3, получили диаграмму 33, *в*. Читателю предлагается самостоятельно отследить процесс распутывания до конца, поочерёдно обращаясь к рис. 32 и 33.

Оказывается, что любой (тривиальный!) узел можно развязать таким образом. А именно, имеет место

Лемма Рейдемейстера. Если узел можно развязать (превратить в окружность) в пространстве, то его плоскую диаграмму можно распутать на плоскости с помощью операций Рейдемейстера.

*) Курт Вернер Фридрих Рейдемейстер (1893—1971) — тополог, один из основателей математической теории узлов.

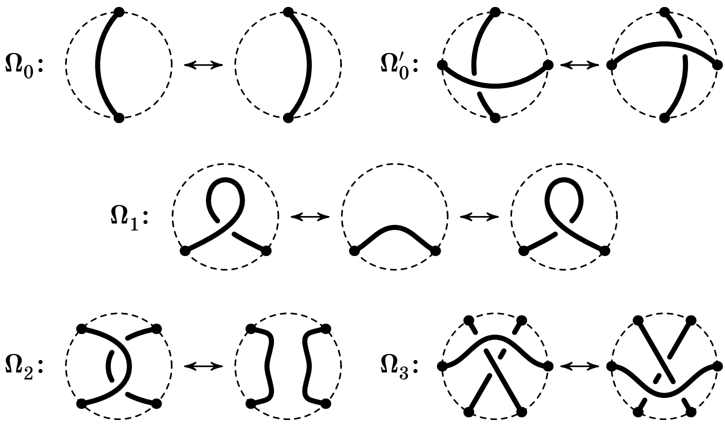

Ω_0:
Ω'_0:
Ω_1:
Ω_2:
Ω_3:

Рис. 32

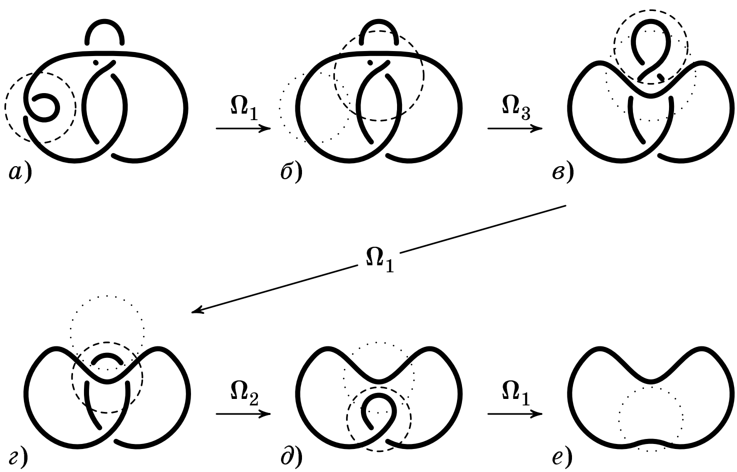

Ω_1
Ω_3
Ω_1
Ω_2
Ω_1
а)
б)
в)
г)
д)
е)

Рис. 33

Чтобы доказать эту лемму, достаточно проверить, что одну элементарную операцию (см. рис. 7) можно реализовать на плоской проекции с помощью конечного набора операций Ω_0, Ω_0', Ω_1, Ω_2, Ω_3. Эта проверка показана на одном конкретном примере на рис. 34, и читателю предлагается отследить её шаг за шагом.

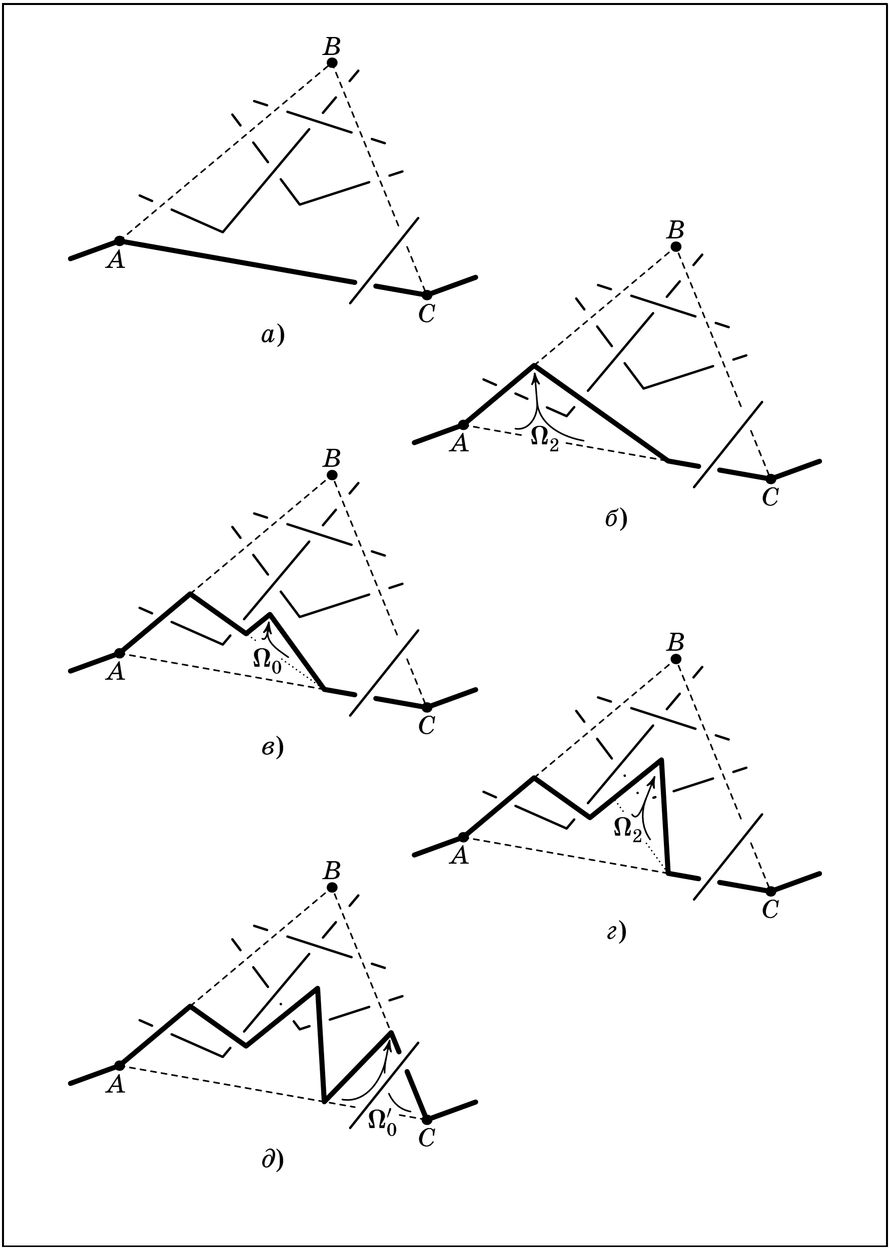
Конечно, проверка на одном частном примере никоим образом не является доказательством. Приводить настоящее доказательство, однако, мы не будем — идейно оно не отличается от приведённого примера, но его строгое изложение скучно и длинно.

* * *

Таким образом, лемма Рейдемейстера сводит задачу распутывания узла к выполнению простеньких операций над кривой на плоскости. На первый взгляд кажется, что, более того, эта лемма даёт быстрый и эффективный алгоритм для распутывания. Действительно, заметив, что среди операций Рейдемейстера две у м е н ь ш а ю т ч и с л о п е р е к р ё с т к о в диаграммы узла (а именно, убирание петель Ω_1 и уничтожение парных перекрёстков Ω_2), при распутывании будем применять только эти «упрощающие» операции. Такой алгоритм (стремящийся на каждом шагу упростить узел, т. е. уменьшить число перекрёстков) называется *жадным*: он «жаден» до упрощения, он не думает вперёд и на каждом шагу делает ход, добиваясь сиюминутного упрощения.

Вернёмся к рис. 33, *б* и посмотрим, как бы повёл себя жадный алгоритм в этой ситуации. Ясно, что он не стал бы применять операцию Ω_3 (как это сделали мы при переходе *б* $\to$ *в*), а применил бы операцию Ω_2, убрав одним махом два перекрёстка! Читателю мы предлагаем нарисовать окружность на рис. 33, *б*, охватывающую парный перекрёсток, применить операцию Ω_2, а затем, выступая в роли жадного алгоритма, довести распутывание до конца. Ясно, что для выполнения этих указаний человека вполне может заменить компьютер.

И что же, мы нашли простой способ распутывания узлов? Увы. Жадный алгоритм не всегда умеет распутывать тривиальные узлы. В самом деле, посмотрите на узел на рис. 35, *а*. Ясно, что перед ним жадный алгоритм бессилен: нельзя произвести ни одной упрощающей операции Рейдемейстера. А между тем читатель, поглядев немного на рис. 35, без труда убедится, что узел-то — тривиальный! (Для этого достаточно перекинуть верхний «пузырь» над верхней


B
A
C
а)
B
A
Ω_2
C
б)
B
A
Ω_0
C
в)
B
A
Ω_2
C
г)
B
A
Ω_0'
C
д)

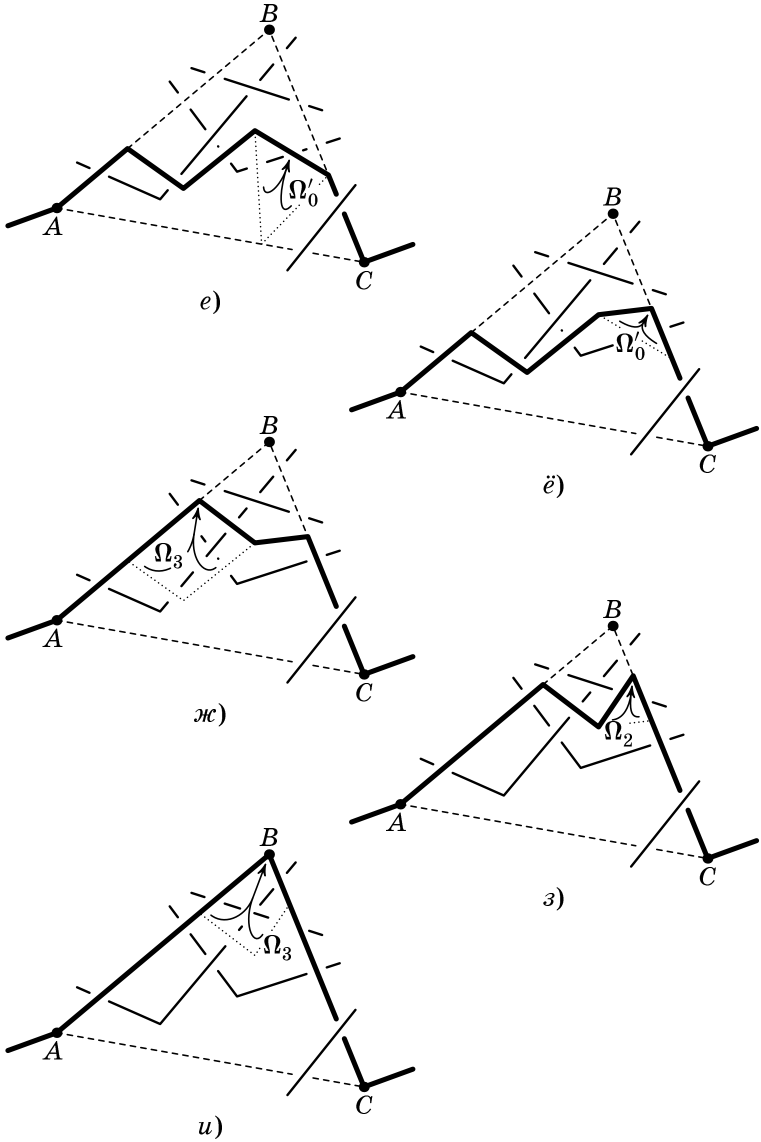
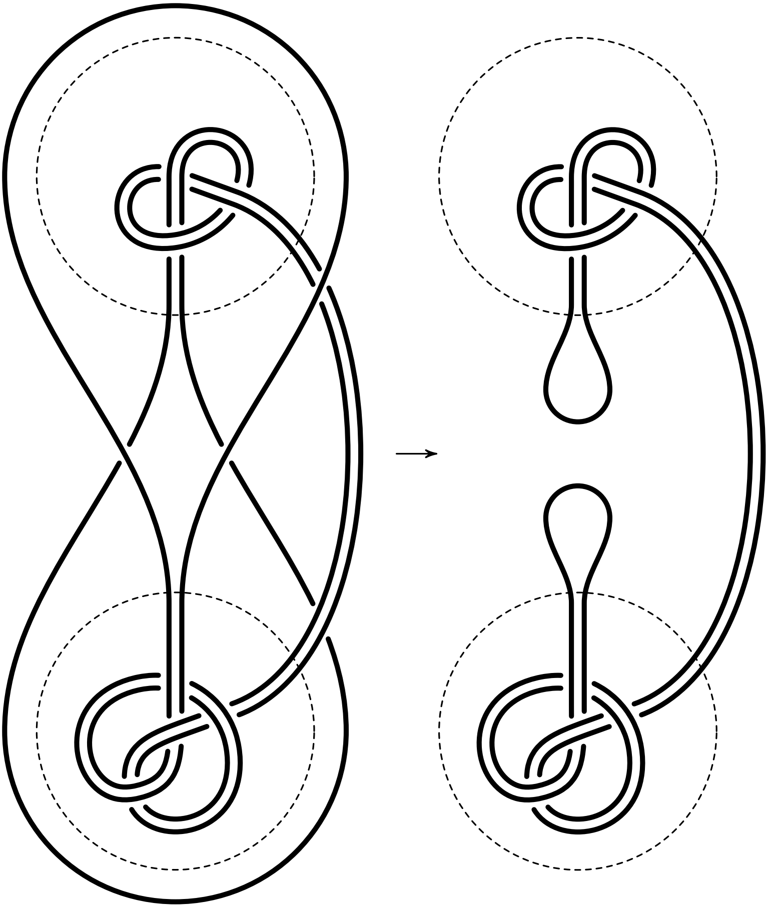

B
A
C
Ω_0'
е)
B
A
C
Ω_0'
ё)
B
A
C
Ω_3
ж)
B
A
C
Ω_2
з)
B
A
C
Ω_3
и)

Рис. 34



а)　　　　*б*)

Рис. 35

пунктирной окружностью, а нижний «пузырь» протащить под нижней пунктирной окружностью, рис. 35, *б*, после чего развязать получившуюся «верёвку» из сдвоенной нити.)

В полном соответствии с леммой Рейдемейстера, узел на рис. 35, *а* можно развязать и с помощью операций Ω_0, Ω'_0, Ω_1, Ω_2, Ω_3, но при этом начинать придётся с операции Ω_2, которая у в е л и ч и в а е т число перекрёстков на 2.

* * *

У этой истории есть мораль, выходящая за рамки теории узлов. А именно, пример развязывания узлов показывает, что жадность (здесь — желание с и ю м и н у т н о упростить или улучшить ситуацию) не всегда приводит к цели. В теории распутывания узлов, как и в жизни, приходится проявлять дальновидность: *прежде чем упрощать, иногда сначала целесообразно ещё более усложнить ситуацию, и только после этого удастся упростить её окончательно.*

* * *

Раз жадный алгоритм не годится, по какому же алгоритму мой компьютер распутывает узел? Очень просто — по алгоритму *полного перебора с запоминанием*. Действует он так. Диаграмма узла рисуется на экране мышкой, а затем кодируется самим компьютером в виде строки из чисел (номеров перекрёстков), букв В, Н (обозначающих проход сверху или снизу) и знаков +, – (обозначающих знак ориентации перекрёстков, рис. 36). Например, узел на рис. 33, *а* кодируется так:

$$1(\text{В}+)\ 2(\text{В}-)\ 3(\text{В}-)\ 3(\text{Н}-)\ 4(\text{В}-)\ \ldots\ 1(\text{Н}+)\ \ldots \tag{4}$$

Глядя на диаграмму узла (вернее, на её код), компьютер находит в с е в о з м о ж н ы е конфигурации, к которым применимы операции Ω_1, Ω_2, Ω_3, и выполняет их. Например, он находит петельку 3(В–) 3(Н–) и удаляет её, т. е. из кода (4) выбрасывает кусок 3(В–) 3(Н–). Все полученные коды он запоминает, каждый вновь преобразует всеми возможными способами и т. д. Если данный узел был тривиальным, наш переборный алгоритм рано или поздно (в силу леммы Рейдемейстера) его распутает, т. е. вычеркнет все символы кода, получив, как говорят математики, «пустое слово». Если же

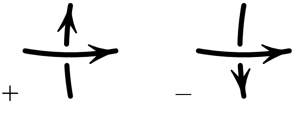


Рис. 36

в компьютер ввести диаграмму нетривиального узла, наш алгоритм будет «работать вечно».

Алгоритм полного перебора страдает очевидным недостатком: он требует огромных ресурсов машинной памяти, ибо сильно ветвится и требует запоминания всех промежуточных ходов. На практике он эффективно работает только тогда, когда не нужно выполнять операцию Ω_2 в сторону увеличения числа перекрёстков. Например, узел на рис. 35, *а* (имеющий всего 34 перекрёстка) ему не по зубам.

Для более эффективного машинного распутывания узлов требуется принципиально другой способ их кодирования.

* * *

Совсем недавно молодой математик Иван Дынников открыл новый, совершенно оригинальный способ кодировать узлы (и получил за свою работу премию Московского математического общества).

Узлы — трёхмерные объекты, и почти все из них не удаётся изобразить на плоскости без перекрещиваний, иначе говоря, их нельзя положить на плоскость. Идея Дынникова состоит в том, чтобы использовать не плоскость, а три полуплоскости, т. е. «трёхстраничную книжку» (рис. 37). Обозначим «страницы» этой «книжки» через P_1, P_2 и P_3, а их общую границу — через l.

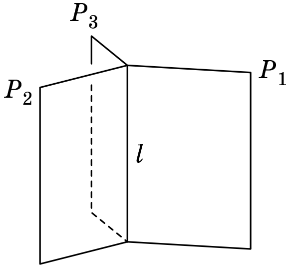


Рис. 37

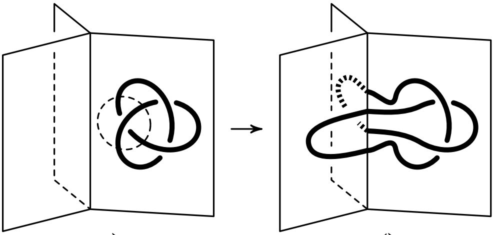
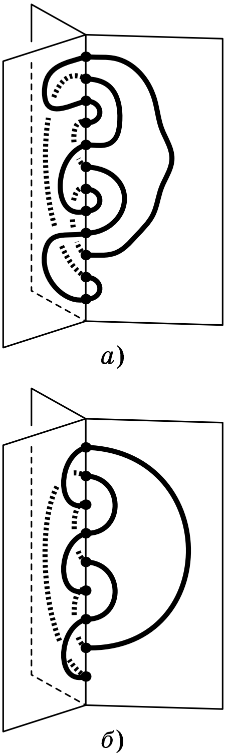


Рис. 38

Лемма. Для любого узла K существует эквивалентный ему узел K', который целиком содержится в объединении полуплоскостей P_1, P_2 и P_3.

Другими словами, любой узел можно уложить на эти три полуплоскости без самопересечений.

Идея доказательства. Сначала нарисуем наш узел на полуплоскости P_1, как мы раньше рисовали узлы на плоскости (рис. 38, *а*). Возьмём какой-нибудь перекрёсток, подвинем его к прямой l, «верхнюю» нить пустим в полуплоскость P_2, а «нижнюю» — в P_3 (рис. 38, *б*). Так поступим со всеми точками скрещивания. В итоге на полуплоскостях не останется двойных точек (рис. 39, *а*), а после стягивания лишних дуг на полуплоскости P_1 (не охватывающих точки пересечения с l) и «скругления» других дуг, узел будет состоять из полуокружностей, лежащих в полуплоскостях P_1, P_2 и P_3, с концами в точках прямой l (рис. 39, *б*).



а)

б)

Рис. 39

Нам достаточно знать, как ведёт себя узел *вблизи* точек пересечения с прямой l (тогда, продолжая эти полуокружности, можно восстановить всю картинку). А этих точек конечное число, причём они бывают только 12-ти типов. Действительно, в каждой точке есть следующие возможности. Во-первых, можно тремя способами выбрать из трёх полуплоскостей две, по которым пройдёт нить. Во-вторых, по каждой полуплоскости нить может пойти либо вверх, либо вниз — это четыре возможности. Пусть буква a означает, что нить проходит по P_1 и P_2, тогда можно договориться, что символы a_1, a_2, a_3, a_4 приписаны типам точек так, как показано на рис. 40; буква b пусть означает полуплоскости P_1 и P_3, буква c — P_2 и P_3. Теперь с помощью этих 12-ти букв можно закодировать любой узел, уложенный на «книжке»: надо выписать по порядку (сверху вниз) все буквы, соответствующие точкам пересечения узла и прямой l. Заметим, что раньше, когда мы кодировали косы, с увеличением числа нитей увеличивалось и число символов, необходимых для кодирования. При описанном способе кодирования узлов нужно всего 12 символов, какой бы сложный узел мы ни взяли.

Например, для узла, изображённого на рис. 39, *б* (трилистника), получаем такую последовательность:

$$a_4\ b_4\ c_3\ a_3\ b_3\ c_3\ a_3\ b_1\ c_1.$$

Так же, как и в группе кос, между этими последовательностями возникают соотношения, из которых можно получить все равенства в системе узлов. Эти соотношения позволяют ответить с помощью компьютера на вопрос: является ли данный узел тривиальным (можно ли его распутать)? Правда, это не теорема, а экспе-

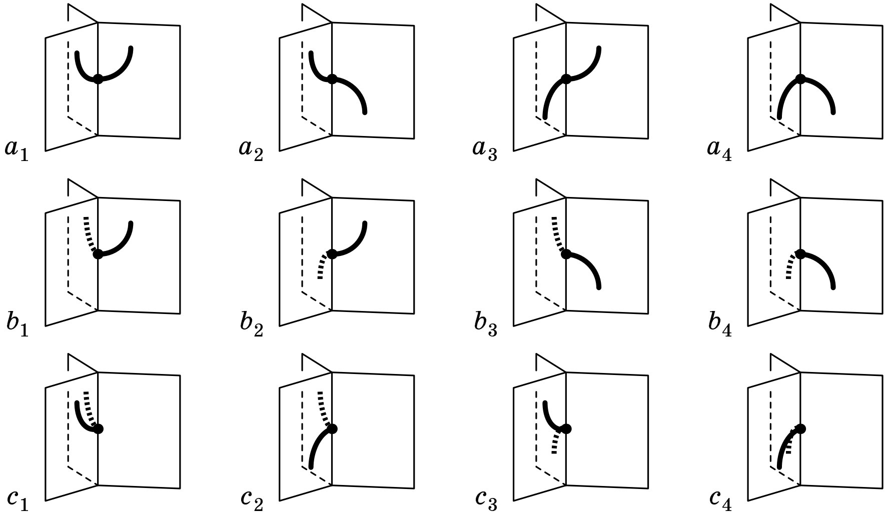


Рис. 40

риментальный факт: имеется программа, написанная Дынниковым, которая очень быстро распутывает чрезвычайно сложные узлы. Например, на обычном персональном компьютере эта программа менее чем за секунду распутывает не только узел на рис. 35, *а* (в нём — 34 перекрёстка), но и тривиальные узлы с несколькими сотнями перекрёстков (с которыми человеку уже не справиться). Если же на вход программы подать нетривиальный узел, компьютер будет «работать бесконечно долго», честно и тщетно пытаясь выполнить невозможное задание (также ведёт себя в таких случаях и программа в моём ноутбуке). Чтобы установить нетривиальность узла, нужны *инварианты* — но это уже совсем другая история.