Игры на графах

Андрей Станкевич

Аннотация

Данный обзор посвящен анализу комбинаторных игр на графах. К сожалению, существует очень мало литературы на русском языке, в которой уделено внимание этой интересной и достаточно глубокой теме. В то же время на олимпиадах по информатике часто встречаются задачи, решение которых подразумевает анализ такой игры, напрямую, либо с помощью теории Шпрага–Гранди. Эти темы подробно рассмотрены в обзоре.

Введение

Рассмотрим классическую математическую игру, встречающуюся во многих книгах по «занимательной математике». Пусть на столе лежит кучка из n спичек. Два игрока по очереди берут спички из кучки, разрешается взять одну, две или три спички. Тот, кто берет последнюю спичку, выигрывает. Требуется определить, кто выигрывает при оптимальной игре обоих игроков.

Проанализируем зависимость того, кто выигрывает в описанной игре, от числа n. Пусть на столе лежит одна, две или три спички. Тогда первый игрок может сразу взять все спички и выиграть. Следовательно при $n = 1$, $n = 2$ и $n = 3$ выигрывает первый игрок. Пусть $n = 4$. Тогда сколько бы спичек не взял первый игрок, на столе останется от одной до трех спичек, и второй игрок выиграет, забрав их все. Следовательно при $n = 4$ выигрывает второй игрок. Пусть теперь n равно пяти, шести или семи. Тогда взяв одну, две или три спички, соответственно, первый игрок оставляет второму игроку четыре спички, а это, как мы уже знаем, проигрышная позиция. Следовательно в этих случаях снова выигрывает первый игрок. В случае же восьми спичек в исходной кучке снова выигрывает второй.

Теперь уже несложно сформулировать и доказать по индукции гипотезу: позиции, когда количество спичек на столе кратно четырем являются выигрышными для второго игрока, а остальные позиции — для первого.

Рис. 1. Кто выигрывает в игре со спичками в зависимости от количества спичек в исходной кучке

Фундаментальная идея, использованная при анализе — позиция является выигрышной для того игрока, чей ход, если из нее можно сделать ход в проигрышную позицию, и проигрышной, если все ходы из нее ведут в выигрышные.

Обобщим предыдущую игру. Пусть по прежнему на столе лежит кучка из n спичек, но теперь разрешается брать от 1 до k спичек за ход. Аналогичный анализ позволяет получить следующий критерий: если количество спичек кратно $k+1$, то выигрывает второй игрок, а в противном случае — первый.

Усложним игру. Пусть по прежнему исходно на столе исходно расположена кучка из n спичек, причем первым ходом разрешается взять не более k спичек, а вот каждым следующим — не больше, чем взял противник своим предыдущим ходом. В отличие от предыдущей задачи здесь простого правила, кто выигрывает в начальной позиции, сразу не видно. Математическая задача начинает приобретать «программистский» оттенок. Для определения выигрышности позиции воспользуемся некоторым подобием динамического программирования.

Что характеризует позицию в данной игре? Количество спичек, оставшихся на столе — i, и количество спичек, которые противник взял предыдущим ходом — j. В начальной позиции $i = n$, и удобно считать, что $j = k$. Заведем массив $a[i][j]$, в котором будем хранить, является ли позиция (i, j) выигрышной для первого игрока. Ясно, что все позиции вида $(0, j)$ являются проигрышными. Позиция же (i, j) для $i > 0$ является выигрышной, если найдется такое t от 1 до $\min(i, j)$, такое что $a[i-t][t]$ — проигрышная позиция. Тем самым, массив можно заполнить требуемыми значениями простой программой из трех циклов.

Рассмотрим еще один пример игры. Пусть задано прямоугольное поле, состоящее из $m \times n$ клеток, некоторые клетки свободны, а на некоторых расположены непроходимые препятствия. Исходно на одной из клеток расположен бандит, а на другой — полицейский. Они делают ходы по очереди, своим ходом бандит может переместиться на клетку, имеющую общую сторону с клеткой, на которой он находится, а полицейский — на клетку, имеющую с клеткой, на которой он находится, общую точку (вершину или сторону). Если бандит делает ход за границу поля, то он выигрывает. Если полицейский делает ход на клетку, на которой находится бандит, то выигрывает он. По заданной начальной позиции требуется определить, кто выигрывает при оптимальной игре.

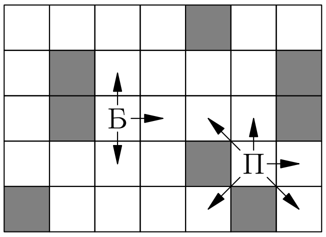


Рис. 2. Пример игрового поля в задаче про бандита и полицейского.

На первый взгляд, связь задачи про бандита и полицейского с задачами про спички не прослеживается. Однако при внимательном рассмотрении у них оказывается много общего. В этой задаче снова появляется понятие «позиции» игры — на этот раз позиция включает в себя положение бандита и полицейского на поле и информацию чей ход, снова возникает вопрос о выигрышности или проигрышности каждой позиции для очередного игрока, да и алгоритм определения, кто выигрывает в данной позиции очень похож.

Эти, а также многие другие задачи, в которых два игрока ведут некоторую игру с полной информацией, делая ходы по очереди, изучает теория *комбинаторных игр*.

Отличие комбинаторных игр от игр, которые обычно изучаются в классической («экономической») теории игр, в том, что в них игроки делают ходы по очереди, а не одновременно (классическая теория игр освещается в огромном количестве книг, в названии которых фигурируют слова «теория игр» или «исследование операций»). Поэтому, в частности, в комбинаторных играх обычно не возникает необходимости в смешанных стратегиях с привлечением случайности, и следовательно результат игры является не случайной величиной, а детерминированной характеристикой игры.

Нас будут интересовать в основном *справедливые* игры, в которых возможные ходы обоих игроков во всех позициях совпадают (следует отметить, что большинство реальных игр, на самом деле, не являются справедливыми по этому определению, например в шахматах игрок может двигать только фигуры своего цвета. Из приведенных примеров справедливой не является игра из задачи про бандита и полицейского. По-английски справедливые игры называют более точным термином impartial games). Кроме того, мы будем считать, что ведется *нормальная* игра, то есть игрок, который не может сделать ход проигрывает (в противоположность игре *в поддавки*, когда тот, кто не может сделать ход, выигрывает).

Примером нормальных справедливых игр могут служить рассмотренные нами в самом начале игры со спичками. Другим (похожим) примером является известная среди любителей занимательной математики игра ним. Пусть на столе расположено несколько кучек камней. За один ход игрок должен выбрать одну из кучек и взять из нее один или несколько камней. Игрок, который не может сделать ход, поскольку камни во всех кучках кончились, проигрывает.

Более сложный пример — игра Хакенбуш, предложенная Конвэем. Задан неориентированный граф, одна из вершин которого выбрана в качестве корня. Игроки по очереди выбирают ребро в графе и «перерезают» его. После этой операции часть графа, не достижимая из корня, удаляется. Игрок, который не может сделать ход, поскольку от графа остался только корень, проигрывает.

Эти игры, как и многие другие, могут быть обобщены следующим образом. Пусть задан ориентированный граф G и начальная вершина

u в нем. Игра протекает следующим образом. Специальная фишка помещается в вершину u. Игроки по очереди делают ходы, ход состоит в перемещении фишки по дуге графа, исходящей из вершины, в которой она находится. Если игрок не может сделать очередной ход, поскольку таких дуг нет, он проигрывает. Вершины, из которых нельзя сделать ход, называют *терминальными*.

Посмотрим, каким образом описанные выше игры можно интерпретировать как игры на графах.

Рассмотрим сначала игры со спичками. В игре, где на каждом ходе можно взять от одной до k спичек позиция характеризуется количеством спичек, оставшемся на столе. Соответственно, вершины графа игры будут числами от 0 до n, вершина 0 является терминальной, для $i > 0$ из вершины i дуги идут в вершины от $\max(0, i-k)$ до $i-1$.

В игре, где разрешается брать не больше спичек, чем взял противник на предыдущем ходе, вершинами будут пары чисел (i, j), описанные в решении задачи, а дуги соответствуют переходам в динамическом программировании.

В игре про бандита и полицейского вершинами графа будут наборы, состоящие из координат бандита, полицейского, а также информации, чей ход (в отличие от остальных игр, как уже упоминалось, просто информации о положении бандита и полицейского недостаточно, потому что возможные позиции, в которые есть ход, разные для игроков).

Вершинами графа, соответствующего игре ним, являются наборы чисел, характеризующие количество камней в кучках, а дуги из набора ведут в наборы, в которых одно из чисел исходного набора заменено на произвольное меньшее.

В игре Хакенбуш вершинами будут все возможные связные подграфы заданного графа, содержащие корень, дуги соответствуют возможным ходам.

Отметим, что графы, соответствующие всем описанным играм, кроме игры в задаче про бандита и полицейского, получились ациклическими: в игре ним после каждого хода уменьшается суммарное количество камней, а в игре Хакенбуш — количество ребер в графе. Можно рассмотреть и другие игры, которым соответствуют графы, не являющиеся ациклическими. Например, если в игре ним разрешить еще один вариант хода — переложить некоторое количество камней из одной кучки в другую, то граф позиций для нее будет содержать циклы.

Основной целью анализа игры является выяснение, кто из игроков выигрывает при оптимальной игре обоих. В первом разделе мы рассмотрим алгоритмы определения победителя для игр на конечных графах в случае, если граф задан в явном виде. Второй раздел развивает теорию прямой суммы игр, которая позволяет во многих случаях эффективно проанализировать игры, заданные в компактной форме. В третьем разделе строится теория Шпрага-Гранди для прямой суммы игр на ациклических графах. В заключительном четвертом разделе рассматривается теория Смита, обобщающая эту технику на случай графов с циклами.

1. Анализ игр на графах

Игрой на графе называется пара $A = \langle G, u \rangle$ из ориентированного графа G и вершины этого графа u. Вершины графа G называют *позициями* игры A, а его дуги — *ходами* в игре A. Если задан граф G и игры $A = \langle G, u \rangle$ и $B = \langle G, v \rangle$ на этом графе, то если в графе присутствует дуга uv, говорят, что из игры A возможен ход в игру B.

Процесс игры протекает следующим образом. Два игрока по очереди делают ходы. В качестве хода игрок заменяет рассматриваемую игру A на одну из игр, в которую из A возможен ход. Если такой игры нет, то игрок, который должен сделать ход, проигрывает.

Поскольку граф, на котором происходит игра, в ее процессе не меняется, а меняется только текущая вершина, то удобно визуализировать процесс игры на графе с помощью перемещения по графу специальной фишки.

Замечание. Отметим терминологический момент. С одной стороны обычно игрой называют собственно процесс перемещения фишки по графу. С другой стороны, формально игрой $A = \langle G, u \rangle$ называется пара из графа и вершины в нем. Обычно эта небольшая несогласованность не вызывает неоднозначности в понимании, поэтому не будем акцентировать на ней внимание, однако следует помнить, что каждой вершине на графе соответствует своя игра с начальной позицией в этой вершине.

Зададимся вопросом — кто из игроков выигрывает в этой игре при оптимальных действиях обоих. Игра A называется *выигрышной*, если вне зависимости от действий второго игрока, игрок, который должен делать ход в A (первый игрок) выигрывает. Игра A называется *проигрышной*, если вне зависимости от действий первого игрока, второй игрок может выиграть. Игра, которая не является ни выигрышной, ни проигрышной, называется *ничейной*.

Заметим, что ничейная игра при оптимальной игре обоих игроков продолжается бесконечно. Чтобы такое было возможно, граф должен содержать циклы. Если же граф игры ациклический, то игра всегда завершается победой одного из игроков. Рассмотрим, для начала, случай ациклического графа.

Пусть задан ациклический граф G. Для каждой вершины этого графа можно однозначно определить, какой из игроков выиграет в игре, начинающейся в этой вершине — тот, кто ходит первым (такие вершины называют *выигрышными*), или тот, кто ходит вторым (такие вершины называют *проигрышными*).

Теорема 1. Все вершины ациклического графа $G = \langle V, E \rangle$ можно разбить на два непересекающихся множества N и P выигрышных и проигрышных вершин[1], соответственно. В играх, которые начинаются в выигрышных вершинах, первый игрок может выиграть вне зависимости от действий второго игрока. В играх, которые начинаются в проигрышных вершинах, второй игрок может выиграть вне зависимости от действий первого игрока.

При этом вершина выигрышная, если из нее есть ход в проигрышную и проигрышная в противном случае, то есть

$$N = \{u \mid \text{существует } uv \in E, \text{ такая что } v \in P\};$$
$$P = \{u \mid \text{если } uv \in E, \text{ то } v \in N\}.$$

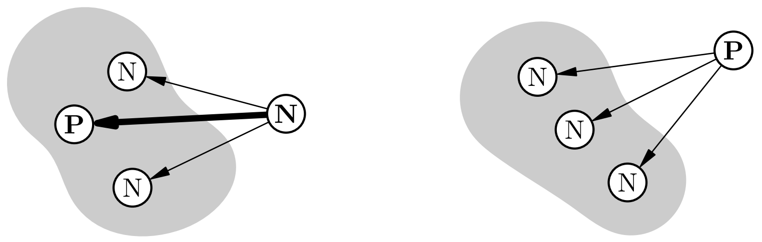


Рис. 3. Если есть ход в проигрышную вершину, то вершина — выигрышная. Если все ходы ведут в выигрышные вершины, то вершина — проигрышная.

▷ Поскольку граф G ациклический, то у него существует *топологическая сортировка* — его вершины можно занумеровать таким образом, что дуги только идут из вершин с меньшими номерами в вершины с большими. Нам будет удобно использовать нумерацию вершин в обратном топологическом порядке, когда дуги идут от вершин с большими номерами в вершины с меньшими.

Докажем утверждение теоремы индукцией по обратной топологической сортировке графа.

[1] N — от английского слова next — следующий, выигрывает тот, кто ходит следующим, P — от английского слова previous — предыдущий, выигрывает тот, кто ходил предыдущим.

База — пустое множество вершин можно разбить искомым образом.

Пусть все вершины с номерами, меньшими номера вершины u, разбиты на два множества указанным образом. Рассмотрим дуги, которые ведут из вершины u. Если хотя бы одна из них ведет в вершину, которая отнесена в множество P, то отнесем u в множество N. В этом случае для выигрыша в игре, соответствующей вершине u первый игрок ходит по дуге в вершину множества P, где по индукционному предположению, тот кто делает ход — теперь это второй игрок — проигрывает вне зависимости от своих действий.

В противном случае — если все дуги из данной вершины ведут в множество N — отнесем вершину u к множеству P. Действительно, какой бы ход не сделал первый игрок, получится игра для вершины из множества N и второй игрок, который будет делать в ней ход, сможет выиграть.

Таким образом мы можем классифицировать все вершины. $\triangleleft$

Разбиение вершин, построенное в доказанной теореме, называется *выигрышно-проигрышным* разбиением вершин графа. Из доказательства автоматически следует, что выигрышно-проигрышное разбиение вершин графа единственно. Доказательство теоремы описывает, в частности, алгоритм построения такого разбиения.

Будем помечать вершины как выигрышные, либо проигрышные. Отсортируем вершины графа в обратном топологическом порядке и просмотрим их по возрастанию номера. При просмотре вершины u перебираем все дуги uv, выходящие из нее, и если одна из них ведет в вершину, помеченную как проигрышную, то помечаем вершину u как выигрышную. В противном случае помечаем ее как проигрышную. Поскольку топологическая сортировка выполняется за $O(E)$, а в дальнейшем каждая дуга графа просматривается ровно один раз, то алгоритм разбиения вершин ациклического графа на выигрышные и проигрышные работает за $O(E)$.

В качестве примера, рассмотрим применение алгоритма к игре на графе, приведенном на рис. 4.

Сначала вершина E как терминальная помечается как проигрышная. Затем вершины C и D помечаются как выигрышные (из них есть ход в E). Затем вершина B помечается как проигрышная (из нее оба хода ведут в выигрышные вершины). Наконец, вершина A помечается как выигрышная (из нее есть ход в проигрышную вершину B).

Отметим, что поскольку алгоритм топологической сортировки с использованием обхода в глубину (см., например, [1]) помечает вершины как раз в порядке обратной топологической сортировки, то удобно «встроить» анализ игры на графе прямо в код обхода в глубину.

А именно, при просмотре ребер, исходящих из вершины u, делаем

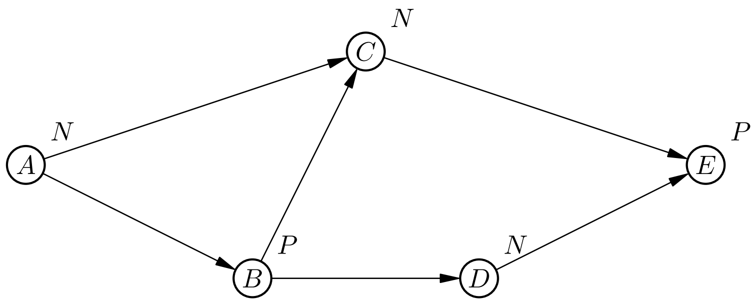


Рис. 4. Пример ациклического графа и выигрышно-проигрышного разбиения его вершин

следующее. Пусть рассматривается ребро uv. После рекурсивного вызова (если вершина v еще не была посещена) или сразу (в противном случае) проверяем, является ли вершина v проигрышной. Если да, то вершина u точно является выигрышной. Если же все вершины, в которые есть ход из u выигрышные, то в соответствии с теоремой, вершина u является проигрышной.

Заметим, что выигрышно-проигрышное разбиение вершин ациклического графа позволяет также найти *выигрышную стратегию* для того игрока, который выигрывает в соответствующей игре.

Вообще, *стратегией* игрока в игре на графе называют отображение s из множества нетерминальных вершин в множество вершин графа, причем для любой вершины u в графе должна присутствовать дуга $u \to s(u)$.

Стратегия указывает, в какую вершину следует пойти игроку из вершины u, если он должен сделать ход в соответствующей игре. Стратегия называется выигрышной для первого игрока в игре на графе G, если, следуя этой стратегии, первый игрок выигрывает, вне зависимости от действий второго игрока. Аналогично можно определить выигрышную стратегию для второго игрока.

Предложение 2. В любой игре на ациклическом графе существует стратегия s, такая что для игр, начинающихся в N-вершинах она является выигрышной для первого игрока, а для игр, начинающихся в P-вершинах — выигрышной для второго игрока.

$\triangleright$ Если $u \in N$, то в качестве $s(u)$ можно выбрать произвольную вершину $v \in P$, такую, что $uv \in E$. Если $u \in P$, то в качестве $s(u)$ можно выбрать произвольную вершину, в которую из u ведет дуга.

Следуя этой стратегии, игрок, который выигрывает в данной вершине, перед своим ходом всегда оказывается в N-вершине, и следовательно не может проиграть. ◁

Упражнение 1. Укажите, как модифицировать алгоритм построения выигрышно-проигрышного разбиения вершин ациклического графа, чтобы он находил также и выигрышную стратегию.

Пример 1. Ним Витхоффа. Задано две кучки камней. За один ход разрешается взять любое количество камней из одной из кучек, либо равное количество из обеих кучек. Тот, кто не может сделать ход — проигрывает.

Эта игра является игрой на графе, позицией является пара целых чисел — количество камней в кучках. В качестве визуализации нима Витхоффа можно использовать четверть бесконечной шахматной доски, на которой установлен шахматный ферзь. За один ход можно переместить ферзя вниз и/или влево. Если ферзь перед ходом находится в левом нижнем углу, то очередной игрок проиграл.

Используя анализ игры на ациклическом графе, легко выделить выигрышные и проигрышные позиции в ниме Витхоффа. Ясно, что для каждого i найдется не более одного j, что (i, j) является проигрышной позицией. Первые несколько проигрышных позиций в ниме Витхоффа — $(0, 0)$, $(1, 2)$, $(3, 5)$, $(4, 7)$, $(6, 10)$.

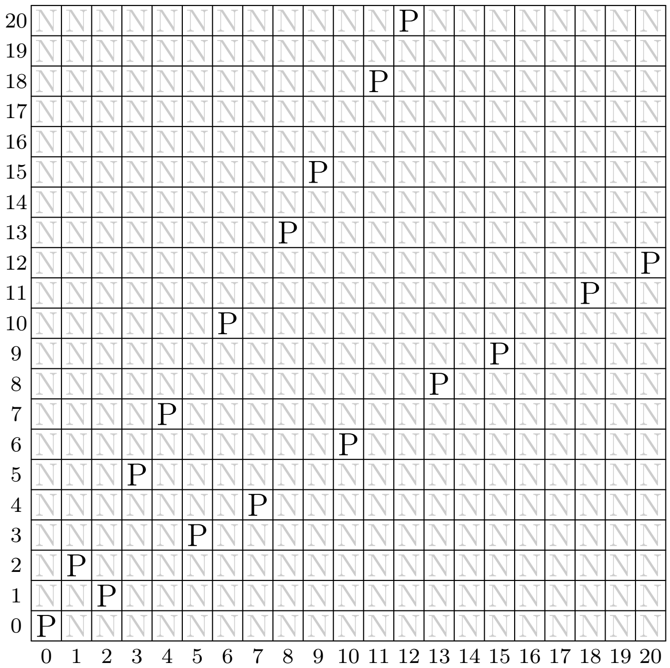


Рис. 5. Выигрышные и проигрышные позиции в ниме Витхоффа.

Пример 2. Вычитание квадратов. Задано положительное целое число n.

Своим ходом игрок может вычесть из n квадрат любого целого положительного числа. Игрок, которому достается число 0, проигрывает.

Эта игра является игрой на графе, позициями в игре являются неотрицательные целые числа. Используя метод анализа игры на ациклическом графе, получаем выигрышные и проигрышные позиции.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	N	P	N	N	P	N	P	N	N	P	N	P	N	N	P
-	1	-	1	2	-	1	-	1	2	-	1	-	1	2	-
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	...
N	P	N	N	P	N	P	N	N	N	N	N	N	N	N	...
1	-	1	2	-	1	-	1	2	5	2	5	4	3	5	...

Таблица 1. Выигрышные и проигрышные позиции в игре «Вычитание квадратов». В третьей строке указан выигрышный ход.

Вернемся теперь к общему случаю. В отличии от ациклических графов, не все игры на графе с циклами являются выигрышными или проигрышными. В частности, в графе может вообще не быть терминальных вершин, и тогда любая игра на этом графе будет ничейной.

Даже если можно построить формально разбиение вершин графа $G = \langle V, E \rangle$ на множества, удовлетворяющие соотношениям

$$P = \{u \mid \text{если } uv \in E, \text{ то } v \in N\};$$
$$N = \{u \mid \text{существует } uv \in E, \text{ такая что } v \in P\},$$

как это было сделано в теореме 1, оно может оказаться не единственным и не всегда имеет прежний выигрышно-проигрышный смысл. Например, в графе, приведенном на рис. 6, есть два способа построить такое разбиение, но при этом на самом деле все вершины в нем являются ничейными.

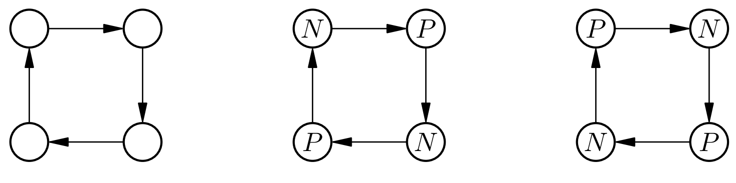


Рис. 6. Граф, в котором есть два N-P-разбиения. При этом все его вершины — ничейные.

Чтобы понять, кто выигрывает в игре на графе с циклами, нам потребуется немного уточнить цели игроков. Рассмотрим, например, граф, приведенный на рис. 7. Вершина y на этом рисунке является выигрышной, для того, чтобы выиграть, достаточно пойти в вершину x. Вершина же z является проигрышной, поскольку единственный ход из нее ведет в вершину y. С другой стороны, делать ход из y в z бессмысленно, из нее противник просто вернется в вершину y.

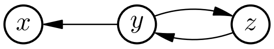


Рис. 7. Вершина y — выигрышная, а z проигрышная. Но ход из y в z не имеет смысла, противник возвращается в y.

Рассмотрим выигрышную игру A. Вне зависимости от действий второго игрока, первый игрок может выиграть. Потребуем, чтобы он действовал таким образом, чтобы количество ходов, которое потребуется для победы в худшем случае, было минимально. В свою очередь, потребуем, чтобы игрок, который проигрывает в данной позиции действовал так, чтобы в худшем для него случае игра продолжалась как можно дольше. Покажем, что в случае конечного графа эти требования всегда можно удовлетворить.

Рассмотрим граф $G = \langle V, E \rangle$. Обозначим как P_0 множество терминальных вершин графа. Это вершины, проигрышные за 0 ходов.

$$P_0 = \{u \mid \text{из } u \text{ не выходит ни одной дуги}\}$$

Вершины, в которых первый игрок может выиграть за один ход — это в точности в ершины, из которых существует ход в терминальную вершину. Обозначим множество этих вершин как N_1.

$$N_1 = \{u \mid \exists v \in P_0 : uv \in E\}$$

В свою очередь, вершины, в которых первый игрок проигрывает не более чем за два хода — это вершины, из которых все ходы ведут в множество N_1.

$$P_2 = \{u \mid \forall v : uv \in E \Rightarrow v \in N_1\}$$

Продолжая аналогичным образом, мы получаем множества P_{2i} и N_{2i+1} вершин, проигрышных не более чем за $2i$, и выигрышных не менее чем за $2i + 1$ ходов, соответственно, для всех i.

$$P_{2i} = \{u \mid \forall v : uv \in E \Rightarrow v \in N_{2i-1}\}$$
$$N_{2i+1} = \{u \mid \exists v \in P_{2i} : uv \in E\}$$

Несложно по индукции доказать, что $P_i \subset P_{i+2}$ и $N_i \subset N_{i+2}$ для всех допустимых i. Введем обозначения

$$P = \bigcup_{i \geq 0} P_i$$
$$N = \bigcup_{i \geq 0} N_i.$$

Поскольку каждое из множеств P_i и N_i является подмножеством конечного множества V, то для некоторого i должно выполняться равенства $P_{2i} = P$ и $N_{2i+1} = N$.

Упражнение 2. Покажите, что равенства $P_{2i} = P$ и $N_{2i+1} = N$ выполняются для $i = O(V)$, где V — число вершин в графе.

Покажем, что множества P и N содержат все проигрышные и выигрышные вершины в графе, соответственно.

Теорема 3. Множество P есть множество всех проигрышных вершин графа, множество N — множество всех выигрышных вершин, а множество $D = V \setminus (P \cup N)$ — множество всех ничейных вершин.

$\triangleright$ Проигрышность и выигрышность всех вершин из P и N, соответственно, легко доказать индукцией по номеру шага, на котором они были помещены в соответствующее множество.

Покажем, что лучшее, чего можно добиться в остальных вершинах — это ничья. Пусть фишка находится в вершине $u \in D$. Из этой вершины есть дуга в некоторую вершину v из множества D (если бы все дуги вели в вершины из N или P, то вершина также оказалась бы в N или P). При этом нет дуги, ведущей в вершину из P (иначе вершина была бы помещена в N). Перемещать фишку в вершину из N не имеет смысла — тогда противник выиграет. Поэтому переместим ее в v и применим аналогичное рассуждение к этой вершине.

Чтобы формализовать это рассуждение, можно, например, по индукции доказать, что при оптимальной игре для любого числа k фишка через k ходов будет все еще в вершине из множества D, и, следовательно, игра будет продолжаться бесконечно. $\triangleleft$

По определению множеств P_i и N_i легко видеть, как построить их в графе за $O(VE)$. Покажем как построить эти множества за время $O(E)$. Если в случае ациклического графа мы использовали для построения множеств P и N обход в глубину (точнее, он использовался при топологической сортировке графа), то теперь используем некоторое подобие обхода в ширину.

Пусть задан граф $G = \langle V, E \rangle$. Нашей целью будет для каждой вершины u выяснить, принадлежит ли она P либо N, а также найти минимальное $c(u)$ такое, что u лежит в множестве $P_{c(u)}$ либо $N_{c(u)}$. Функцию c иногда называют *функцией счета* для игры на графе G.

В качестве вспомогательного инструмента заведем для каждой вершины счетчик z_u. В нем будем хранить, сколько вершин, в которые ведут дуги из u, все еще не принадлежат множеству N. Исходно счетчик каждой вершины равен количеству исходящих из нее дуг. Если в процессе работы алгоритма счетчик вершины обнуляется, то она является проигрышной. Терминальные вершины исходно имеют нулевой счетчик и помещаются в множество P_0.

Мы будем по очереди просматривать вершины, которые уже классифицированы как принадлежащие P_i или N_i. Для организации просмотра используем очередь, в которую исходно помещаются все вершины из множества P_0. Рассмотрим очередную вершину u.

Пусть вершина u проигрышная. Тогда все вершины v, из которых есть дуги в u, являются выигрышными. Для некоторых из них это может быть уже известно, а остальные поместим в $N_{c(u)+1}$, положим для них $c(v) = c(u) + 1$ и добавим их в очередь.

Пусть вершина u, наоборот, является выигрышной. Переберем все входящие в нее дуги. Для каждой вершины v, такой что $vu \in E$, уменьшим счетчик z_v на единицу (мы нашли еще одну вершину из N, в которую из нее ведет дуга). Если счетчик стал равен нулю, то поместим вершину v в $P_{c(u)+1}$, положим $c(v) = c(u) + 1$ и добавим v в очередь.

Вершины, которые остались не классифицированы после опустошения очереди, являются ничейными.

Ясно, что во время работы алгоритма каждая дуга графа будет просмотрена $O(1)$ раз, следовательно время работы есть $O(E)$. Корректность алгоритма несложно доказывается индукцией по числу шагов.

Данный алгоритм в литературе иногда называют ретро-анализом.

Упражнение 3. Как мы уже видели в игре на рис. 7, при наличии циклов хода из выигрышной вершины в произвольную проигрышную может быть недостаточно для победы. Покажите, как с помощью функции $c(u)$ построить выигрышную стратегию для игрока, который выигрывает в анализируемой игре. Укажите, как следует модифицировать приведенный алгоритм, чтобы в результате своей работы он также формировал массив $s[u]$, указывающий для каждой вершины, в какую вершину игроку следует сделать из нее ход для достижения оптимального результата.

Упражнение 4. (Несправедливая игра) Пусть дуги ориентированного графа G раскрашены в красный, синий и зеленый цвет. Одна из вершин графа содержит фишку. Игра протекает как обычная игра на графе, за следующим исключением: по ребрам синего цвета фишку может перемещать только первый игрок, по ребрам красного цвета — второй, а про ребрам зеленого цвета —

оба игрока. Сведите задачу об определении победителя в несправедливой игре к задаче об определении победителя в справедливой игре на графе.

Упражнение 5. (Игра в поддавки) Рассмотрим игру на графе, но изменим критерий выигрыша. Пусть теперь игрок, который не может сделать ход, выигрывает. Сведите задачу об определении победителя в игре в поддавки к задаче об определении победителя в нормальной игре.

Замечание. Отметим, что в рассуждениях о минимизации числа ходов до выигрыша и максимизации числа ходов до проигрыша требование конечности графа существенно. На рис. 8 приведен пример проигрышной игры на бесконечном графе, в которой первый игрок может заставить игру продолжаться более k ходов для любого целого k.

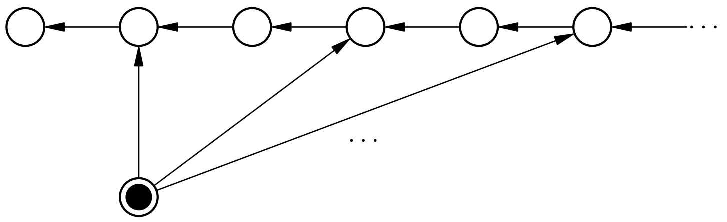


Рис. 8. Случай бесконечного графа. Вершина, в которой находится фишка — проигрышная, но первый игрок может заставить игру продолжаться более k ходов для любого числа k.

2. Сумма игр. Эквивалентность по Гранди

Анализ игры на графе с использованием выигрышно-проигрышного разбиения множества вершин, рассмотренный в предыдущем разделе, позволяет выяснить, кто из игроков выигрывает в игре на графе, за $O(E)$, где E — число ребер в графе. Если граф игры задан явно, то эта оценка, очевидно, является оптимальной.

Однако часто графы игр, которые можно задать компактно, оказываются очень большими. Простые примеры — игры ним и Хакенбуш, описанные во введении. Например, граф игры ним с кучками $\langle a_1, a_2, \ldots a_n \rangle$ имеет порядка $a_1 \times a_2 \times \ldots \times a_n$ вершин, а граф игры Хакенбуш на графе с n вершинами может иметь порядка 2^{n^2} вершин.

Теория прямой суммы игр, которую мы рассмотрим в этом разделе, позволяет проанализировать те игры, которые распадаются на несколько независимых частей, в одной из которых игрок должен сделать ход.

Такой является, к примеру, игра ним — количество камней, которые можно взять из одной кучки, не зависит от состояния других кучек. В этом случае, оказывается, с помощью отдельного анализа каждой из частей, можно сделать вывод о выигрышности или проигрышности игры в целом.

На самом деле, теория прямых сумм игр является даже более мощным инструментом, с ее помощью оказывается возможно проанализировать, например, такие игры, как Хакенбуш, в которых представление в виде прямой суммы независимых частей естественным образом не возникает.

В этом разделе нам потребуется сравнивать различные игры. Мы будем считать две игры $A = \langle G, u \rangle$ и $B = \langle H, v \rangle$ равными, если подграфы G и H, индуцированные вершинами, достижимыми из u и v, соответственно, изоморфны. Иначе говоря, при сравнении игр мы будем игнорировать те вершины их графов, которые не достижимы из начальных, а графы будем сравнивать с точностью до изоморфизма. Использование этого допущения позволяет уменьшить количество технических деталей в рассуждениях.

Рассмотрим две игры A и B. Прямая сумма $A+B$ игр A и B устроена следующим образом. Расположим графы игр A и B рядом, и в каждом из них поместим фишку в стартовую вершину. Игрок, который должен сделать ход в игре $A+B$ сначала выбирает одну из игр, и затем делает в ней ход. Если в одной из игр фишка находится в терминальной вершине, то эту игру выбирать нельзя. Если ход нельзя сделать ни в одной из двух игр, то тот, кто должен делать ход, проигрывает.

Покажем, что игра $A+B$ также является игрой на графе. Вершинами графа игры $A + B$ являются пары $\langle u_1, u_2 \rangle$ вершин графов игр A и B, соответственно. Дуги из вершины $\langle u_1, u_2 \rangle$ ведут в те вершины $\langle v_1, u_2 \rangle$, что в графе игры A есть дуга $u_1 v_1$, и в те вершины $\langle u_1, v_2 \rangle$, что в графе игры B есть дуга $u_2 v_2$.

Итак, операция сложения является двухместной операцией на множестве игр на графах. Легко видеть, что она является коммутативной и ассоциативной, то есть для любых игр A, B и C выполнены равенства

$$A + B = B + A,$$
$$(A + B) + C = A + (B + C).$$

Изоморфизм графов соответствующих игр проверяется тривиально. Таким образом, множество игр, наделенное операцией сложения, образует коммутативную полугруппу[2].

[2] *Полугруппой* называется множество, на котором введена некоторая операция (на-

Благодаря ассоциативности, мы можем писать $A_1 + A_2 + \ldots + A_n$ для прямой суммы более чем одной игры. Собственно, комбинаторный смысл суммы нескольких игр такой же как и у суммы двух — графы игр «располагаются рядом», и игрок выбирает одну из игр и делает в ней ход.

Пример 3. Рассмотрим игру ним, описанную во введении. Обозначим игру ним с одной кучкой из k камней как $*k$. Тогда игра с n кучками, содержащими $a_1, a_2, \ldots, a_n$ камней, соответственно, представляет собой сумму игр $*a_1 + *a_2 + \ldots + *a_n$.

Пример 4. Клетки прямоугольника $1 \times m$ пронумерованы от 0 до $m-1$ слева направо. В некоторых клетках находятся фишки. За один ход разрешается взять любую фишку и переместить ее влево, не более чем на d клеток. При этом в одной клетке может оказаться несколько фишек. Тот, кто не может сделать ход, поскольку все фишки находятся в самой левой клетке, проигрывает.

Если обозначить игру с одной фишкой, находящейся в клетке k как D_k, то игра с несколькими фишками, находящимися в клетках $a_1, a_2, \ldots, a_n$, представляет собой сумму игр $D_{a_1} + D_{a_2} + \ldots + D_{a_n}$.

Пример 5. Обобщение предыдущего примера. Пусть задан граф G, в некоторых вершинах которого находятся фишки. Разрешается за один ход переместить любую фишку по дуге, исходящей из вершины, в которой она находится. Несколько фишек могут находится в одной и той же вершине.

Ясно, что игра с одной фишкой в некоторой вершине u — это просто игра G_u на графе G. Поскольку фишки не взаимодействуют друг с другом, то игра с несколькими фишками представляет собой прямую сумму соответствующих игр: $G_{u_1} + G_{u_2} + \ldots + G_{u_n}$.

Рассмотрим игру, граф которой состоит из одной вершины и не содержит ребер. Обозначим эту игру как $*0$. Эта игра является нейтральным элементом относительно операции сложения игр, поскольку граф игры $A + *0$ естественным образом изоморфен графу игры A. Таким образом, для любой игры A выполнено равенство

$$A + *0 = A.$$

Следовательно полугруппа игр на самом деле является моноидом[3].

пример, можно обозначить ее как «+»), обладающая свойством ассоциативности: $(a+b)+c = a+(b+c)$. Например, полугруппу образую строки с операцией конкатенации. Если полугрупповая операция также коммутативна: $a+b = b+a$, то полугруппу логично назвать коммутативной.

Пример коммутативной полугруппы — вещественные числа с операцией взятия минимума.

[3] *Моноидом* называется полугруппа, в которой есть нейтральный элемент e, такой

Простейшим примером анализа прямой суммы игр является лемма о симметричной стратегии.

Лемма 4. (О симметричной стратегии) Если A — игра на ациклическом графе, то $A + A$ — проигрышная игра.

$\triangleright$ Какой бы ход первый игрок не совершил в одной из копий игры в сумме $A + A$, второй игрок просто повторяет этот ход в другой копии, снова предоставляя первому игроку возможность ходить в сумме одинаковых игр. Поскольку A — игра на ациклическом графе, игра рано или поздно завершится тем, что первый игрок получит пару игр в терминальных вершинах. $\triangleleft$

Для более точного анализа зададимся целью охарактеризовать все игры с точки зрения того, как они ведут себя при сложении с другими играми.

Любая игра является выигрышной, проигрышной или ничейной. Введем обозначение $v(A)$ для *значения* игры, положим

$$v(A) = \begin{cases} +1, & \text{если } A \text{ — выигрышная,} \\ -1, & \text{если } A \text{ — проигрышная,} \\ 0, & \text{если } A \text{ — ничейная.} \end{cases}$$

Можно считать, что игроки играют «на деньги», и проигравший платит выигравшему один рубль. В этом случае $v(A)$ означает выигрыш первого игрока в игре A.

Будем называть две игры A и B *неразличимыми* или *эквивалентными по Гранди* и писать $A \simeq B$, если для любой игры C выполнено равенство

$$v(A + C) = v(B + C).$$

Поскольку мы не будем использовать других отношений эквивалентности, мы будем называть эквивалентные по Гранди игры просто эквивалентными. Ясно, что введенное отношение действительно является отношением эквивалентности на множестве всех игр.

Лемма 5. Если $A_1 \simeq A_2$ и $B_1 \simeq B_2$, то $A_1 + B_1 \simeq A_2 + B_2$.

что для любого элемента a выполнено $a + e = e + a = a$. Например, в полугруппе строк с операцией конкатенации нейтральным элементом будет пустая строка.

На самом деле любую полугруппу можно пополнить нейтральным элементом, положив для всех элементов $a + e = e + a = a$. Например, если полугруппу вещественных чисел с операцией min пополнить нейтральным элементом (он в данном случае будет «бесконечностью»), то она превратится в моноид.

$\triangleright$ Действительно, для любой игры C выполняется равенство

$$v((A_1 + B_1) + C) = v(A_1 + (B_1 + C)) = v(A_2 + (B_1 + C)) = \\ v(B_1 + (A_2 + C)) = v(B_2 + (A_2 + C)) = v((A_2 + B_2) + C),$$

а значит, по определению, $A_1 + B_1 \simeq A_2 + B_2$. $\triangleleft$

Таким образом, неразличимые игры можно свободно заменять друг на друга в рассуждениях, если речь идет о сложении игр[4]. Отождествим все эквивалентные игры и рассмотрим операцию сложения на классах эквивалентности. Чтобы получить класс эквивалентности суммы классов $\mathcal{A}$ и $\mathcal{B}$ выберем в каждом из классов по представителю, скажем $A \in \mathcal{A}$ и $B \in \mathcal{B}$, и назовем суммой $\mathcal{A} + \mathcal{B}$ класс эквивалентности игры $A + B$. В силу леммы 5, результат не зависит от выбора конкретных представителей, и следовательно операция определена корректно.

Изучим подробнее, как устроены классы эквивалентности игр и каким образом на них действует операция сложения.

Начнем со следующего важного факта: все проигрышные игры эквивалентны. Это утверждение следует из следующей теоремы.

Теорема 6. Для любой проигрышной игры L и любой игры C выполнено равенство $v(L + C) = v(C)$.

$\triangleright$ Требуется рассмотреть три случая — C выигрышная, проигрышная или ничейная.

Если C выигрышная или проигрышная, то выигрывающий в C игрок (первый или второй, соответственно) играет в C в соответствии со своей выигрышной стратегией, а если противник делает ходы в L, то отвечает в ней в соответствии с выигрышной стратегией в L для второго игрока.

Пусть теперь C — ничейная. Проведем индукцию по числу ходов до проигрыша в L. Если L проигрышная за 0 ходов, то $L + C$ очевидно ничейная — в L просто нельзя сделать ход.

Пусть теперь L проигрышная, и для всех игр, в которых первый игрок проигрывает быстрее чем в L, утверждение теоремы доказано. Поставим себя на место первого игрока в игре $L + C$. Сможем ли мы выиграть?

Делать ход в L с этой целью бессмысленно — противник всегда может ответить в ней и перейти в пару игр, из которых одна — проигрышная за меньшее число ходов, чем L, а другая — все еще ничейная, поэтому мы не выиграем (а вот проиграть в случае неудачного хода в принципе можем). Ходить же в C в выигрышную позицию тем более не следует — тогда противник получит пару из проигрышной и выигрышной игр, в которой, как мы уже убедились, он выигрывает.

[4]Отношение эквивалентности, согласованное с некоторой операцией (в данном случае — с операцией сложения игр) называют *конгруэнтностью* или *конгруэнцией*.

Значит надо ходить в C в ничейную позицию, и игра $L + C$ — ничейная. $\lhd$

Следствие 7. Если A и B — проигрышные игры, то $A \simeq B$.

Таким образом, все проигрышные игры эквивалентны тривиальной игре $*0$. Ясно, что никакие две игры с разными исходами не могут быть эквивалентны, поэтому класс эквивалентности игры $*0$ состоит в точности из всех проигрышных игр.

Из леммы о симметричной стратегии следует, что для любой игры A на ациклическом графе выполнено соотношение

$$A + A \simeq *0.$$

С алгебраической точки зрения это означает, что класс эквивалентности игры A является обратным самому себе относительно операции сложения. Таким образом, классы эквивалентности игр на ациклических графах образуют относительно операции сложения *абелеву группу*, причем каждый класс является сам себе обратным.

Было бы удобно, если бы все выигрышные игры также были эквивалентны друг другу (и в этом случае описанная выше группа оказалась бы изоморфна группе $\mathbb{Z}/2\mathbb{Z}$), но это не так. Существует бесконечно много классов эквивалентности выигрышных игр, что легко продемонстрировать на примере игры ним с одной кучкой.

Будем по прежнему обозначать игру ним с одной кучкой из i камней как $*i$. Ясно, что $*0$ — проигрышная игра, а для $i > 0$ игра $*i$ — выигрышная (достаточно взять все камни). Оказывается, игры $*i$ и $*j$ не эквивалентны ни для каких $i \neq j$.

Предложение 8. Если $i \neq j$, то игры $*i$ и $*j$ не эквивалентны по Гранди.

$\rhd$ Пусть $i < j$. Возьмем в качестве C игру $*i$. Тогда игра $*i + *i$ проигрышная по лемме о симметричной стратегии.

С другой стороны, игра $*j + *i$ выигрышная — взяв из кучки с j камнями $j - i$ камней, мы получаем проигрышную игру $*i + *i$. $\lhd$

Следовательно для каждого i игра $*i$ порождает свой класс эквивалентности. Оказывается, этими классами игры на ациклических графах исчерпываются. Иначе говоря, любая игра на ациклическом графе эквивалентна по Гранди игре ним с одной кучкой камней. Данное утверждение вместе с правилом сложения классов эквивалентности игры ним составляют основу теории Шпрага–Гранди для игр на ациклических графах.

Упражнение 6. (Несправедливая игра) В упражнении 4 было выполнено сведение несправедливой игре к справедливой. Почему этим сведением нельзя воспользоваться для каждого слагаемого прямой суммы игр по отдельности с целью анализа суммы несправедливых игр?

Упражнение 7. (Игра в поддавки) В упражнении 5 было выполнено сведение игры в поддавки к нормальной игре. Почему этим сведением нельзя воспользоваться для каждого слагаемого прямой суммы игр по отдельности с целью анализа суммы игр в поддавки?

3. Теория Шпрага–Гранди для игр на ациклических графах

Докажем сначала вспомогательную лемму. Отметим (это нам понадобится в будущем), что в этой лемме игра A не обязательно должна быть игрой на ациклическом графе.

Лемма 9. (Об эквивалентности ниму) Игра A эквивалентна игре $*i$ тогда и только тогда, когда $A + *i \simeq *0$.

$\triangleright$ Если A эквивалентна $*i$, то $A + *i \simeq *i + *i \simeq *0$.

Наоборот, если $A+*i \simeq *0$, то $A \simeq A+*0 \simeq A+(*i+*i) \simeq (A+*i)+*i \simeq *0 + *i \simeq *i$. $\triangleleft$

Рассмотрим конечное множество неотрицательных целых чисел X. Обозначим как $\operatorname{mex} X$ минимальное неотрицательное целое число, которое не встречается в X (обозначение mex происходит от английского выражения *m*inimal *ex*cluded — минимальное не входящее). Например, $\operatorname{mex}\{0, 1, 3, 5, 9\} = 2$.

Теорема 10. (Шпраг, Гранди) Если A — игра на ациклическом графе, то $A \simeq *i$ для некоторого числа i.

Будем обозначать число i, такое что $A \simeq *i$, как $g(A)$. Если из игры A существуют ходы в игры $A_1, A_2, \dots, A_n$, то

$$g(A) = \operatorname{mex}\{g(A_1), g(A_2), \dots g(A_n)\}.$$

$\triangleright$ Доказательство будем вести по индукции, параметром индукции будет выступать длина максимального пути в графе игры A от начальной вершины до терминальной. Эта величина ограничивает максимальное количество ходов в процессе игры A. Если эта длина равна 0, то начальная вершина игра A является терминальной, и следовательно она является проигрышной и эквивалентна $*0$.

Пусть для всех игр с длиной максимального пути меньше чем в A утверждение теоремы доказано. Докажем его для игры A. Пусть из игры

A существуют ходы в игры $A_1, A_2, \ldots, A_n$. Максимальная длина пути из начальной вершины до терминальной в каждой из этих игр меньше чем в игре A, значит каждая из них эквивалентна игре ним. Выберем $i = \text{mex}\{g(A_1), g(A_2), \ldots g(A_n)\}$.

По лемме об эквивалентности ниму для того, чтобы доказать что $A \simeq *i$, достаточно показать, что $A + *i$ — проигрышная игра.

Пусть первый игрок сделал в рассматриваемой сумме ход в игре A. Теперь второй игрок должен ходить в сумме $A_j + *i$ для некоторого j. Поскольку $A_j \simeq *g(A_j)$, и $i \neq g(A_j)$, то $A_j + *i$ — выигрышная игра и второй игрок выигрывает.

Значит первый игрок должен сделать ход в $*i$. Такой ход возможен только в случае $i > 0$. После этого хода второй игрок должен делать ход в сумме $A + *k$ для некоторого $k < i$. Поскольку i — минимальное число среди тех, которые не встречаются среди $g(A_1), \ldots, g(A_n)$, то для некоторого j выполнено $g(A_j) = k$. Тогда второй игрок может сделать в A ход в игру A_j и первый игрок получит игру $A_j + *k$, которая является проигрышной, так как $A_j \simeq *k$.

Итак, вне зависимости от своих действий первый игрок проигрывает, следовательно $A + *i$ — проигрышная игра, следовательно $A \simeq *i$, что и требовалось. $\triangleleft$

Число i из предыдущей теоремы называется *числом Шпрага–Гранди* или *числом Гранди* для игры A, и обозначается как $g(A)$. Отображение из множества игр на ациклических графах в множество неотрицательных целых чисел, которое сопоставляет игре ее число Гранди, называется *функцией Гранди* или *функционалом Шпрага–Гранди*. Если рассматривать только игры на некотором заданном графе, то часто функцией Гранди для этого графа называют отображение множества вершин этого графа в неотрицательные целые числа, которое сопоставляет вершине число Гранди соответствующей ей игры.

Теорема 10 дает конструктивный способ вычисления функции Гранди для вершин заданного графа. Перебирая вершины в порядке обратной топологической сортировки, для каждой вершины находим минимальное число, которое не является числом Гранди ни для одного из ее потомков. Оно и будет числом Гранди для этой вершины. Поскольку каждая дуга графа будет просмотрена в этом процессе ровно один раз, а найти минимальное число, которое не встречается в множестве из n чисел, можно за $O(n)$, то время работы алгоритма составляет $O(E)$.

Упражнение 8. Укажите, как найти минимальное целое неотрицательное число, которое не встречается в множестве из n чисел, за $O(n)$. Учтите, что сами числа могут быть больше чем n.

Пример 6. Рассмотрим в качестве примера ним Витхоффа, рассмотренный в примере 1. На рис. 9 приведены значения функции Гранди для игр с размерами кучек не больше 20.

20	20	18	19	22	21	23	24	25	26	27	28	3	0	1	29	11	30	9	31	12	17
19	19	20	18	17	14	21	11	16	24	22	23	1	5	4	26	27	28	10	13	25	12
18	18	19	20	21	17	16	15	22	23	4	5	0	3	24	25	7	11	26	12	13	31
17	17	15	16	13	18	11	14	12	19	3	4	5	23	22	8	24	25	21	26	10	9
16	16	17	15	14	19	18	20	21	12	2	1	4	6	10	22	9	13	25	11	28	30
15	15	16	17	18	10	13	12	19	14	0	3	21	22	8	23	20	9	24	7	27	11
14	14	12	13	16	15	17	18	10	9	1	2	20	21	7	11	23	22	8	25	26	29
13	13	14	12	11	16	15	17	2	0	5	6	19	20	9	7	8	10	22	24	4	1
12	12	13	14	15	11	9	16	17	18	19	7	8	10	20	21	22	6	23	3	5	0
11	11	9	10	7	12	14	2	13	17	6	18	15	8	19	20	21	4	5	0	1	3
10	10	11	9	8	13	12	0	15	16	17	14	18	7	6	2	3	1	4	5	23	28
9	9	10	11	12	8	7	13	14	15	16	17	6	19	5	1	0	2	3	4	22	27
8	8	6	7	10	1	2	5	3	4	15	16	17	18	0	9	14	12	19	23	24	26
7	7	8	6	9	0	1	4	5	3	14	15	13	17	2	10	19	21	12	22	16	25
6	6	7	8	1	9	10	3	4	5	13	0	2	16	17	18	12	20	14	15	11	24
5	5	3	4	0	6	8	10	1	2	7	12	14	9	15	17	13	18	11	16	21	23
4	4	5	3	2	7	6	9	0	1	8	13	12	11	16	15	10	19	18	17	14	21
3	3	4	5	6	2	0	1	9	10	12	8	7	15	11	16	18	14	13	21	17	22
2	2	0	1	5	3	4	8	6	7	11	9	10	14	12	13	17	15	16	20	18	19
1	1	2	0	4	5	3	7	8	6	10	11	9	13	14	12	16	17	15	19	20	18
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Рис. 9. Значение функции Гранди для нима Витхоффа.

Поскольку сумма двух игр на ациклическом графе снова является игрой на ациклическом графе, то должно быть правило, которое позволяет по числам Гранди игр A и B найти число Гранди их суммы $A+B$. В силу теоремы 10 достаточно найти это правило для игры ним. Оказывается, это правило очень простое.

Для целых неотрицательных чисел a и b обозначим как $a \oplus b$ побитовое исключающее или a и b. Например, $3 \oplus 5 = 6$. Эту операцию обычно называют xor (от английского e$x$clusive or).

Отметим некоторые простые свойства операции $\oplus$:

$$a \oplus b = b \oplus a,$$
$$(a \oplus b) \oplus c = a \oplus (b \oplus c),$$
$$a \oplus a = 0.$$

Именно операция $\oplus$ на целых числах оказывается аналогом операции сложения на играх. Поэтому в теории комбинаторных игр операцию $\oplus$ часто называют *ним-сложением* (по-английски: nim sum).

Теорема 11. (Шпраг, Гранди) $*i + *j \simeq *(i \oplus j)$.

$\triangleright$ Снова проведем доказательство по индукции. На этот раз параметром индукции будет сумма $i + j$.

В случае, когда $i + j = 0$ каждое из чисел i и j равно нулю, и следовательно $*i + *j = *0 + *0 \simeq *0$.

Пусть $i + j > 0$ и для всех меньших значений этой суммы теорема доказана. Для доказательства теоремы достаточно показать, что

$$i \oplus j = \operatorname{mex} F(i, j),$$

где $F(i, j)$ — множество чисел Гранди всех игр, в которые возможен ход из игры $*i + *j$:

$$F(i,j) = \{i \oplus 0, i \oplus 1, \ldots i \oplus (j-1), 0 \oplus j, 1 \oplus j, \ldots (i-1) \oplus j\}.$$

Во-первых заметим, что $i \oplus j$ не лежит в $F(i, j)$. Действительно, если $i \oplus j = i \oplus k$, то $j = k$, аналогично если $i \oplus j = l \oplus j$, то $i = l$. Осталось доказать, что все числа меньшие $i \oplus j$ есть в этом множестве.

Зафиксируем число $r < i \oplus j$. Пусть t — номер самого старшего разряда, в котором r отличается от $i \oplus j$. Ясно, что r имеет в этом разряде ноль, а $i \oplus j$ — единицу. Значит t-й разряды i и j различны, пусть, не ограничивая общности, t-й разряд числа i равен единице. Тогда построим число i' следующим образом: разряды старше t-го в i' сделаем равными соответствующим разрядам i, t-й разряд обнулим, а более младшие выберем таким образом, чтобы выполнялось равенство $i' \oplus j = r$:

$$\begin{array}{ccccccccc}
i & = & i_u & \ldots & i_{t+1} & i_t = 1 & i_{t-1} & \ldots & i_0 \\
j & = & j_u & \ldots & j_{t+1} & j_t = 0 & j_{t-1} & \ldots & j_0 \\
i \oplus j & = & (i \oplus j)_u & \ldots & (i \oplus j)_{t+1} & (i \oplus j)_t = 1 & (i \oplus j)_{t-1} & \ldots & (i \oplus j)_0 \\
\vee & & \shortparallel & \ldots & \shortparallel & \vee & & \ldots & \\
r & = & r_u & \ldots & r_{t+1} & r_t = 0 & r_{t-1} & \ldots & r_0 \\
i' & = & i_u & \ldots & i_{t+1} & 0 & (j \oplus r)_{t-1} & \ldots & (j \oplus r)_0
\end{array}$$

Поскольку, по построению, $i' < i$, то число $i' \oplus j = r$ встречается в $F(i, j)$.

Таким образом, $i \oplus j$ является минимальным числом, которое не встречается в множестве $F(i, j)$, а следовательно $*i + *j \simeq *(i \oplus j)$, что и требовалось. $\lhd$

Итак, по графу игры A и ее начальной позиции можно за линейное время вычислить $g(A)$. Кроме того, получено простое правило, позволяющее по $g(A)$ и $g(B)$ вычислить $g(A + B)$, а именно:

$$g(A + B) = g(A) \oplus g(B).$$

Таким образом, теория Шпрага–Гранди позволяет эффективно определять победителя в игре, представимой в виде прямой суммы игр, за время пропорциональное сумме размеров графов этих игр, а не произведения, как в случае непосредственного применения алгоритма анализа игры на графе.

Мы рассмотрим несколько примеров применения этой теории для анализа конкретных игр.

Пример 7. Игра ним. В данном случае мы имеем наиболее непосредственное применение теории. Поскольку $*i + *j \simeq *(i \oplus j)$, то в силу ассоциативности $*a_1 + *a_2 + \ldots + *a_n \simeq *(a_1 \oplus a_2 \oplus \ldots \oplus a_n)$. Поскольку среди $*i$ лишь игра $*0$ является проигрышной, получаем следующий алгоритм определения того, кто выигрывает в игре ним.

Если $a_1 \oplus a_2 \oplus \ldots \oplus a_n = 0$, то позиция проигрышная, выигрывает второй игрок. Иначе позиция выигрышная, выигрывает первый.

Теперь изучим, как найти выигрышный ход, если игра выигрышна. Обозначим $a_1 \oplus a_2 \oplus \ldots \oplus a_n$ как a. Пусть старший единичный бит в a находится в k-м разряде. Значит у нечетного количества чисел среди $a_1 \ldots a_n$ в k-м разряде находится единица. Выберем любое такое число a_i. Заменим в нем k-й разряд на ноль, а более младшие разряды сделаем единицами или нулями таким образом, чтобы количество единиц в каждом из этих разрядов среди всех чисел стало четным. Получим число $a_i' < a_i$, причем $a_1 \oplus a_2 \oplus \ldots \oplus a_i' \oplus \ldots \oplus a_n = 0$. Отметим, кстати, что $a_i' = a_i \oplus a$.

Таким образом, выигрышный ход заключается в том, чтобы из кучки, содержащей a_i камней, взять $a_i - a_i'$ камней.

Рассмотрим пример: пусть есть 4 кучки, которые содержат 2, 3, 5 и 7 камней, соответственно. $2 \oplus 3 \oplus 5 \oplus 7 = 3$, $3_{10} = 11_2$, старший единичный бит в числе 3 находится в первом разряде (разряды нумеруются с нуля). В этом разряде бит установлен у чисел 2, 3 и 7. Значит возможные выигрышные ходы: $2 \to 1$, $3 \to 0$ и $7 \to 4$.

Отметим, что тот же алгоритм поиска выигрышного хода можно применить и в произвольной прямой сумме игр на графе, только вместо количества камней в кучке следует рассматривать функцию Гранди соответствующей игры.

Пример 8. Игра «ОМакс»[5]. Приведем пример игры, в которой концепция прямой суммы возникает менее естественным образом.

Рассмотрим окружность, вдоль которой нанесены n различных точек. За один ход разрешается провести хорду, которая не должна иметь общих точек с ранее проведенными хордами (в том числе они не должны иметь общих концов). Игрок, который не может провести хорду по этим правилам, считается проигравшим. Пример позиции в игре для $n = 7$ после хода первого игрока и возможные ходы второго игрока приведены на рис. 10.

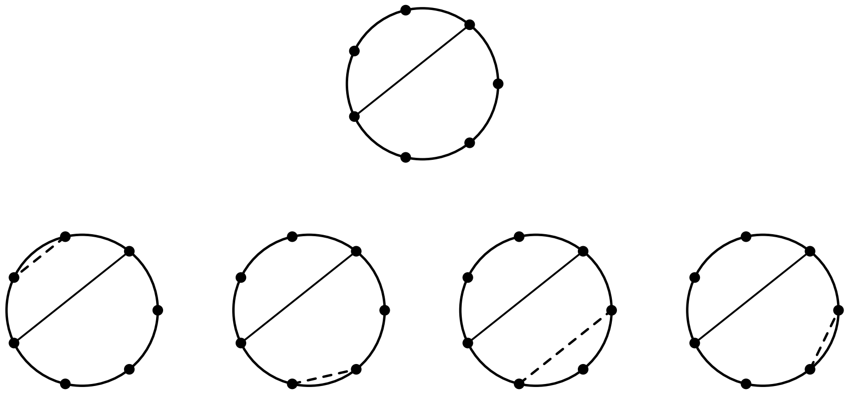


Рис. 10. Пример позиции и возможных ходов в игре «ОМакс».

Посмотрим, как в этой игре можно применить теорию прямой суммы. Рассмотрим, например, позицию, сложившуюся в игре на рисунке после первого хода первого игрока. Заметим, что ни одна из будущих хорд не должна пересекать уже проведенную, следовательно ни одна их будущих хорд не будет иметь концы по разные стороны от нее. Таким образом, по сути, процесс игры по разные стороны от хорды протекает независимо. Игрок перед ходом выбирает, в какой из частей он сделает ход, и делает в ней ход. Каждая часть окружности при этом эквивалентна целой окружности (действительно, факт, что окружность является именно окружностью, активно не используется, вместо окружности можно было взять любую границу выпуклой фигуры). Если обозначить игру с n точками как M_n, то игра сверху на рис. 10 эквивалентна M_2+M_3. Вообще, после первого хода игра M_n превращается в сумму $M_{k_1}+M_{k_2}$, где $k_1 + k_2 = n - 2$ (k_1 и k_2 могут быть равны нулю).

Игра внутри каждой из них протекает также, после каждого хода игра распадается в прямую сумму двух игр.

Замечание. Отметим, кстати, что неверно, что каждая игра в прямой сумме соответствует отрезку окружности между концами уже проведенных хорд,

[5]Название этой игры не является общепринятым, с таким названием задача об анализе этой игры предлагалась на летних сборах школьников, кандидатов в сборную России по информатике, в 2003 году.

см. например, рис. 11, что, впрочем с точки зрения анализа игры совершенно не важно.

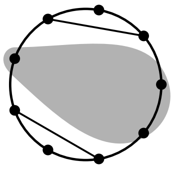


Рис. 11. Пример, когда одна из игр прямой суммы соответствует нескольким отрезкам на окружности.

Таким образом, получаем следующий алгоритм определения победителя в игре «ОМакс»: игры M_0 и M_1 являются проигрышными, поскольку в них нельзя сделать ни одного хода. Их функции Гранди равны нулю. Пусть известны функции Гранди всех игр M_k для $k < n$. Тогда $g(M_n) = \operatorname{mex}\{g(M_{k_1}) \oplus g(M_{k_2}) | k_1 + k_2 = n - 2\}$.

Пример 9. Крестики и крестики. Два игрока играют на поле $1 \times n$ ($n \geq 3$), своим ходом игрок может поставить крестик в любую свободную клетку. Игрок, после хода которого на поле есть ряд из трех крестиков, побеждает.

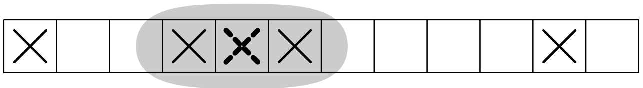


Рис. 12. Позиция в игре «Крестики и крестики» и выигрышный ход.

Чтобы проанализировать эту игру с помощью теории прямой суммы, надо сначала переформулировать ее правила, чтобы выполнялось условие «тот, кто не может сделать ход — проигрывает». Заметим, что если перед ходом игрока на поле есть два крестика рядом, то он всегда может выиграть, поставив рядом с ними третий крестик. Следовательно ставить крестик рядом с уже стоящим крестиком бессмысленно — следующим ходом немедленно проигрываешь. Аналогично, нельзя ставить два крестика через клетку друг от друга, противник тоже немедленно выигрывает, поставив крестик между ними.

Таким образом, можно переформулировать правила игры так: Два игрока играют на поле $1 \times n$ ($n \geq 3$). Пронумеруем клетки от 1 до n слева направо. Своим ходом игрок может поставить крестик в свободную клетку i, если все существующие из пяти клеток $i-2$, $i-1$, i, $i+1$ и $i+2$ свободны. Тот, кто не может сделать ход — проигрывает.

Теперь заметим, что сделав ход, игрок разбивает последовательный отрезок свободных клеток на два отрезка, на каждом из которых игра протекает независимо. Таким образом игру можно представить как сумму игр на каждом из отрезков. Есть лишь одна тонкость — некоторые отрезки ограничены крестиками (там ставить крестик на крайнюю клетку и на клетку, соседнюю с крайней, нельзя), а некоторые — краями поля (около края поля ставить крестик можно). Для единообразия можно добавить с обоих краев поля по две клетки и считать, что на них нельзя ставить крестики.

Итого, получаем следующую формулу для функции Гранди игры C_n на поле размером $1 \times n$ клеток (где n — размер поля после его расширения на две клетки в каждую сторону): $g(C_n = \mathrm{mex}\{g(C_i) \oplus g(C_{n-i-5}) \mid 0 \le i \le n-5\}$.

Пример 10. Октальные игры. Обобщением предудущиз примера являются игры, которые в англоязычной литературе называют *taking and breaking games*, *игры типа «бери и разбивай»*. Общая конструкция такой игры следующая. Позицией в игре является набор кучек камней. Возможны следующие три типа ходов:

1) забрать целиком кучку из k камней;

2) забрать из кучки k камней, если в ней строго больше k камней;

3) забрать из кучки k камней, если в ней не меньше больше $k+2$ камней, и разбить оставшиеся камни на две непустые кучки.

При этом для каждого k может быть разрешено различное множество типов ходов. Например, может быть разрешено забрать целиком кучку из 2 камней, а также также забрать из кучки, содержащей более трех камней 3 камня, а если кучка содержала пять и более камней, то после этого разбивать ее на две части.

Правила такой игры (для каких k какие типы ходов возможны) можно описать в виде кодовой последовательности восьмеричных цифр: $0.a_1a_2a_3\ldots a_k\ldots$ (в общем случае бесконечной), здесь a_k — цифра от 0 до 7 — задает возможные типы ходов, когда из кучки забирается k камней: младший бит отвечает за возможность забрать кучку из k целиком, средний — за возможность забрать k камней из кучки, если в ней еще остаются камни, и старший — за возможность забрать k камней и затем разбить остаток на две непустые кучки. Начальные 0. будем пока считать лишь частью обозначения. Например, описанная выше игра имеет код 0.016.

Игра, которую можно закодировать таким способом называется *октальной.* Игра ним в этой нотации имеет код $0.333333\ldots$ (тройки продолжаются до бесконечности). Игра «ОМакс», рассмотренная в примере 8, имеет код 0.07, а игра «Крестики и крестики» — код 0.00007. Иногда октальными называют только игры, у которых после точки лишь конечное число отличных от нуля разрядов.

Любая октальная игра может быть проанализирована с помощью динамического программирования способом, аналогичным тому, которым мы проанализировали игры «ОМакс» и «Крестики и крестики» — после первого хода игра сводится к одной либо двум меньшим играм, функции Гранди которых известны.

С октальным играми связана важная, в настоящий момент не доказанная гипотеза. Зафиксируем октальную игру с конечным количеством ненулевых разрядов после точки, и рассмотрим последовательность $G(0), G(1), G(2), \ldots, G(n), \ldots$ значений функции Гранди для одной кучки с $0, 1, 2, \ldots, n, \ldots$ камнями, соответственно. Имеется гипотеза о том, что для любой такой игры эта последовательность является периодичной. Несмотря на то, что периодичность была установлена для многих октальных игр, в общем виде данная проблема остается открытой. Отметим, что для некоторых достаточно простых игр предпериод и период оказываются очень большими, например для игры 0.16 предпериод равен 105351, а период — 149459. Еще более впечатляющими выглядят числа для игры 0.106 — предпериод 465384263797 и период 328226140474. Большинство известных в настоящее время периодов и предпериодов функций Гранди октальных игр, а также много другой интересной информации об этом классе игр можно найти на интернет-ресурсе `http://wwwhomes.uni-bielefeld.de/achim/octal.html` [10].

Упражнение 9. Доказательство периодичности функции Гранди октальной игры может быть выполнено автоматически.

Рассмотрим октальную игру, в которой после точки следует k разрядов, а затем все разряды равны нулю (иначе говоря, максимальное количество камней, которое разрешается взять из кучки, равно k). Пусть мы хотим доказать, что предпериод ее функции Гранди равен t, а период равен p. Докажите, что достаточно проверить равенство $G(i) = G(i - p)$ для всех i от $x + p$ до $2x + 2p + k$ (убедитесь, что если это равенство выполнено для всех таких i, то для $i > 2x + 2p + k$ при вычислении функции Гранди mex берется в том же множестве, что и при вычислении для $i - p$).

Пример 11. Дальнейшее обобщение игр типа «бери и разбивай». Октальные игры можно далее обобщить по двум направлениям. Во-первых, ноль перед точкой появился не случайно. Можно разрешить разбивать кучку на две непустых, не забирая из нее камней вообще. Это соответствует «нулевому» разряду — если разрешить такой ход в дополнение к разрешенным в игре $0.a_1a_2\ldots$, то получится игра $4.a_1a_2\ldots$.

Кроме того, можно разрешить разбивать кучку не только на две, но и на большее число частей. Например, можно добавить еще один бит, разрешив разбивать кучку на три непустых части. Получатся *шестнадцатеричные* игры. Вообще, можно вместо «цифры» считать a_i числом, и считать что его j-й двоичный разряд равен единице, если разрешается забрать из кучки i камней, разбив ее затем на j непустых кучек (отметим, что это удачно согласуется с октальной нотацией, что неслучайно).

Упражнение 10. Проанализируйте варианты нима, в которых разрешается разбивать кучки — игры $0.777777\ldots$, $4.333333\ldots$ и $4.777777\ldots$ (во всех трех случаях считается, что цифры после точки продолжаются до бесконечности).

Пример 12. Игра Гранди. Одна из простейших игр, которая не может быть записана в виде октальной либо аналогичной нотации — игра Гранди. Позиция в этой игре, как и ранее — множество кучек камней. Однако на этот раз за один ход можно разбить любую кучку на две *неравные*.

Посмотрим на функцию Гранди кучки из n камней для игры Гранди.

n	$G(n)$	n	$G(n)$	n	$G(n)$	n	$G(n)$	n	$G(n)$
0	0	20	0	40	1	60	2	80	5
1	0	21	4	41	5	61	1	81	2
2	0	22	3	42	4	62	3	82	4
3	1	23	0	43	1	63	2	83	5
4	0	24	4	44	5	64	4	84	2
5	2	25	3	45	4	65	3	85	4
6	1	26	0	46	1	66	2	86	3
7	0	27	4	47	5	67	4	87	7
8	2	28	1	48	4	68	3	88	4
9	1	29	2	49	1	69	2	89	3
10	0	30	3	50	0	70	4	90	7
11	2	31	1	51	2	71	3	91	4
12	1	32	2	52	1	72	2	92	3
13	3	33	4	53	0	73	4	93	7
14	2	34	1	54	2	74	3	94	4
15	1	35	2	55	1	75	2	95	3
16	3	36	4	56	5	76	4	96	5
17	2	37	1	57	2	77	3	97	2
18	4	38	2	58	1	78	2	98	3
19	3	39	4	59	3	79	4	99	5

Видно, что функция «старается» стать периодической с периодом 3, но каждый раз как только кажется, что ей это уже удалось, неожиданно период портится и появляются неподходящие значения.

Периодичность функции Гранди игры Гранди до сих пор не доказана, хотя ее значения посчитаны до 2^{35} (много данных по значениям функции Гранди игры Гранди на странице `http://wwwhomes.uni-bielefeld.de/achim/grundy.html` [9]). Известно 42 значения n, для которых $G(n) = 0$ (42 проигрышных позиции из одной кучки), максимальное из этих значений — 1222.

То, что значения функции Гранди для игры Гранди вычислены для n до 2^{35} впечатляет. Действительно, ведь наивный алгоритм вычисления функции Гранди работает за $O(n^2)$ — для вычисления $G(n)$ надо перебрать все пары (i, j), где $i \neq j$, $i + j = n$, а таких пар $O(n)$. Поскольку для вычисления $G(n)$ нам требуются значения $G(i)$ для всех $i < n$, то наивному алгоритму для вычисления $G(n)$ требуется $O(n^2)$ времени.

Покажем, как можно ускорить алгоритм за счет элегантного эвристического метода, разбивающего значения функции Гранди игры на «редкие» и «частые». Любители строгих математических построений найдут последующий текст (до конца примера) несколько неформальным, и может даже декларативным. Однако, несмотря на нестрогость обоснования, приведенный метод является единственным известным в настоящее время методом, позволяющим вычислять значения функции Гранди для игр с очень большим числом камней, поэтому рассмотрим его.

Сначала заметим, что функция Гранди игры Гранди растет достаточно медленно — для $n \leq 100\,000$ значение $G(n)$ не превышают 230. Кроме того, некоторые значения (например, 0, 1, 6, 7) встречаются редко, а некоторые (например, 2, 3, 4, 5) — часто. Частично формализуем эти наблюдения.

Разобьем все неотрицательные числа на два множества: редкие значения R и частые значения C (от английских слов «rare» — редкий и «common» — частый). При этом пусть среди значений $G(i)$ значения из множества C встречаются намного чаще значений из R. Мы не будем давать строгого определения выражению «намного чаще», поэтому анализ времени работы будет нестрогим.

При разбиении на R и C потребуем, чтобы выполнялись следующие свойства замкнутости: $a \oplus b$ лежит в множестве R, если a и b выбраны из одного и того же множества, и в C, если из разных.

$$\begin{gathered}
a \in R,\, b \in R \Rightarrow a \oplus b \in R \\
a \in R,\, b \in C \Rightarrow a \oplus b \in C \\
a \in C,\, b \in R \Rightarrow a \oplus b \in C \\
a \in C,\, b \in C \Rightarrow a \oplus b \in R
\end{gathered}$$

Пусть известны значения $G(i)$ для i от 1 до $n-1$ и мы хотим вычислить значение $G(n)$. Известно, что $G(n) = \operatorname{mex} X$, где

$$X = \{G(i) \oplus G(n-i) \mid 1 \leq i \leq n-1, i \neq n-i\}.$$

Вычислим $G(i) \oplus G(n-i)$ для всех значений $1 \leq i \leq n-1$, $i \neq n-i$, таких что $G(i) \in R$:

$$X' = \{G(i) \oplus G(n-i) \mid 1 \leq i \leq n-1, i \neq n-i, G(i) \in R\}.$$

Заметим, что при этом все значения X, которые лежат в C окажутся вычисленными. Найдем теперь минимальное число a из C, такое что $a \notin X'$. Ясно, что $G(n) \leq a$. Если $G(n) \in C$, то $G(n) = a$. Поскольку в множестве $\{G(1), G(2), \ldots, G(n-1)\}$ большинство значений лежат в C, а для $a, b \in C$ значение $a \oplus b$ лежит в R, то скорее всего все значения из R, не превышающие a, окажутся в множестве X.

Из этого можно сделать два вывода: во-первых, скорее всего $G(n)$ лежит в C (то есть является частым значением — частое множество имеет свойство оставаться частым, а редкое, соответственно, редким), во-вторых, если

$G(n)$ лежит в C, то для его определения достаточно знать X', а это множество можно построить за $O(r)$, где r — количество редких значений среди $\{G(1), G(2), \ldots, G(n-1)\}$.

Приведем теперь окончательный алгоритм вычисления $G(n)$. Алгоритм состоит из трех частей.

Часть 1. Вычислим X'. Эта часть работает за $O(r)$.

Часть 2. Найдем $a = \operatorname{mex}(X' \cup R)$. Вычисляя a, заодно найдем c — количество значений $b \in R$, $b \leq a$, таких что b не лежит в X'. $G(n)$ равно либо a, либо одному из таких b. Эта часть также работает за $O(a)$.

Часть 3. Теперь надо убедиться, что $G(n) = a$ (либо, что это не так). Будем перебирать i, такие что $1 \leq i \leq n-1$, $i \neq n-i$, $G(i) \in C$, $G(n-i) \in C$, и каждый раз, когда значение $G(i) \oplus G(n-i)$ не лежит в X', будем добавлять его туда, уменьшая счетчик c. Как только c станет равно нулю, станет ясно, что $G(n) = a$. Если этого так и не произойдет, то ничего не сделать, $G(n) \in R$, придется вычислить его как $\operatorname{mex} X$. Отметим, что после выполнения этой части алгоритма $X = X'$.

Таким образом время вычисления $G(n)$ в целом есть $O(r + a + t)$, где t — количество значений i, которые необходимо просмотреть во время выполнения третьей части. Хотя точно оценить это количество не удается, на практике оно обычно невелико (поскольку функция Гранди растет медленно).

Осталось указать, как разбить неотрицательные целые числа на множества R и C. Отметим, что благодаря свойствам замкнутости, достаточно про каждую степень двойки указать, принадлежит ли она R, или C. Приведем аргументацию, согласно которой большие степени двойки скорее всего принадлежат множеству C. Действительно, пусть про числа $1, 2, 4, \ldots, 2^i$ известно, какие из них принадлежат C, а какие R. Будем вычислять $G(n)$, увеличивая n. До определенного момента $G(n) \leq 2^{i+1}-1$ и информации об этих числах достаточно, чтобы знать, какие значения частые, какие — редкие. Но если $G(n)$ не ограничена, то рано или поздно появится число, большее $2^{i+1}-1$. Однако это число должно быть равно 2^{i+1}, поскольку чтобы выполнялось $\operatorname{mex} X > 2^{i+1}$, необходимо, чтобы $2^{i+1} \in X$, а это невозможно, если каждый элемент X имеет вид $a \oplus b$, где $a, b \leq 2^{i+1} - 1$. Далее для каждого нового числа, большего 2^{i+1} его появление как $\operatorname{mex} X$ возможно лишь если в нем встречается 2^{i+1}, а также все предыдущие. Число же 2^{i+1}, в свою очередь, пока может появиться в X только как $2^{i+1} \oplus 0$. Отсюда можно сделать (не вполне строгий) вывод, что 2^{i+1} должно встречаться достаточно часто, чтобы рано или поздно сложилась ситуация, что X содержит все числа до 2^{i+1}, включительно. Следовательно, если функция Гранди рассматриваемой игры не ограничена, то скорее всего 2^{i+1} является частым значением.

Для игры Гранди эффективные результаты удается получить, считая 1 редким значением, а все остальные степени двойки — частыми. Таким образом, число i является частым тогда и только тогда, когда количество единиц в числе $\lfloor i/2 \rfloor$ нечетно. При таком разбиении среди первых 10^7 значений $G(n)$ лишь 1274 являются редкими (причем, последнее из них — $G(82860) = 108$).

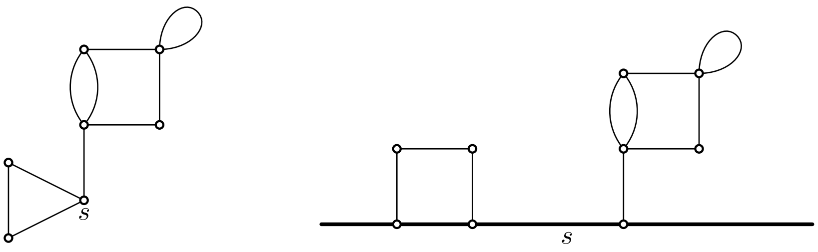
Следующее редкое значение возникает при вычислении $G(47461861) = 261$, а максимальное, известное на данный момент — $G(48399022) = 259$ (см. [9]). Всего известно 1287 редких значений.

Максимальное известное значение функции Гранди игры Гранди — $G(21544358589) = 297$. Наиболее часто встречающееся значения среди первых 2^{35} значений — 128, оно встречается 1806678461 раз. При вычислении 2^{35} значений $G(n)$ в третьей части алгоритма в среднем приходилось использовать 3130 значений $i \in C$.

Отметим, наконец, что приведенный метод можно использовать для ускорения вычисления функции Гранди не только для игры Гранди, но и для других игр типа «бери и разбивай».

Пример 13. Следующий важный пример, который мы рассмотрим — игра Хакенбуш, изобретенная Конвеем. В фундаментальном четырехтомнике [3] авторы используют Хакенбуш как основной инструмент для демонстрации различных приемов анализа комбинаторных игр. Хакенбуш мог бы, в частности заменить ним в роли фундаментальной игры. Версия Хакенбуша, которая нас будет интересовать — справедливый Хакенбуш (или «зеленый Хакенбуш», если следовать терминологии Конвея).

Пусть задан неориентированный граф G и вершина s в нем. Два игрока делают ходы по очереди. Своим ходом игрок может выбрать любое ребро графа и удалить его. Если в результате этого действия часть графа перестает быть связана с вершиной s, то она удаляется. Игрок, который не может сделать ход, поскольку граф состоит из единственной вершины s — проигрывает. При изображении позиций в игре Хакенбуш вершину s часто представляют в виде «земли», на которой располагаются остальные элементы графа.



(а) (б)

Рис. 13. а) Пример позиции в игре Хакенбуш. б) При изображении позиций в игре Хакенбуш вершину s часто представляют в виде «земли», на которой располагаются остальные элементы графа.

Хакенбуш, очевидно, является игрой на графе, где количество позиций есть $O(2^m)$ (m — число ребер в исходном графе, в котором удаляются ребра). Однако, оказывается, с помощью теории прямой суммы можно проанализировать игру Хакенбуш за полиномиальное относительно m время.

Рассмотрим, сначала, несколько упрощенных версий Хакенбуша. Начнем с «Хакенбуша на бамбуке». Пусть граф G имеет вид вершины s, к которой присоединены k путей длины $a_1, a_2, \dots a_k$.

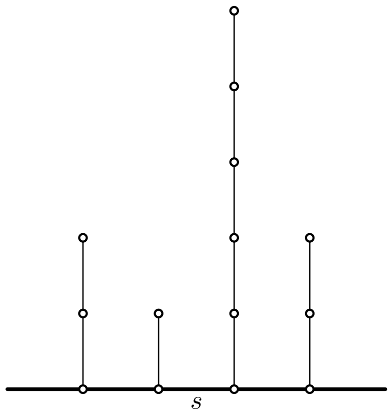


Рис. 14. Хакенбуш на бамбуке

В этом частном случае представление Хакенбуша в виде прямой суммы игр не вызывает сомнений, поэтому достаточно научиться считать функцию Гранди одного «стебля бамбука» длины i. Легко видеть, что ход в таком Хакенбуше представляет собой просто удаление произвольного количества ребер. Заменив ребра на камни, получаем в точности игру ним. Таким образом для Хакенбуша на бамбуке с длинами стеблей $a_1, a_2, \dots a_k$ функция Гранди равна $g = a_1 \oplus a_2 \oplus \dots \oplus a_k$.

Пусть теперь граф в Хакенбуше представляет собой дерево. Будем считать, что дерево подвешено за вершину s.

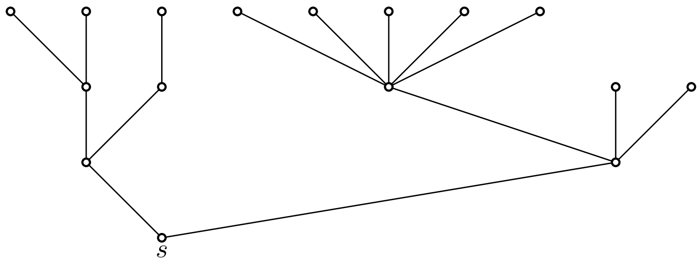


Рис. 15. Хакенбуш на дереве

Докажем, что функция Гранди игры Хакенбуш на дереве T равна

$$G(T) = \bigoplus_i (G(T_i) + 1),$$

где T_i — поддеревья дерева T с вершинами в детях вершины s. Достаточно доказать утверждение для случая одного поддерева, поскольку игры на поддеревьях независимы. Докажем, что если у s один ребенок с поддеревом T_1, то $G(T) = G(T_1) + 1$.

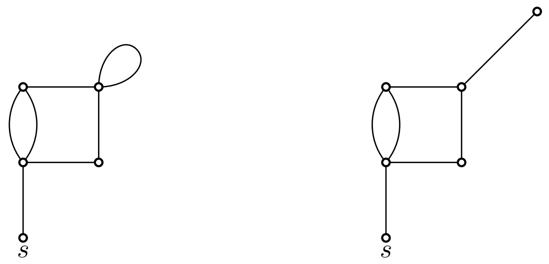
Для этого покажем, что прямая сумма игры на T и игры на бамбуке длиной $G(T_1)+1$ — проигрышная. Доказательство проведем индукцией по числу ребер в T_1, если там нет ребер, то $G(T_1) = 0$, а T — бамбук длины 1, следовательно $G(T) = 1 = G(T_1) + 1$ и условие выполнено. Пусть теперь в T_1 $k > 0$ ребер. Будем играть за второго игрока. Пусть первый игрок сделал ход.

- Если первый игрок сделал ход в бамбуке, оставив 0 ребер, удалим в T ребро, которое ведет из s, у противника нет больше ходов.
- Если первый игрок сделал ход в бамбуке, оставив $i > 0$ ребер, сделаем в T_1 ход, который оставляет поддерево с функцией Гранди $i - 1$ (такой ход возможен, поскольку $i < G(T_1)+1$, следовательно $i-1 < G(T_1)$). По индукционному предположению получившаяся игра — проигрышная.
- Если первый игрок сделал ход в T, удалив ребро, выходящее из s, удалим целиком бамбук, у противника нет больше ходов.
- Если первый игрок сделал ход в T_1, оставив дерево с функцией Гранди $i < G(T_1)$, удалим в бамбуке $G(T_1) - i$ ребер, оставляя $i + 1$ ребро. Получившаяся сумма по индукционному предположению проигрышная.
- Наконец, если первый игрок сделал ход в T_1, оставив дерево с функцией Гранди $i > G(T_1)$, сделаем в T_1 ход обратно в дерево с функцией Гранди i, снова уменьшая количество ребер, и тем самым получая проигрышную по индукционному предположению игру.

Упражнение 11. Проведите альтернативное доказательство формулы для функции Гранди Хакенбуша на дереве, показав, что функция Гранди для всего дерева не меняется, если любое поддерево заменить на другое поддерево с такой же функцией Гранди.

Перейдем теперь к анализу Хакенбуша для произвольного графа. Сделаем сначала первое наблюдение: петля в графе с точки зрения игры эквивалентна висячему ребру, исходящему из этой вершины. Действительно, как удаление петли, так и удаление висячего ребра не меняет остального графа.

Опишем теперь метод как любой граф с циклами можно превратить в дерево, которое имеет такую же функцию Гранди. Рассмотрим две вершины, лежащие на цикле. Объединим их в одну, превратив ребра, соединявшие эти вершины, в петли. Оказывается, получившийся граф будет иметь такую же функцию Гранди, как и исходный граф. Продолжая процесс пока в графе не исчезнут все циклы, получим в результате дерево.



(а) (б)

Рис. 16. Петля эквивалентна висячему ребру.

Опишем идею доказательства, что приведенная операция слияния двух вершин цикла действительно не меняет функцию Гранди графа. Полное доказательство можно прочитать в книге [3]. Доказательство ведется от противного. Пусть найдется граф, слияние двух вершин цикла в котором приводит к графу с другой функцией Гранди. Выберем среди таких граф H с минимальным количеством вершин.

Доказательство проводится в виде серии утверждений, которые мы сформулируем в виде упражнений.

Упражнение 12. Докажите, что в H нет двух вершин, соединенных тремя или более реберно непересекающимися путями.

Указание: пусть в H есть такие две такие вершины, рассмотрим граф H', получающийся в результате слияния этих вершин. Рассмотрите сумму игр на H и на H' и докажите, что она является проигрышной игрой. Используйте тот факт, что во всех графах, которые имеют меньше вершин, чем H, можно выполнять слияние вершин цикла, не меняя функции Гранди.

Из этого следует, что циклы в H не могут иметь общих ребер.

Упражнение 13. Докажите, что в H любой цикл проходит через вершину s.

Из этих двух утверждений следует, что в H всего один цикл, поскольку если бы в H было несколько циклов, то игра на H в явном виде представлялась бы в виде прямой суммы игр на нескольких графах.

Таким образом, H имеет вид цикла, проходящего через начальную вершину s, к каждой вершине которого присоединено дерево. В соответствии с упражнением 11 каждое из этих деревьев можно заменить на стебель бамбука длины, равной функции Гранди этого дерева.

Упражнение 14. Докажите, что если граф имеет вид цикла, проходящего через вершину s, к каждой вершине которого может быть присоединен стебель бамбука некоторой длины, то операция слияния двух вершин цикла не меняет функцию Гранди игры Хакенбуш на этом графе.

На этом анализ игры Хакенбуш завершен.

Приведем еще несколько интересных примеров.

Упражнение 15. Малыш и Карлсон. У Малыша и Карлсона есть торт, имеющий вид прямоугольного параллелепипеда, размером $a \times b \times c$ сантиметров, где a, b и c — целые числа.

Они играют в следующую игру: Малыш и Карлсон делают ходы по очереди. За один ход игрок может разрезать торт параллельно одной из его граней на две неравные части (длины ребер каждой из частей снова должны быть целыми), после чего съесть меньшую часть. Тот, кто не может сделать ход, поскольку торт имеет размер $1 \times 1 \times 1$, проигрывает.

Укажите, как представить игру Малыша и Карлсона в виде прямой суммы трех игр.

Упражнение 16. Шахматы Доусона. Рассмотрим доску $3 \times n$, на первом ряду которой стоят белые пешки, а на последнем — черные пешки. За один ход можно либо подвинуть свою пешку на одну клетку вперед (клетка, на которую перемещается пешка, должна быть свободна), либо побить пешку соперника на одну клетку вперед по диагонали. Если есть возможность побить пешку, то бить обязательно.

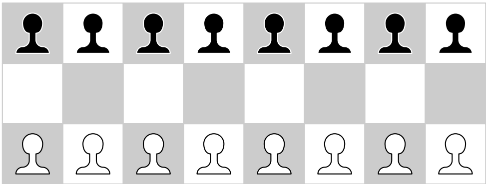


Рис. 17. Шахматы Доусона.

Найдите октальную игру, которой эквивалентна игра в шахматы Доусона (функция Гранди $G(n)$ для этой игры должна совпадать с функцией Гранди шахмат Доусона для доски $3 \times n$).

Упражнение 17. Посчитайте функцию Гранди для игры ним, в которой кучки упорядочены некоторым фиксированным образом, и количество камней в кучках не убывает. Это же свойство должно сохраниться после каждого хода (порядок кучек при этом остается прежним).

Например, из позиции $\langle 2, 3, 3, 5 \rangle$ возможен ход в позицию $\langle 2, 3, 3, 3 \rangle$ или $\langle 0, 3, 3, 5 \rangle$, а в позицию $\langle 2, 3, 3, 2 \rangle$ — нет.

Упражнение 18. Сумму игр на одном и том же графе можно интерпретировать как игру, в которой по графу перемещается не одна фишка, а несколько. За ход игрок должен переместить одну их фишек, при этом несколько фишек могут одновременно находиться в вершине графа. Фрэнкель и Йеша в работе [5] вводят понятие игр с аннигиляцией. Игра протекает таким же образом, но если две фишки оказываются в одной вершине графа, они «аннигилируют» — обе фишки удаляются из графа.

Покажите, что в игре с аннигиляцией на ациклическом графе выигрывает тот же игрок, что и в игре без аннигиляции (обычной прямой сумме игр).

Почему это рассуждение не проходит в случае, если в графе есть циклы?

Пример 14. Ним в поддавки. Несмотря на то, что теория Шпрага-Гранди развита только для нормальных игр, где игрок, который не может сделать ход, выигрывает, ее оказывается возможно применить и для некоторых случаев игр в поддавки. Одним из таких случаев является ним.

Пусть заданы кучки, содержащие $a_1, a_2, \dots a_n$ камней, соответственно. За один ход разрешается из любой кучки взять произвольное число камней. Тот, кто не может взять камень, *выигрывает*.

Рассмотрим сначала игру с одной кучкой. Анализ этой игры как игры на графе показывает, что единственной проигрышной позицией является позиция с одним камнем. Все остальные позиции — выигрышные.

Еще один частный случай — пусть имеются только кучки из одного камня. Тогда, очевидно, позиция является выигрышной тогда и только тогда, когда количество кучек четно.

Теперь можно доказать удивительное утверждение: если в позиции есть хотя бы одна кучка, содержащая более одного камня, то эта позиция является выигрышной в ниме в поддавки тогда и только тогда, когда она является выигрышной в обычном ниме. Докажем это, предъявив выигрышную стратегию за первого игрока в случае, если $a_1 \oplus a_2 \oplus \dots \oplus a_n \neq 0$. А именно, ему следует делать в точности те же ходы, что и в обычном ниме (ходы, ведущие в позиции, где ним-сумма размеров кучек равна нулю), за исключением случая, когда этот ход приводит к тому, что на поле остаются только кучки размера 1. В этом случае вместо этого следует сделать ход, который оставит нечетное количество кучек размера 1. Заметим, что если $a_1 \oplus a_2 \oplus \dots \oplus a_n = 0$ и хотя бы одно из a_i больше единицы, то хотя бы две кучки имеют размер больше единицы, поэтому за один ход сделать все кучки размером 1 не удастся. Так что игрок, делающих ход в такой позиции помешать другому осуществить свой план не может.

К сожалению, приведенное рассуждение годится только для нима. Общая теория комбинаторных игр в поддавки в настоящий момент не построена. Много интересного про игры в поддавки можно найти на сайтах `http://www.plambeck.org/oldhtml/mathematics/games/misere/` [6] и `http://miseregames.org/` [7], а также во втором томе книги [3].

4. Теория Смита для игр на графах с циклами

В случае игр на графах с циклами теория, развитая в предыдущем разделе, оказывается неприменима. Действительно, даже утверждение

$A + A \simeq *0$ не всегда верно, например, для игры *Loop*, изображенной на рис. 18 игра $Loop + Loop$ является ничейной.

Рис. 18. В случае игр с циклами сумма двух одинаковых игр может не являться проигрышной.

В частности, игра *Loop* не эквивалентна игре $*i$ ни для какого i. Это, вообще говоря, не удивительно, поскольку *Loop* — ничейная игра, а $*i$ — выигрышная или проигрышная. Вообще, никакая ничейная игра не может быть эквивалентна игре ним.

Впрочем, некоторые выигрышные игры тоже не эквивалентны игре $*i$ ни для какого i. Пример такой игры *Save* изображен на рис. 19. В игре $Save + *i$ первый игрок всегда может пройти по петле, тем самым отдав противнику ту же игру, следовательно эта игра не может быть проигрышной. Значит *Save* не эквивалентна игре ним.

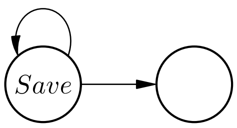


Рис. 19. Игра *Save* является выигрышной. Но сумма ее с любой игрой ним не является проигрышной.

Однако некоторые игры на графах с циклами все же эквивалентны игре ним. Рассмотрим, например, игру A, такую что все ходы из нее возможны только в игры, которые эквивалентны игре ним. Тогда A и сама эквивалентна игре ним.

Лемма 12. Пусть из игры A существуют ходы в игры $A_1, A_2, \ldots, A_n$, каждая из которых эквивалентна ниму. Тогда игра A эквивалентна ниму, причем

$$g(A) = \mathrm{mex}\{g(A_1), g(A_2), \ldots g(A_n)\}.$$

$\triangleright$ Согласно лемме 9 об эквивалентности ниму достаточно показать, что сумма $A + *i$ — проигрышная игра. Это можно сделать также как в теореме 10 (там нигде не использовалась ацикличность графов игр). $\triangleleft$

К сожалению, толку от леммы в такой формулировке мало — с ее помощью можно исследовать только ациклические игры, для любой игры, из начальной вершины которой достижим цикл, без дополнительной информации установить эквивалентность ниму с помощью этой леммы не представляется возможным.

А такие игры есть, например, уже знакомая нам игра, граф которой приведен на рис. 20 с начальной позицией в вершине z. Она проигрышная, и следовательно эквивалентна $*0$, с другой стороны с помощью только леммы 12 этого установить нельзя.

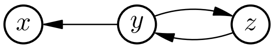


Рис. 20. Вершина z проигрышная. Однако она лежит на цикле $y \to z \to y$, так что анализ этой игры только с помощью леммы 12 провести нельзя.

Чтобы анализировать игры с циклами предыдущую лемму придется усилить следующим образом.

Лемма 13. Пусть из игры A существуют ходы в игры из множества $F(A) = \{A_1, A_2, \ldots, A_n\}$. Обозначим как i минимальное неотрицательное целое число, такое что никакая игра из $F(A)$ не эквивалентна $*i$.

Пусть для каждой игры A_j из множества $F(A)$ выполнено хотя бы одно из следующих утверждений:

1) A_j эквивалентна игре ним;

2) из игры A_j возможен ход в игру, эквивалентную $*i$.

Тогда игра A эквивалентна $*i$.

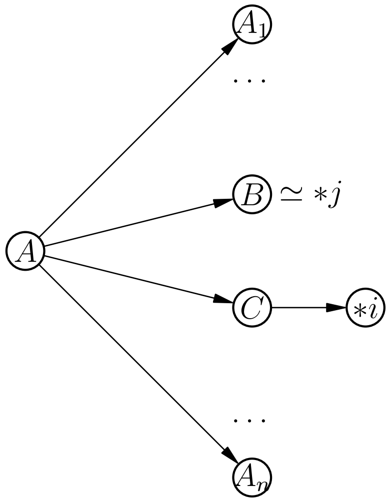
$\triangleright$ Достаточно показать, что $A + *i$ — проигрышная игра.

Если первый игрок делает ход в $*i$, то второй игрок ходит из A в игру с такой же функцией Гранди, и первому игроку достается проигрышная игра.

Если первый игрок делает ход в A в игру, эквивалентную игре ним, то второй игрок получает выигрышную игру (сумма двух различных игр ним).

Наконец, если первый игрок делает ход из A в игру, не эквивалентную игре ним, то в силу условия леммы второй игрок может ответить в ней в игру, эквивалентную $*i$, и первый игрок снова получает проигрышную игру.

Таким образом, при любом своем ходе первый игрок проигрывает и следовательно игра $A + *i$ проигрышная. $\triangleleft$



Обозначим как i минимальное неотрицательное целое число, не являющееся функцией Гранди игры, в которую из A возможен ход. Если из игры A есть ход в некоторую другую игру, то она должна быть либо эквивалентна ниму (как B на этом рисунке), либо из нее должен быть ход в игру, эквивалентную $*i$ (как из игры C на этом рисунке).

Рис. 21. Эквивалентность ниму в случае наличия циклов.

Отметим также важный момент: лемму 13 можно применять даже, если для некоторых игр из $F(A)$ неизвестно, эквивалентны ли они ниму, и если да, то какому.

Действительно, пусть i — минимальное целое неотрицательное число, не встречающееся среди функций Гранди тех игр, в которые возможен ход из A, и про которые достоверно известно, что они эквивалентны ниму. Если для всех остальных игр из $F(A)$ выполнен второй вариант, то есть из них возможен ход в игру, эквивалентную $*i$, то ясно, что они не могут быть эквивалентны $*i$, и следовательно i на самом деле является минимальным целым числом, не равным функции Гранди *никакой* из игр, в которые возможен ход из A. А значит можно применить лемму.

Рассмотрим в качестве примера применение последней леммы для анализа игры на рис. 20. Сначала заметим, что из вершины x нет возможных ходов, поэтому она эквивалентна $*0$. Теперь обратим внимание на вершину z. Из нее есть единственный ход в вершину y, эквивалентность которой ниму пока не установлена. Заметим, однако, что для вершины z выполнено утверждение леммы 13 с $i = 0$. Следовательно $z \simeq *0$. Наконец, теперь все вершины, в которые можно сделать ход из y, имеют функцию Гранди равную нулю, следовательно $y \simeq *1$.

Оказывается, лемма 13 в некотором смысле описывает все игры, эквивалентные игре ним. А именно, пусть задан граф G. Найти все вершины в этом графе, игры, соответствующие которым, эквивалентны игре ним, можно с использованием следующего алгоритма.

Будем помечать позиции эквивалентные игре ним, их числами Гранди. Исходно ни одна позиция не помечена. Если в графе существует позиция, для которой, исходя из уже помеченных позиций, можно с использованием леммы 13 сделать вывод, что она эквивалентна некоторой игре $*i$, то пометим ее числом i. Если же таких позиций нет, то ни одна из непомеченных позиций не эквивалентна игре $*i$ ни для какого i.

Предложение 14. Приведенный алгоритм корректно помечает все позиции графа, эквивалентные игре ним, а все непомеченные после его выполнения вершины являются неопределенными.

$\triangleright$ Корректность всех поставленных пометок легко доказать индукцией по номеру шага, на котором она сделана. Докажем, что все позиции в графе, эквивалентные ниму, были помечены.

Проведем доказательство от противного. Пусть есть позиции, эквивалентные ниму, но не помеченные нашим алгоритмом. Выберем среди них ту позицию u, что для соответствующей ей игре G_u выполнено соотношение $G_u \simeq *i$ для некоторого i, и количество ходов до проигрыша в игре $G_u + *i$ минимально среди всех непомеченных позиций.

Ясно, что это количество больше нуля, поскольку если из игры G_u нет возможных ходов, то алгоритм пометит ее как эквивалентную $*0$.

Пусть Z — множество пометок вершин, в которые есть ход из вершины u, и $j = \operatorname{mex} Z$. Рассмотрим три варианта.

Если $i < j$, то $G_u + *i$ — выигрышная позиция (достаточно пойти в позицию, помеченную i, противнику достанется игра $*i + *i$).

Если $i > j$, то сделаем в игре $*i$ ход в игру $*j$. Тогда противник, чтобы выиграть, должен сделать ход в игре G_u (поскольку если он сделает из $*j$ ход в некоторую $*k$, он не сможет получить проигрышную позицию, так как $G_u + *i$ и $G_u + *k$ не могут быть одновременно проигрышными для $i \neq k$). Но он не может пойти в позицию, помеченную числом j, следовательно он должен сделать ход в непомеченную позицию. Получаем пару из непомеченной позиции и $*j$, которая по предположению является проигрышной, причем за меньшее число ходов, чем $G_u + *i$, что противоречит выбору u.

Наконец, если $i = j$, то сделаем ход в непомеченную позицию v, из которой нет хода в позицию, помеченную числом j (такая есть, иначе G_u удовлетворяла бы условиям леммы 13 и была бы помечена алгоритмом). Но тогда противник вынужден либо делать ход из G_v в непомеченную позицию (иначе нам достанется выигрышная игра), либо делать ход в игре $*j$. В любом случае получаем пару из непомеченной позиции и игры ним, которая должна быть проигрышной за меньшее число ходов, чем $G_u + *i$, противоречие.

Таким образом, ни одна непомеченная позиция не может быть эквивалентна игре ним, что и требовалось. $\triangleleft$

Будем называть игры, которые не эквивалентны игре ним, *неопределенными.* Чтобы полностью охарактеризовать неопределенные игры, нам потребуется следующая лемма. Она показывает, что только игры, эквивалентные ниму, могут давать при сложении проигрышные игры.

Лемма 15. Пусть для игры A существует игра B, такая что $A+B \simeq 0$. Тогда $A \simeq *i$ для некоторого i.

$\triangleright$ Мы проведем доказательство по индукции. Для каждой игры A, удовлетворяющей условию леммы, найдется такая игра B, что проигрыш в игре $A+B$ происходит за минимальное количество ходов. Будем рассматривать эту игру B и проведем индукцию по количеству ходов до проигрыша в $A+B$.

Если $A+B$ — проигрышная за 0 ходов, то в A нельзя сделать ни одного хода, следовательно она и сама проигрышная и эквивалентна $*0$.

Пусть теперь для всех игр A', для которых найдется B', что в $A'+B'$ проигрыш происходит за меньшее число ходов, чем в $A+B$, утверждение теоремы доказано. Пусть из игры A возможны ходы в игры из множества $F(A) = \{A_1, A_2, \ldots, A_n\}$. Рассмотрим два варианта.

Если найдется такая игра A_j, что при нашем ходе из A в A_j противник, чтобы выиграть за минимальное число ходов, вынужден ответить в A_j, то после его хода нам достается некоторая игра $A'_j + B$, в которой мы проигрываем за меньшее число ходов, чем в $A+B$. Значит, по индукционному предположению, $A'_j \simeq *i$ для некоторого i, а значит $B \simeq *i$ и $A \simeq *i$.

Если же при любом нашем ходе в A противник может ответить в B и по прежнему выиграть за минимальное число ходов, то это значит что для каждой A_j найдется такая игра B_j, в которую есть ход из игры B, что $A_j + B_j$ — проигрышная игра, причем проигрыш в ней происходит за меньшее число ходов, чем в $A+B$. Значит A_j эквивалентно некоторому $*i_j$ для всех j и игра A тоже эквивалентна некоторой игре ним по лемме 12. $\triangleleft$

Следствие 16. Если A — неопределенная игра, то для любой игры B игра $A+B$ является либо выигрышной, либо ничейной. В частности, сама игра A является либо выигрышной, либо ничейной.

Следствие 17. Если A и B — неопределенные игры, то игра $A+B$ является ничейной.

$\triangleright$ Игра $A+B$ не может быть проигрышной.

Если она выигрышная, то из нее есть ход в проигрышную игру. Однако после хода в $A+B$, получившаяся игра по прежнему представляет собой сумму двух игр, одна из которых неопределенная, поэтому она не может быть проигрышной.

Таким образом, игра $A + B$ должна быть ничейной. $\lhd$

Итак, сумма неопределенных игр всегда является ничейной. Следовательно неопределенная игра A с точки зрения эквивалентности по Гранди характеризуется в точности множеством ним-игр, в сумме с которыми она становится выигрышной.

Пусть $K = \{k_1, k_2, \ldots, k_n\}$ — конечное множество целых неотрицательных чисел. Обозначим как ∞_K игру, из которой возможны ходы в игры из множества $\{*k_1, *k_2, \ldots, *k_n, \infty_K\}$. Игра ∞_K является выигрышной если $0 \in K$ и ничейной в противном случае.

Для множества $K = \{k_1, k_2, \ldots, k_n\}$ и целого неотрицательного числа i обозначим как $K \oplus i$ множество $\{k_1 \oplus i, k_2 \oplus i, \ldots, k_n \oplus i\}$.

Следующая теорема завершает классификацию игр на графах с точки зрения эквивалентности по Гранди.

Теорема 18. (Смит, Фрэнкель) Пусть A — игра на графе. Тогда A либо эквивалентна игре $*i$ для некоторого $*i$, либо эквивалентна игре ∞_K для некоторого конечного множества целых неотрицательных чисел K.

При этом:

- $*i + *j \simeq *(i \oplus j)$;
- $*i + \infty_K \simeq \infty_{K \oplus i}$;
- $\infty_K + \infty_L \simeq \infty_\varnothing$.

Упражнение 19. Докажите теорему 18.

Упражнение 20. Разработайте алгоритм, который для каждой вершины u заданного графа G находит игру ним либо игру вида ∞_K, эквивалентную игре на G с начальной вершиной u.

Заключение

Мы рассмотрели базовые алгоритмы для анализа игр на графах, сопроводив их некоторыми примерами использования. К сожалению, в русскоязычной литературе эта тема освещена достаточно мало. К счастью, развитие интернета, и, в частности, интернет-магазинов и интернет-библиотек, дает доступ в том числе к зарубежной литературе, никогда не издававшейся на русском языке. Заинтересовавшимся данной темой можно порекомендовать четырехтомник [3], в котором рассмотрены сотни конкретных комбинаторных игр, а на примере таких фундаментальных игр как ним и Хакенбуш показаны общие методы анализа комбинаторных игр.

Мы совсем не затронули анализ прямой суммы несправедливых игр, а эта теория, хоть и не имеющая таких же простых алгоритмических методов, как теория Шпрага-Гранди, также является очень красивой и интересной. Анализ игр в поддавки в настоящий момент переживает бурное развитие, последние результаты можно найти на сайте `http://miseregames.org/`. Так что теория комбинаторных игр остается интересной областью с множеством открытых задач, которые ждут своего решения.

Список литературы

[1] *Кормен Т., Лейзерсон Ч., Р. Р.* Алгоритмы: Построение и Анализ. — М.: МЦНМО, 1999.

[2] *Куммер Б.* Игры на графах: Пер. с нем. — М.: Мир, 1982.

[3] *Berlekamp E., Conway J. H., Guy R. K.* Winning Ways for Your Mathematical Plays (4 vols.). — 2 edition. — A. K. Peters, 2000.

[4] *Conway J.* On Numbers and Games. — Academic Press Inc. (London), 1976.

[5] *Fraenkel A. S., Yesha Y.* Theory of annihilation games // *J. Combin. Theory (Ser. B)*. — 1982. — Vol. 33. — Pp. 60–86.

[6] Misere games. `http://www.plambeck.org/oldhtml/mathematics/games/misere/`.

[7] Misere games. `http://miseregames.org/`.

[8] Periods in taking and splitting games / I. Caines, C. Gates, R. Guy, R. Nowakowski // *The American Mathematical Monthly*. — 1999. — Vol. 106, no. 4. — Pp. 359–361.

[9] Sprague-grundy values of grundy's game. `http://wwwhomes.uni-bielefeld.de/achim/grundy.html`.

[10] Sprague-grundy values of octal-games. `http://wwwhomes.uni-bielefeld.de/achim/octal.html`.