Последовательные игры

Вопросы

1. Понятие последовательной игры

> Изучите понятие последовательной игры. Какие из известных вам игр являются последовательными? Являются ли последовательными матричные игры? Могут ли быть последовательными игры, в которых нужно бросать кубик или имеется другой элемент случайности?

◆ Главным элементом последовательной игры является её состояние. Игра начинается с некоторого фиксированного значения этого состояния. В некоторой последовательности (которая может определяться в процессе игры) игроки делают ходы — изменяют состояние игры. Множество возможных ходов игрока может быть ограничено. Существует множество терминальных значений состояния, при достижении любого из которых игра заканчивается и игроки получают выигрыши в соответствии с этим значением. Если состояние не терминальное, то игрок, чья очередь делать ход, обязан его сделать.

В последовательной игре стратегия игрока — это функция состояния игры, значение которой — ход, который следует сделать игроку в этом состоянии.

◆ Примеры последовательных игр:

Игра	Состояние	Ходы	Терминальность состояния	Кто выигрывает	Ничья
Крестики-Нолики	Расположение крестиков и ноликов на поле 3×3	Поставить свой знак на незанятое поле	Поставлено три одинаковых знака в ряд	Поставивший три знака в ряд	Возможна
Шахматы	Расположение белых и чёрных фигур по поле 8×8	Передвинуть фигуру согласно правилам	Королю любого из игроков поставлен мат, или достигнута ничья	Поставивший королю противника мат	Возможна
Ним	Набор кучек из предметов	Взять некоторое кол-во предметов из любой кучки	Кучки закончились	Нормальна игра: сделавший последний ход Мизер: сделавший предпоследний ход	Невозможна

◆ Матричные игры не являются непосредственно последовательными, поскольку в них нет изменяющегося состояния, и игроки делают ходы одновременно, а не последовательно. Однако матричная игра может быть представлена в виде последовательной, в которой игроки делают по одному ходу (в любом порядке), но ходы предыдущих игроков не известны последующим. Т.е. такая игра не будет игрой с совершенной информацией.

◆ Игры с элементами случайности могут быть последовательными. Пример такой игры — покер: случайным здесь является начальное расположение карт в колоде. Кроме того покер — игра с несовершенной информацией.

2. Дерево игры

> Изучите понятие дерева игры. Любую ли игру можно задать деревом? Вся ли информация об игре хранится в дереве? В каких случаях целесообразнее использовать направленный ациклический граф вместо дерева?

Дерево игры — дерево, вершины которого — состояния игры, а рёбра — возможные ходы игроков. Корень дерева соответствует начальному состоянию игры, листья — конечным состояниям. С каждым листом ассоциированы выигрыши игроков в том случае, если игра придёт в соответствующее листу состояние.

Теоретически любую игру можно представить в виде дерева, однако оно может иметь следующие особенности:

- Оно может иметь бесконечную глубину (по крайней мере по некоторым ветвям)
  - Пример. Шахматы без правил 50 ходов и троекратного повторения.
- Оно может иметь бесконечное (или даже несчётное) кол-во вариантов хода в некоторых вершинах
  - Пример. Модель Штакельберга.

Непосредственно в дереве храниться только структура ходов игроков, основными характеристиками которой являются глубина и коэффициент ветвления. Кроме них есть:

- Ассоциация каждой промежуточной вершины с одним из игроков: данный игрок принимает решение и делает ход в этой вершине. Кроме того вершина может быть ассоциирована с псевдоигроком, Природой.
- Ассоциация с каждой вершиной, в которой играет Природа, распределения вероятностей по дочерним вершинам.
- Ассоциация каждой конечной вершины с кортежем выигрышей игроков.
- (Опционально) Деление множества вершин каждого из игроков на информационные множества.

Направленный ациклический граф целесообразнее использовать вместо дерева, если некоторые последовательности ходов игроков могут привести к одному и тому же её состоянию: использование НАГа сокращает структурное представление игры, а значит и временны́е ресурсы, требуемые на её просчёт.

3. Стратегия по дереву игры

> Предложите способ вычисления беспроигрышной стратегии, если известно полное дерево игры. Найдите такую стратегию для игры крестики-нолики на доске 3x3.

Искомый способ вычисления беспроигрышной стратегии — обратная рекурсия.

Начиная с нижних уровней дерева и поднимаясь к корню, подсчитаем для каждой вершины дерева две следующие характеристики:

- выигрыши игроков, которые будут достигнуты при оптимальной игре каждого из них;
- оптимальный ход для игрока, который делает ход в данной вершине.

Для листьев выигрыши задаются явно, а оптимальный ход неопределён, поскольку игра закончена.

Для каждой промежуточной вершины А оптимальный ход — в такую вершину Б, которая максимизирует выигрыш игрока, делающего ход в А (этот выигрыш уже известен, поскольку Б располагается ниже А). Выигрыш игроков в А будет равен их выигрышу в Б.

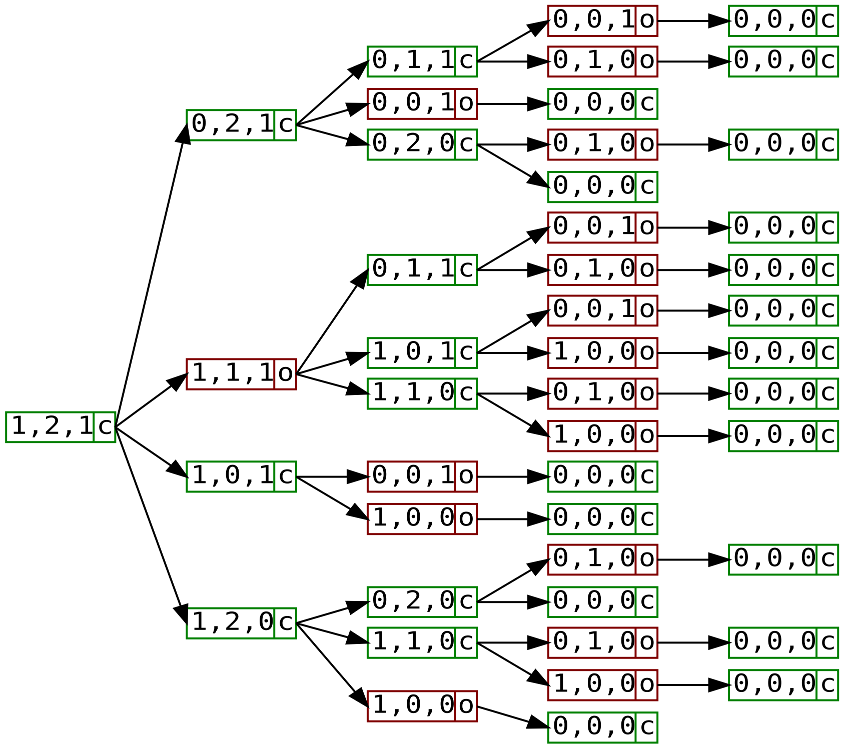
В итоге будут подсчитаны оптимальные ходы для всех вершин дерева, в том числе и для корня. Беспроигрышная стратегия (или, как минимум, дающая максимально возможный гарантированный выигрыш) — при попаданию в любую вершину совершать подсчитанный оптимальный ход для неё.

См. output/ttt.svg. (См. также https://xkcd.com/832/.)

4. Ним

> Постройте дерево для игры Ним. Какими свойствами обладает дерево этой игры? Предложите способ нахождения оптимальной стратегии для игры Ним. Изучите понятие функции Шпрага-Гранди. Как эта функция помогает найти беспроигрышную стратегию? В каких случаях ходящий первым игрок гарантированного выигрывает при оптимальной игре?

◆ Дерево выигрышных (зелёные) и проигрышных (красные) позиций для Нима с тремя кучками из 1, 2 и 1 объектов при условии, что сделавший последний ход игрок проигрывает (мизер):



Легенда:

- Левая часть узла — список количеств предметов в кучках
- Правая часть узла — выигрывающий игрок:
  - с — текущий
  - о — оппонент

◆ Свойства дерева:

- Выигрышность-проигрышность позиции не зависит от её предыстории, поэтому все узлы, обозначающие одну и ту же позицию, могут быть объединены.
- У позиций с одинаковым мультимножеством количеств предметов в кучках поддеревья изоморфны — эти позиции симметричны.

◆ Построив дерево игры, к нему можно применить обратную рекурсию для вычисления оптимальных ходов в каждой вершине и оптимальной стратегии в целом. Кроме того оптимальные ходы могут быть определены через подсчёт двоичного исключающего «или» от размеров кучек.[1]

◆

Функция Шпрага-Гранди[2]

Функция Шпрага-Гранди определена на позициях последовательной игры следующим образом:

$$F(x)=min\{n\in\mathbb{N}_0 : n\neq F(y), y\in G(x)\} ,$$

где $G(x)$ — множество позиций, в которые можно перейти из позиции x за один ход.

Из определения следуют следующие свойства:

- $F(x)=0$ равносильно тому факту, что все возможные ходы из x ведут в такие позиции y, что $F(y)\neq 0$.
  - В частности если x — конечная позиция, то $F(x)=0$.
- $F(x)\neq 0$ равносильно тому факту, что из позиции x существует ход в такую позицию y, что $F(y)=0$.

1 https://ru.wikipedia.org/wiki/Ним_(игра)#Стратегия_игры
2 https://ru.wikipedia.org/wiki/Функция_Шпрага_—_Гранди

Выигрышные и проигрышные позиции

Одно из полезных свойств функции Шпрага-Гранди заключается в том, что она равна нулю для всех проигрышных позиций и положительна для всех выигрышных позиций. Это даёт метод нахождения выигрышной стратегии:

1. Найти функцию Шпрага-Гранди, например, строя её рекуррентно, начиная с конечных позиций.
2. Если в начальной позиции функция Гранди равна нулю, то игра для первого игрока проигрышна.
3. В противном случае, первый игрок может выиграть, перемещаясь каждым ходом в позицию с нулевым значением функции Гранди.

Сумма игр

Если у нас имеется *n* игр G_1, G_2, ..., G_n, то можно рассмотреть комбинацию этих игр, для которой игровое поле состоит из совокупности игровых полей для игр G_1, G_2, ..., G_n и за один ход игрок может выбрать некоторое i, $1 \le i \le n$, и сделать ход на игровом поле для игры G_i. Такая комбинация называется суммой игр G_1, G_2, ..., G_n и обозначается $G_1 + G_2 + ... + G_n$. Ситуацию на игровом поле игры $G_1 + G_2 + ... + G_n$, когда игровое поле игры G_i находится в позиции P_i, удобно обозначать как $(P_1, P_2, ..., P_n)$.

Функция Шпрага-Гранди обладает следующим свойством, которое позволяет оптимально играть в сумму игр $G_1 + G_2 + ... + G_n$, зная функцию Шпрага-Гранди для всех позиций каждой из игр G_i:

$$F(P_1,\dots,P_n)=F(P_1)\oplus\cdots\oplus F(P_n),$$

где $\oplus$ — исключающее «или».

Данное свойство функции обобщает вышеприведённую стратегию для игры Ним.

5. Общий алгоритм

Реализуйте алгоритм вычисления оптимальной стратегии в последовательной игре. Создайте класс, представляющий состояние игрового поля. Напишите метод, возвращающий все возможные ходы для данного состояния поля. Напишите метод, возвращающий выигрыш игроков для данного состояния поля. Создайте хеш-таблицу для хранения ожидаемого выигрыша для уже рассмотренных состояний поля, эта таблица поможет заметно ускорить расчет, если одно состояние игры может получиться при разных цепочках ходов. Заполните эту таблицу, обходя каким-либо способом дерево игры. Напишите функцию, выдающую оптимальный ход для данного состояния игры, используя построенную таблицу.

general.hxx

6. Алгоритм, учитывающий симметрии

Модифицируйте алгоритм поиска беспроигрышной стратегии, учитывая симметрии игры. Два состояния называются симметричными, если существует преобразование игрового поля, переводящее дерево подигры для одного состояния в дерево подигры второго. Напишите метод, возвращающий для данного состояния все его симметричные. Сохраняйте в хеш-таблицу только одного представителя из семейства симметричных состояний. При обходе дерева игры делайте обход дерева подигры только для одного из симметричных состояний.

symmetries.hxx

8. Учёт количества ходов

Какие модификации в алгоритм необходимо ввести, чтобы программа выигрывала за наименьшее возможное число ходов?

В дополнение к выигрышам игроков в хэш-таблице нужно хранить ожидаемое кол-во ходов, которое пройдёт от заданной позиции до конца игры в соответствии с оптимальной стратегией. Для конечных состояний это величина равна 0, а для промежуточных она на 1 больше, чем эта величина в состоянии в которое следует сделать ход. Если при выборе оптимального хода есть несколько вариантов с одинаковым выигрышем, то нужно выбирать тот из них, для которого ожидаемое кол-во ходов до конца игры минимально.

7. Эвристика

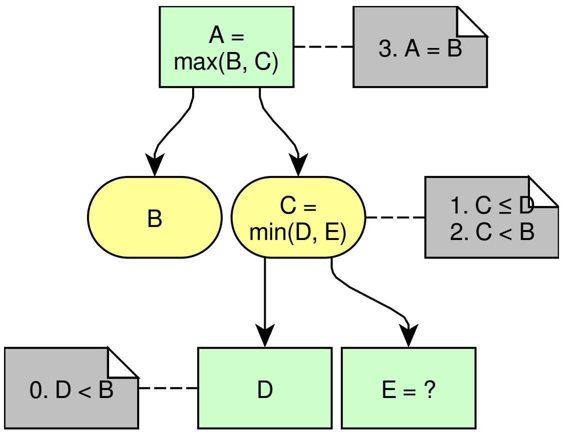
> Какие ходы будет делать алгоритм, если гарантировано выиграть невозможно? Как увеличить шансы на выигрыш?

- Во всех случаях алгоритм будет выбирать последний из списка возможных ходов, который приводит к строгому увеличению выигрыша игрока, делающего ход.
- Если мы допускаем, что противник может ошибиться (сделать не оптимальный для себя ход), то в проигрышных для нас позициях нам следует выбирать ходы, приводящие к наиболее продолжительной игре, в надежде, что противник ошибётся и мы попадём в выигрышную для нас позицию.

9. Альфа-бета отсечение

> Нужно ли делать обход всего дерева игры или можно обойти только его части не уменьшая шансы на выигрыш? Как можно оптимизировать алгоритм?

- Иногда необязательно обходить всё дерево для вычисления оптимальной стратегии. В качества примера такой ситуации рассмотрим часть дерева игры с нулевой суммой, изображённую на рисунке справа.



Игрок 1 делает ходы в вершинах A, D и E, а игрок 2 — B и C. При этом уже посчитаны ожидаемые выигрыши игрока 1 в вершинах B и D, и оказалось, что D < B. В вершине E выигрыш ещё не посчитан. Игрок 2 будет пытаться минимизировать выигрыш игрока 1 в вершине C, поэтому C = min(D, E), а значит C ≤ D < B. Игрок 1 будет пытаться максимизировать свой выигрыш в вершине A, поэтому A = max(B, C) = B, поскольку C < B. Таким образом, мы смогли посчитать ответ для вершины A, не считая ответ для вершины E.

- Формализацией данного наблюдения является алгоритм [альфа-бета отсечения](), позволяющий оптимизировать поиск для игр с нулевой суммой. Он заключается в поддержании при обходе дерева двух следующих характеристик:
  - α — текущая максимальная нижняя граница выигрыша игрока 1;
  - β — текущая минимальная верхняя граница выигрыша игрока 1.

Если при обходе детей некоторой вершины B значение игры у текущего её ребёнка таково, что понятно, что оптимальный ход игры не зайдёт в B, то оставшиеся дети этой вершины не перебираются.

Выбор подходящего порядка обхода детей может привести к более быстрому отсечению неперспективных ветвей. В частности, имеет смысл вначале рассматривать такие ходы, которые приводят к наибольшему улучшению эвристической функции оценки состояния в рассматриваемой вершине или привели к улучшению расчётного значения состояния в уже рассмотренных вершинах.

См. ab/ab.cxx.

10. Крестики-нолики на доске 4x4

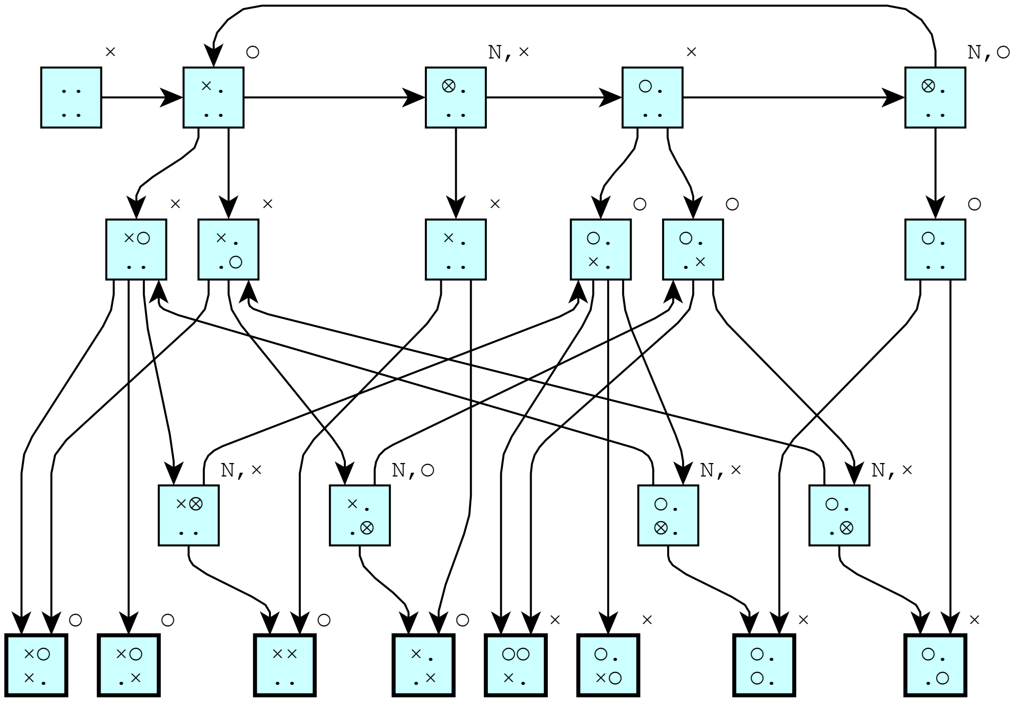
> Напишите программу, играющую в крестики-нолики оптимальным образом на доске 4x4 (выигрывает поставивший 4 значка в ряд).

ttt/4x4-game/4x4-game.cxx

11. Вероятностные крестики-нолики

> Модифицируем правила игры в крестики-нолики, разрешив ходить на клетки, уже занятые противником, при этом с вероятностью 1/2 знак противника будет заменен на ваш, с вероятностью 1/2 знак на поле сохранится. Какие модификации нужно внести в вышеописанный алгоритм, чтобы он был применим к этой игре? Реализуйте алгоритм, играющий оптимальным образом по этим правилам для досок 3x3 и 4x4.

Первая особенность такого варианта крестиков-ноликов — дерево игры становиться бесконечно глубоким, поскольку появляются циклы между состояниями игры. Вот граф (который уже более не является ни деревом, ни даже НАГом) для подобных крестиков-ноликов на доске 2x2:



Для учёта данной особенности исходный алгоритм можно модифицировать двумя способами.

Способ №1. Считать состоянием игры не только поле со знаками, но и номер текущего хода. Граф подобных состояний будет бесконечно глубоким деревом, а значит его нельзя будет просчитать исходным алгоритмом: он превысит предел рекурсии. В качестве приближения можно ограничить поиск решения некоторой глубиной, при достижении которой считать выигрыши игроков через некоторую эвристику, например количество рядов, которые они потенциально могут заполнить своим знаком. Кроме того выигрыш игрока может зависеть от того, на каком ходу он выиграл.

Способ №2. Как и в оригинальной варианте игры считать состоянием только поле со знаками. Граф состояний будет конечный, но циклический. Это также не позволяет использовать исходный алгоритм: он зациклится. Здесь нужен принципиально другой способ расчёта выигрышей в вершинах, лежащих в циклах. Например, через составление системы уравнений, связывающей эти выигрыши между собой и с выигрышами в вершинах вне циклов.

Вторая особенность этого варианта игры — в игру добавляется 3-й игрок — Природа, — который делает ходы после таких ходов обычных игроков, в которых они пытаются «перезаписать» знак противника. Просчёт ходов данного игрока будет заключаться не в выборе максимального выигрыша (который для него и не определён) среди дочерних позиций, а в усреднении этих выигрышей в них по обоим игрокам.

Вследствие первой из особенностей реализация алгоритма решения достаточно нетривиальна. Рассмотрим другую модификацию оригинальной игры, в которой Природа делает ходы по очереди вместе с обычными игроками, и заключаются они в «закрашивании» случайных клеток поля — они становятся недоступными для обычных игроков.

См. ttt/chance/chance.cxx.