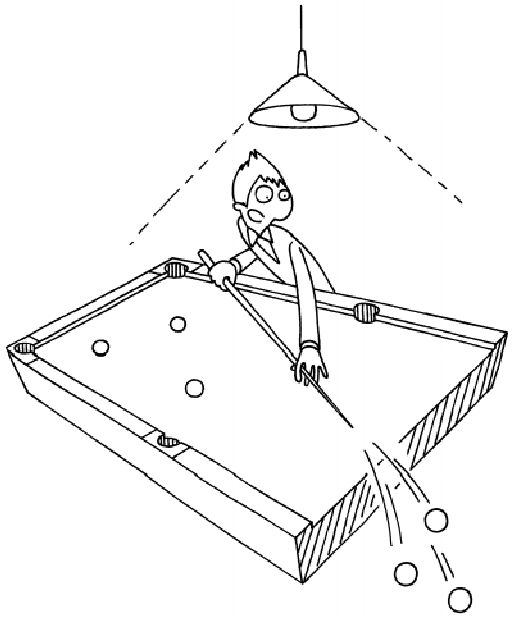
Посов Илья Александрович

РАЗБОР ЗАДАЧ «МАТЕМАТИЧЕСКИЙ БИЛЬЯРД» И «БИЛЬЯРДНЫЙ КОМПЬЮТЕР» КОНКУРСА КИО–2010

Третья задача конкурса КИО–2010 предлагала участникам произвести вычисления с помощью необычного компьютера – бильярдного стола. Несмотря на то, что бильярдный стол теоретически способен производить любые вычисления, а вычислить требовалось очень простую функцию: определить, четное или нечетное количество шариков расположено на столе, – на практике использовать стол в качестве компьютера оказалось нелегко, и получить полное решение задачи удалось только небольшому числу участников. Основная борьба участников развернулась вокруг того, чей компьютер будет совершать меньше ошибок при вычислениях.



Условия задач первого и второго уровня различались. В частности, в задаче второго уровня участникам позволялось ставить на стол дополнительные шары. По этой причине различались и названия задач: «математический бильярд» для первого уровня и «бильярдный компьютер» – для второго.

В статье мы приведем условие задачи, результаты участников обоих уровней, обсудим подходы к решению задачи и разберемся, откуда взялась идея использовать бильярд для вычислений.

УСЛОВИЕ

Приведем условие задачи «математический бильярд», а условие задачи «бильярдный компьютер» обсудим позже.

Участнику дан прямоугольный бильярдный стол. Стол не имеет стен, и катающиеся по нему шарики упадут, если выкатятся за границы. У левой границы стола находятся четыре лунки, в которых могут появляться шарики. Правую границу стола мы тоже будем считать лункой, то есть мы будем следить за шариками, которые упали, докатившись до нее (см. рис. 1). *Участник может расставлять на столе стенки, от которых будут отражаться катающиеся шарики, и изменять*

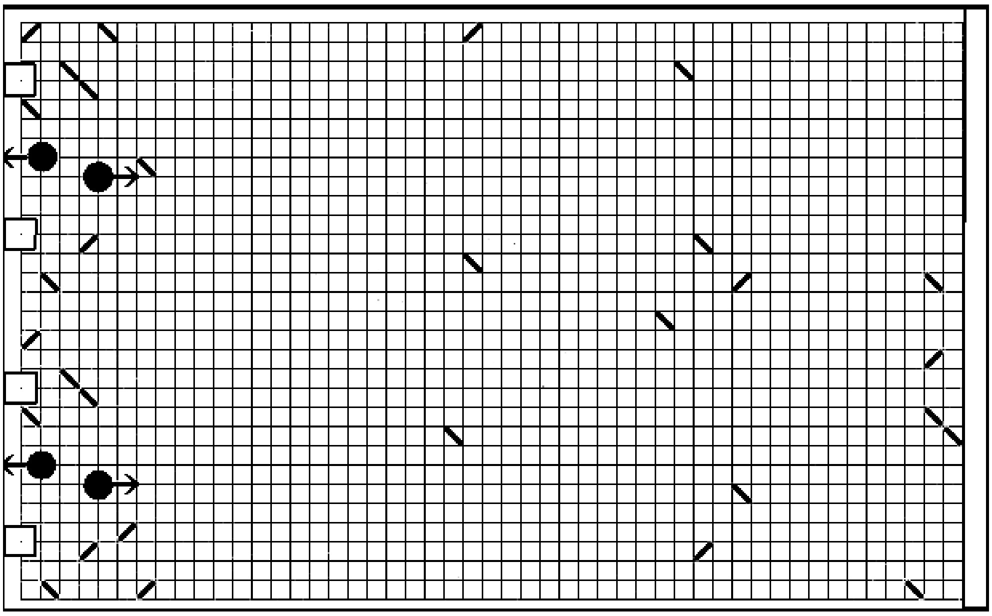


Рис. 1. Бильярдный стол с установленными стенками

расположение луз у левой границы. На рисунке стенки выглядят как небольшие диагональные отрезки.

Участник знает, что из левых лунок одновременно и с одинаковой скоростью выкатится несколько шариков. Стол должен «вычислить», верно ли, что выкатившихся из лунок шариков четное число. Вычисление происходит следующим образом: шарики катятся по столу, отражаются от стенок и при столкновениях отражаются друг от друга. Если после того как все шарики упадут со стола (или пройдет 1000 единиц времени) окажется, что через правую стенку упал хотя бы один шарик, считается, что результат вычисления – «количество нечетно», если через правую стену не упало ни одного шарика, то результат вычисления – «количество четно». Участник должен расставить стенки так, чтобы стол производил вычисление правильно.

Для пояснения представим стол, на котором не установлено ни одной стенки. Любые запущенные шарики обязательно докатятся до правой границы, поэтому можно считать, что такой стол в качестве результата вычисления всегда говорит «количество нечетно». Точнее, почти всегда. В случае, если из лунок не выкатилось ни одного шарика, докатываться до правой стенки будет нечему, и стол даст ответ «четно». Заметим, что в ситуации для нуля шариков стол дает правильный ответ, потому что нуль – четное число. Заметим также, что любой стол дает правильный ответ в ситуации с нулем шариков.

Итак, стол без стенок не всегда дает верные ответы, например, если выкатилось два шарика, его ответом будет «количество нечетно», поэтому такой стол не дает полного решения задачи. Но при подведении итогов учитывались также неполные решения участников. Для каждого стола можно посчитать, сколько ошибок он совершает, и из двух участников решение будет лучше у того, у кого ошибок меньше. Посчитаем, сколько ошибок совершает стол без установленных стенок. Всего существует 16 вариантов вылета шариков из лунок (лунок всего 4, и из каждой шарик либо вылетает, либо нет). Из этих вариантов есть 8 вариантов с четным количеством шариков и 8 – с нечетным. На все 8 четных способов стол будет давать неправильный ответ «нечет», кроме одной ситуации, когда вылетало 0 шаров, следовательно, стол без стенок совершает 7 ошибок.

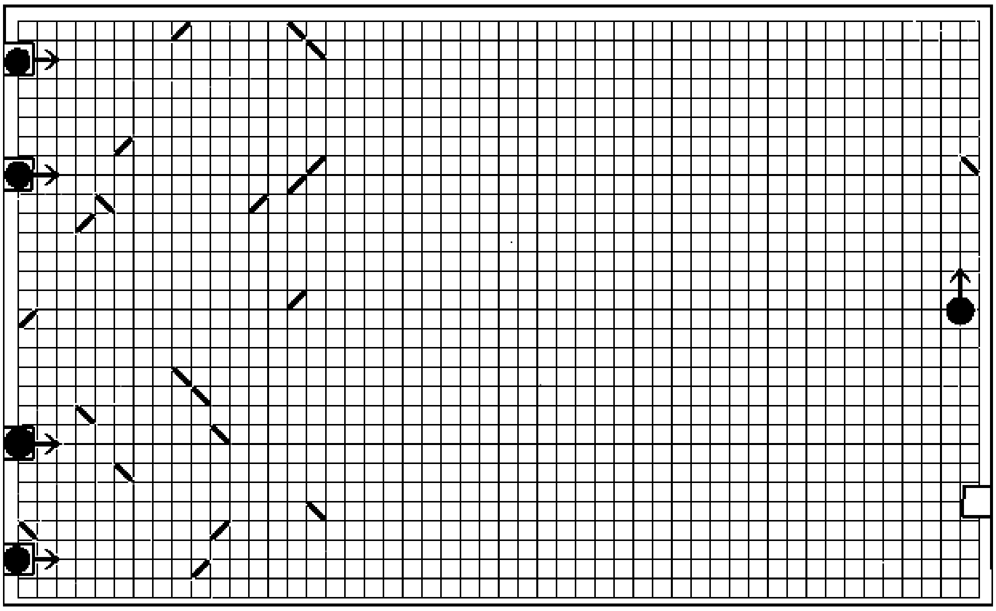
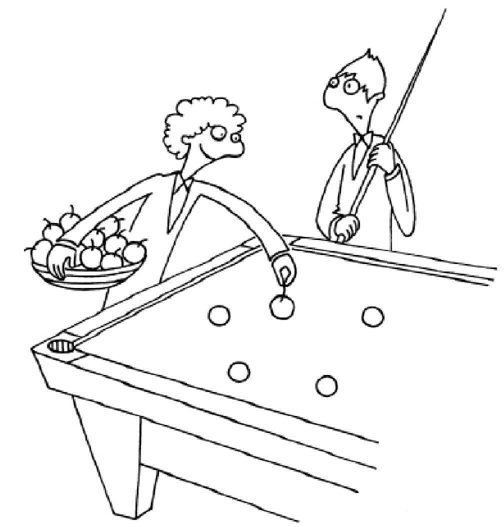


Рис. 2. Бильярдный стол для участников второго уровня

Для участников, в решениях которых количество ошибок совпадало, лучшим решением считалось то, которое использовало меньше установленных стенок. Если же у участников совпадало и количество ошибок, и количество стенок, выбиралось то решение, у которого меньше среднее время катания шариков по столу. (Опустим подробности, как это время вычислялось).

Основное отличие задачи второго уровня «Бильярдный компьютер» состоит в том, что участникам разрешается добавлять на стол собственные шары. Шар добавляется в один из узлов сетки, и ему назначается одно из четырех направлений. Дополнительный шарик, в отличие от исходных, всегда находится на столе, он должен помогать вычислять четность количества исходных шариков и при этом не влияет на нее, то есть, если из лунок выкатилось четыре шарика, их количество считается четным, независимо от того, сколько на столе находится дополнительных шаров. Если у двух участников столы совершают одинаковое количество ошибок при вычислениях, то лучшим считается то решение, которое использует меньшее количество стенок в сумме с количеством дополнительных шаров.

Кроме этого, в задаче второго уровня лункой, в которую должен скатиться шар с ответом, является не вся правая стенка, а обычная лунка, такая же, как та, из которой шарики выкатываются. Последнее отличие задачи второго уровня: считается, что стол дает ответ «четно» в случае, если в лунку попал хотя бы один шарик, и «нечетно», если ни один шарик не попал. Пример стола в задаче второго уровня изображен на рис. 2, шарик, находящийся рядом с правой границей, является дополнительным и установлен участником самостоятельно.



ФИЗИЧЕСКАЯ МОДЕЛЬ БИЛЬЯРДНОГО СТОЛА

Рассмотрим более подробно бильярдные шарики и то, как они ударяются друг о друга и о стенки. Основное упрощение физической модели бильярда в задаче состоит в том, что все ударения шариков считаются абсолютно упругими, а сила трения при движении шариков отсутствует. Это позволяет считать, что скорости шариков неизменны во время движения. Помимо этого, чтобы упростить управление шариками, считается, что кататься они могут только по линиям сетки, горизонтально или вертикально, и узлы сетки шарики проходят одновременно. В начальный момент времени все шарики находятся в узлах сетки, через единицу времени они опять попадают в узлы, и так далее, то есть невозможна ситуация, когда один шарик находится в узле, а другой – нет.

Перечислим другие особенности бильярдного стола:

Расстановка стенок

Все расставляемые стенки ставятся только под углом 45$^{\text{O}}$ к линиям сетки. Как известно, при упругом ударе угол падения равен углу отражения, и только при таком угле после удара шарик будет изменять горизонтальное движение на вертикальное или наоборот. В задаче стенки ставятся как диагонали внутри квадратов сетки. На рис. 3 изображено движение шарика, сталкивающегося со стенкой, он прилетает по одной из стрелок, улетает по другой. Это основной вариант столкновения шариков. Для того чтобы при таких столкновениях центры шариков находились в узлах сетки, их радиус взят равным половине диагонали квадрата сетки.

Столкновение двух объектов

Столкновение шариков друг с другом возможно только так, как это изображено на рис. 3, то есть «по диагонали». Шарик прилетает по одной из нарисованных стрелок, улетает по другой. Все другие способы недопустимы, в случае неправильного столкновения считается, что во время вычисления произошла ошибка, и шарики прекращают кататься по столу. При подведении результатов эта ситуация воспринимается аналогично тому, что стол дал неправильный ответ (например, выдал ответ «нечетное количество», когда количество шаров было четным). Заметим, что лобовое столкновение шариков недопустимо. Действительно, чтобы шарики после удара продолжали синхронно проходить узлы сетки, их радиус должен быть равен половине стороны квадрата, но, как говорилось выше, их радиус выбран равным половине диагонали квадрата.

Столкновение нескольких объектов

Любые одновременные столкновения более чем двух объектов – шариков и стенок – воспринимаются программой как ошибка. Это ограничение программы связано с несовершенством физической модели, которая не позволяет правдоподобно рассчитать движение нескольких шаров после столкновения.

Посмотрим на рис. 4, который изображает одновременное столкновение че-

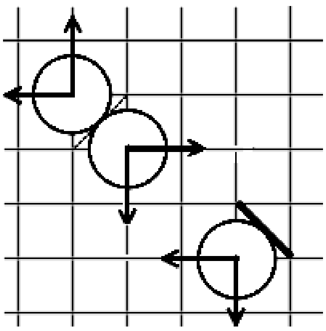


Рис. 3. Столкновение двух шаров и шара со стенкой

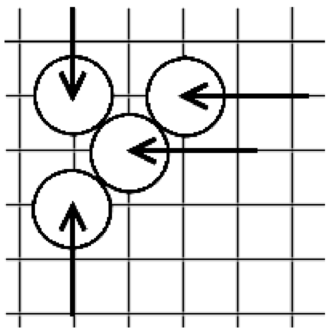


Рис. 4. Соударение с неочевидным результатом

тырех шаров. Эта картинка ставит ряд вопросов: куда будет двигаться средний шар после столкновения? Как объяснить, что шарик будет иметь именно это направление? Не окажется ли, что после столкновения шары будут двигаться не по линиям сетки или изменят скорость (действительно, может быть, средний шар остановится, а три других ускорятся)? Как сформулировать общее правило и алгоритм, который описывает поведение шаров после столкновения? Чтобы упросить условие задачи, было решено отказаться от обработки подобных ситуаций.

Удар в угол стенки

Единственная ситуация, в которой взаимодействие шаров и стенок в задаче физически неестественно, это удар шаром в торец стенки. В этом случае он отскакивает от стены и изменяет направление на противоположное. Шарик отскакивает, как будто у него нулевой радиус, то есть центр шарика полностью достигает угла стены. Такое поведение могло бы быть запрещено, но оно оставлено и немного упрощает задачу участников. Им не придется избегать подобных ударов, которые встречаются достаточно часто.

РЕЗУЛЬТАТЫ УЧАСТНИКОВ

Результаты участников первого уровня сведены в таблицу:

Ошибок	Кол-во участников	Стенок в лучшем решении	Стенок в других решениях
0	7	30	34–57
1	1	35	
2	1	32	
3	5	17	24–26
4	17	10	12–42
5	1	12	
6	19	5	6–26

При обсуждении условия задачи мы заметили, что 7 ошибок можно получить, вообще не устанавливая на столе стены. Поэтому результаты, в которых получено 7 ошибок и больше, в таблице не приведены.

Лучшее решение получено Алексеем Бердовским из Новороссийска, оно было приведено на рис. 1, цветной вариант можно посмотреть на обложке.

Результаты второго уровня:

Ошибок	Кол-во участников	Построений в лучшем решении	Построений в других решениях
0	1	22	
2	3	22	39–42
3	2	8	22
4	1	16	
5	7	6	10–47
6	15	6	7–35

Лучшее решение второго уровня получил Дмитрий Людва из Москвы, оно было приведено на рис. 2, цветной вариант можно посмотреть на обложке.

ОБРАТИМЫЕ ВЫЧИСЛЕНИЯ

Возможности вычислительной техники растут с огромной скоростью, Хорошее описание этого роста дает закон, известный как закон Мура. Гордон Мур, один из основателей компании Intel, лидер в производстве микропроцессоров, утверждает, что количество транзисторов, которые могут быть помещены в интегральную схему, удваивается каждые два года. Этот закон успешно действует уже более 40 лет и по сегодняшний день, но в ближайшее время, по утверждению того же Мура, закон действовать перестанет. Основная причина – приближение технологий к теоретическому максимуму: транзисторы уже состоят всего из нескольких атомов, а сделать транзистор меньше атома при использовании известных технологий невозможно. Другим естественным ограничением на скорость вычислений является скорость передачи информации, которая не может превышать скорость света.

Но для скорости роста возможностей вычислительной техники есть и другие менее очевидные ограничения, одним из них является принцип Ландауэра. Согласно ему, любое вычислительное устройство, независимо от способа физической реализации, при потере одного бита информации выделяет определенное количество теплоты. Потери битов при традиционных вычислениях происходят постоянно. Например, рассмотрим логическую операцию И. До вычисления мы имеем два бита, после – один. Если после вычисления мы имеем бит 0, мы не можем однозначно определить, какие биты были перед этим, возможно, два нуля, возможно, ноль и единица. Другими словами, информация после вычисления потеряна, энтропия (беспорядок) системы увеличилась, тепло выделилось. Аналогично операции И, вычисление любой бинарной логической операции требует выделения теплоты, если, конечно, считать, что после вычисления исходные данные становятся недоступны. При записи данных в память, предыдущая информация в памяти стирается, что также способствует выделению теплоты в соответствии с принципом Ландауэра.

Для скорости вычислений, на которые способен процессор современного персонального компьютера, принцип Ландауэра не накладывает серьезных ограничений. Тепло, предсказываемое принципом Ландауэра для такого процессора, незначительно и намного меньше того тепла, которое процессор выделяет в реальности. Но для практических задач необходимы значительно более быстрые вычисления. Для прогнозирования глобального климата, землетрясений, моделирования молекулярных процессов, ядерных реакций необходима скорость вычислений, соизмеримая с одним зеттафлопсом. Продемонстрируем, что это за величина. Один флопс – это одна операция с вещественными числами в секунду. Килафлопсы, мегафлопсы, гигафлопсы – соответственно, тысяча, миллион, миллиард операций в секунду. Производительность современного процессора для персональных компьютеров как раз составляет около одного гигафлопса, если вспомнить их заявляемую частоту в один гигагерц. Далее, 1000 гигафлопс – это терафлопс, 1000 терафлопс это петафлопс. Самым производительным на данный момент суперкомпьютером является китайский компьютер Тяньхэ-1А, его производительность составляет около 2.5 петафлопс. После петафлопсов идут экзафлопсы и, наконец, зеттафлопсы. Энергопотребление в этом случае, в соответствии с принципом Ландауэра, будет уже значительным, и, кроме поиска источника питания, разработчикам придется решать проблему отведения тепла от вычислительного устройства. С проблемой отведения тепла мы знакомы даже на персональных компьютерах, несложно обратить внимание, что размеры вентилятора и радиатора для охлаждения процессора занимают значительную часть пространства в корпусе компьютера.

Решением проблемы потребления и выделения энергии может быть только использование нетрадиционных вычислительных устройств, которые не теряют информацию в процессе работы. Удивительно, но с теоретической точки зрения для проведения вычислений использование энергии необязательно. Пример, демонстрирующий это, нам теперь известен. Бильярдный стол способен вычислять, не затрачивая энергию, правда, мы предполагаем, что сила трения отсутствует, а удары шаров и стенок абсолютно упругие. Заметим также, что бильярдный стол не теряет информации в процессе вычислений. Действительно, в каждый момент времени мы можем остановить шары, находящиеся на столе, запустить их в противоположном направлении с той же скоростью, с которой они катились до этого, и шары будут двигаться по тому же маршруту, по которому они катились ранее. Другими словами, мы можем восстановить любое предыдущее состояние стола. Конечно, вряд ли в качестве основы для будущей реализации обратимых вычислительных устройств будет использован именно бильярдный стол. Кроме проблем с избавлением от трения понадобится идеально откалибровать все элементы стола, чтобы шары при движении не отклонялись от необходимых траекторий. Кроме того стол придется сделать бесконечным, чтобы шарики не падали на пол. Возможно, проблемы решатся, если минимизировать стол до такой степени, что он из классического объекта превратиться в квантовый.

Рассмотрим обратимое физическое устройство, которое реализует некоторый логический вентиль, то есть получает на вход несколько нулей и единиц и в качестве результата также выдает несколько нулей и единиц. Очевидно, что если устройство обратимо, то и логический вентиль оказывается обратимым: по результатам вычисления можно однозначно определить, какие данные пришли на вход. Логический вентиль И не является обратимым, потому что, как мы видели, по выходу 0 невозможно определить, какой был вход. Аналогично необратимым является вентиль ИЛИ и любой другой логический вентиль, реализующий бинарную логическую операцию. Из простых операций обратимой пока является только операция НЕ. Получается, что классический способ составлять логические схемы из вентилей И, ИЛИ, НЕ не совместим с обратимыми вычислениями. Разберемся теперь, как устроены обратимые логические вентили, и можно ли из них составить произвольную логическую схему.

Очевидно, что обратимый логический вентиль должен иметь хотя бы столько же выходов, сколько у него входов. Иначе количество комбинаций нулей и единиц на выходе будет меньше, чем количество комбинаций на входе. Можно ли добавить к вентилю И еще один выход, чтобы он стал обратимым? Легко проверить, что одного дополнительного выхода недостаточно, нужны как минимум два. Проще всего считать, что обратимый вентиль И имеет три выхода: первый является результатом вычисления, второй и третий повторяют входные данные. Для обратимых вентилей большое количество «лишних» выходов является нормальным, иногда ими удается воспользоваться, иногда они остаются неиспользованными, и в этом случае называются «мусорными». Надо заметить, что, хоть название «мусорные биты» и является общепринятым, на самом деле такие биты не такие уж и мусорные, они необходимы, чтобы после получения результата запустить вычисления в обратном порядке и вернуть вычислительную систему в исходное состояние. Тогда ею можно будет воспользоваться для новых вычислений.

Придуманный нами только что обратимый вентиль И вместе с аналогичным вентилем ИЛИ и вентилем НЕ может использоваться для составления обратимых логических схем. Это показывает, что любое классическое вычисление можно произвести и сделать его обратимым. Тем не менее, традиционно для описания обратимых логических схем используются другие вентили – вентиль Фредкина и вентиль Тоффоли.

- Вентиль Фредкина имеет три входа и три выхода. Первый вход и первый выход называются управляющими, биты на управляющем входе и выходе совпадают. Данные на вторых двух входах не изменяются на выходе, если управление равно нулю, или меняются местами, если управление равно единице: $(0, x, y) \to (0, x, y)$, $(1, x, y) \to (1, y, x)$.
- Вентиль Тоффоли имеет три входа и три выхода. Первые два выхода повторяют первые два входа, третий выход изменяется в том и только том случае, если первые два входа являются единицами: $(1, 1, 0) \to (1, 1, 1)$, $(1, 1, 1) \to (1, 1, 0)$, иначе $(x, y, z) \to (x, y, z)$.

Считается, что каждый из этих вентилей более естественно реализуется физически, и, кроме того, через них выражаются стандартные логические операции: для примера выразим И с помощью вентиля Тоффоли: $(x, y, 0) \to (?, ?, x \text{ И } y)$. Здесь ? обозначает мусорный бит, значение которого нас не интересует. Попробуйте выразить И и НЕ через вентиль Фредкина. Несмотря на то, что операция ИЛИ не выражается с помощью одного вентиля Фредкина или Тоффоли (проверьте это), ее можно получить с помощью нескольких вентилей: a ИЛИ b = НЕ ((НЕ *a*) И (НЕ *b*)). Таким образом, строить обратимые вычисления можно только с помощью вентилей Фредкина или вентилей Тоффоли.

В заключение краткого обзора обратимых вычислений вспомним о Машине Тьюринга – абстрактном устройстве, которое способно производить произвольные вычисления. В общем случае Машина Тьюринга представляет собой необратимое устройство, она перезаписывает символы на ленте, и после перезаписи не всегда возможно восстановить, какой символ на ленте был. Определим обратимую Машину Тьюринга как машину, у которой по текущей конфигурации (состоянию, содержимому ленты, положению головки) можно однозначно определить предыдущую конфигурацию. Оказывается, для любой Машины Тьюринга можно построить обратимую и эквивалентную ей, то есть производящую те же самые вычисления. Другими словами, обратимые Машины Тьюринга также способны делать произвольные вычисления. Попробуйте это доказать.

ВЫЧИСЛЕНИЯ НА БИЛЬЯРДНОМ СТОЛЕ

Перейдем к физической реализации обратимых вычислений с помощью бильярдного стола. Модель таких вычислений была предложена уже известными нам Фредкиным и Тоффоли.

Дальнейшие обсуждения вычислительных возможностей стола не являются в строгом смысле разбором задачи. Мы не будем рассказывать, как расставить стены, чтобы получить решение для подсчета четности количества шариков. Приведенная информация только демонстрирует возможные рассуждения, которые помогают получить решение.

Покажем, как с помощью стенок на бильярдном столе реализовать вентили И и НЕ. К сожалению, для подробного доказательства того, что стол может реализовать произвольную схему, построить эти два базовых вентиля будет недостаточно. Проблемы возникают в тот момент, когда мы хотим расположить вентили на столе так, чтобы они начали правильно друг с другом взаимодействовать. Необходимо сделать так, чтобы шарики вовремя прикатывались к ним на входы и чтобы вентили не преграждали маршруты случайно катящимся шарикам. Думаю, что участники сами сталкивались с похожими проблемами. При расстановке стенок часто возникает ситуация, что поставленная стенка мешает какому-то шарику, который случайно катится по проходящему через нее маршруту. Мы будем касаться этих и некоторых других проблем в процессе обсуждения, но полного способа, как избежать их, не предоставим. Надо сказать, что многие авторы, обсуждающие модель бильярдного стола, обходят вопросы физической реализации схемы из нескольких вентилей, и часто предлагаемые ими вентили на практике оказываются несоединяемыми друг с другом.

Перейдем к реализации. Будем считать сигналом 1 шарик, который в определенный момент времени катится по определенному маршруту, сигналом 0 – отсутствие шарика на маршруте в необходимый момент.

Реализация вентиля И тривиальна, ее идея в том, что после столкновения двух шариков, они оба изменяют маршрут. Если столкновения не было, шарики продолжают катиться, как раньше, поэтому для реализации И мы должны ждать один из шариков на его измененном маршруте. Он прилетит туда, только если произошло

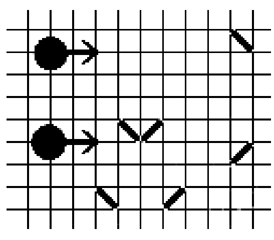


Рис. 5. Вентиль И

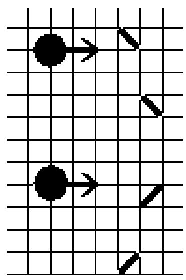


Рис. 6. Вентиль И (другой вариант)

столкновение. На рис. 5 и 6 изображены две реализации вентиля И: шарики с направлениями указывают, где должны располагаться исходные шары, если они есть. Проверьте, что во всех четырех вариантах, есть шарик на входе или его нет, вентиль правильно вычисляет операцию И.

Оба вентиля основаны на описанной выше идее о столкновении, но устроены так, чтобы шарики прилетали в вентиль слева, результат вылетал бы из вентиля тоже направо, а мусорные шарики улетали бы обратно налево. Такое внимание к мусорным шарикам необходимо для того, чтобы они не мешали нормальным шарикам. Не будем подробно останавливаться на том, действительно ли мусорные шарики, летящие справа налево, никому не мешают. Возможно, их надо было направлять вверх или вниз, но в начале раздела мы договорились не обсуждать эти вопросы подробно.

Перейдем к реализации вентиля НЕ. На выходе вентиля шарик есть в том и только том случае, если шарика не было на входе. Очевидно, этот вентиль не реализуется без дополнительных шаров. Если шарика на входе не было, ему неоткуда появиться на выходе. В этом смысле участники первого уровня реализовать вентиль НЕ не могут. А для участников второго уровня идея реализации опять же тривиальна. Запустим дополнительный шарик по определенному маршруту и будем ждать его в конце. Если на вход вентиля подан еще один шарик, он собьет наш шарик с маршрута и мы его не дождемся. Реализация представлена на рис. 7. Опять же шарики влетают и вылетают из вентиля слева направо, мусор улетает налево. Дополнительный шар в этом вентиле прилетает сверху.

Приведем примеры конструкций, которые нужны для соединения базовых вентилей И и НЕ. На рис. 8 две стены позволяют шарику переместиться с одной прямолинейной траектории на параллельную. На рис. 9 показано, как задержать шарик на прямолинейной траектории. Попробуйте доказать, что подобную задержку движения по прямой можно произвести только на четное число единиц времени. Это ограничение на самом деле не является ограничением, потому что задерживать на нечетное время никогда не нужно. На рис. 10 показано, как можно позволить пересекаться траекториям двух шариков. Даже если они летят оба, они все равно продолжат исходное движение, но, к сожалению, задержавшись, по сравнению с тем, как если бы летел один шарик. На рис. 11 показано, как заставить шарик перейти на параллельную траекторию, но только если сверху прилетает контрольный шарик.

Воспользуемся полученными знаниями, чтобы реализовать операцию ИЛИ. Как уже говорилось,

a ИЛИ b = НЕ ((НЕ a) И (НЕ b)),

эта идея использована на рис. 12. Известные нам уже вентили И, НЕ обведены прямоугольниками. Дополнительные стены на рисунке нужны для управления движением шаров, верхняя правая дополнительная стенка нужна, чтобы уменьшить картинку, позволив дополнительному шарику прилетать не сверху, а сбоку. Две оставшиеся стены отбивают мусорные шарики вбок, чтобы они не летали по вентилям. Отметим еще одну проблему, усложняющую проектирование схем, и доказа-

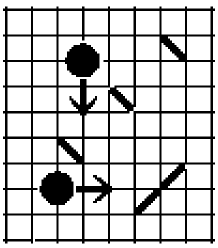


Рис. 7. Вентиль НЕ

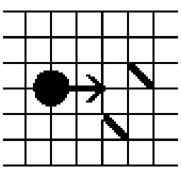


Рис. 8. Сдвиг

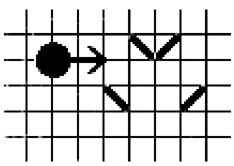


Рис. 9. Задержка

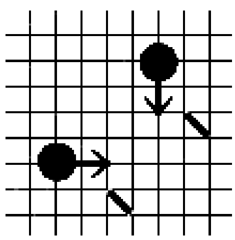


Рис. 10. Пересечение

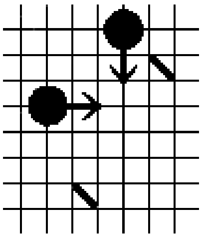


Рис. 11. Управляемый сдвиг

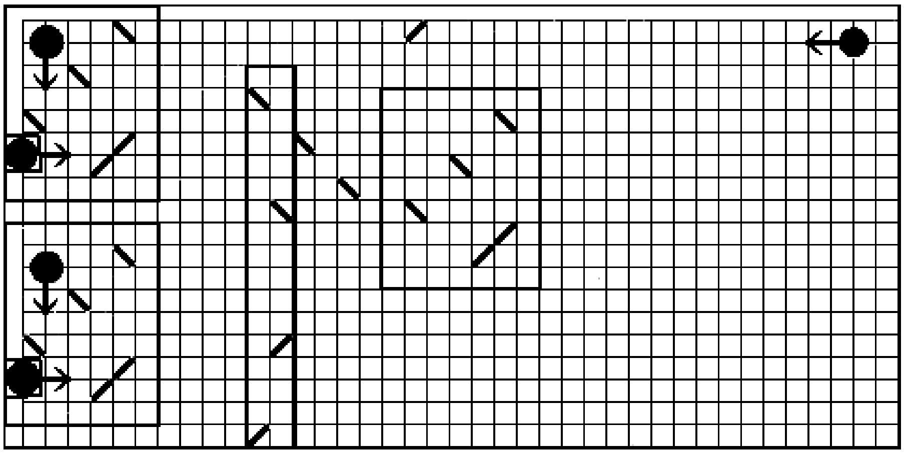


Рис. 12. Реализация вентиля ИЛИ через И и НЕ

тельство того, что стол способен реализовать любую схему. Как видно, дополнительный шар летит очень долго, чтобы в нужный момент попасть в последний вентиль НЕ. Для больших схем дополнительные шары будут летать перед попаданием в вентиль очень долго, для их полета необходимо место, за которым необходимо следить при проектировании.

Заметим, что участники первого уровня не могут реализовать вентиль ИЛИ. В нашем примере мы использовали три дополнительных шара, возможно, существует реализация с одним или двумя дополнительными шарами, но без них это сделать невозможно. Проверим это. На входе дается не более двух шариков, на выходе мы должны получить шарик, летящий по определенному маршруту. Пусть на входе дан первый шарик, но не дан второй, значит, первый шарик сам по себе, отбиваясь от стенок, попадает на этот маршрут. Аналогично, пусть дан второй шарик, но не дан первый. Он тоже попадает на этот маршрут. Поймаем шарик, летящий по маршруту, и запустим обратно. Куда он вернется? С одной стороны туда, где был первый шарик, с другой – туда, где был второй. Противоречие. Кстати, это рассуждение объясняет, почему участники первого уровня имели такую большую лунку размером во всю правую границу.

Еще одна проблема создания произвольной логической схемы состоит в том, что обычно в схеме один результат вычислений используется два раза. Например, операция исключающего или реализуется как (a И НЕ b) ИЛИ (НЕ a И b), но если шарик a уже используется в первом И, как он попадет во второй И. Реализуйте вентиль, удваивающий шар. Если в него не прилетает шар, то и на выходе ничего не вылетает, если прилетает шар, то на выходе вылетают два. Естественно, вам понадобится дополнительный шар.

Попробуйте реализовать вентили Фредкина и Тоффоли, считая, что шарики прилетают слева, а улетают направо. Попробуйте использовать как можно меньше дополнительных шаров. Возможно, вам понадобится поле большего размера, чем предложено в программе.

Посов Илья Александрович,
ассистент кафедры ВМ–2
СПбГЭТУ «ЛЭТИ».