О компьютеризированных доказательствах в теории расписаний

Продолжение

Черных И.Д.

Семинар “Модели и алгоритмы
для задач составления расписаний”
8.04.2023

*Исследование выполнено при поддержке
гранта РНФ № 22-71-10015*

Проблема четырех красок

Определение

Назовем картой некоторое разбиение плоскости на связные области. Области назовем соседними, если у них есть общий участок границы ненулевой протяженности.

Проблема четырех красок

Определение

Назовем картой некоторое разбиение плоскости на связные области. Области назовем соседними, если у них есть общий участок границы ненулевой протяженности.

Определение

Раскраску областей карты (каждую в какой-то цвет) назовём правильной, если никакие две соседние области не раскрашены одним цветом.

Проблема четырех красок

Определение

Назовем картой некоторое разбиение плоскости на связные области. Области назовем соседними, если у них есть общий участок границы ненулевой протяженности.

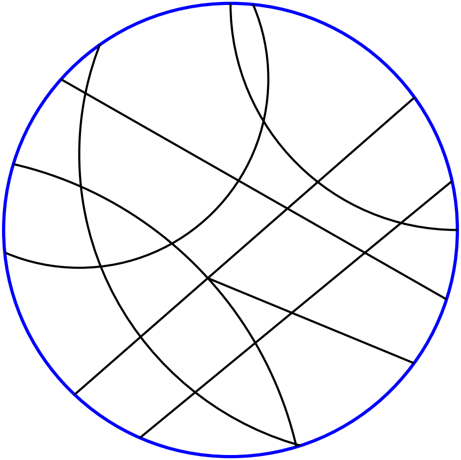
Определение

Раскраску областей карты (каждую в какой-то цвет) назовём правильной, если никакие две соседние области не раскрашены одним цветом.

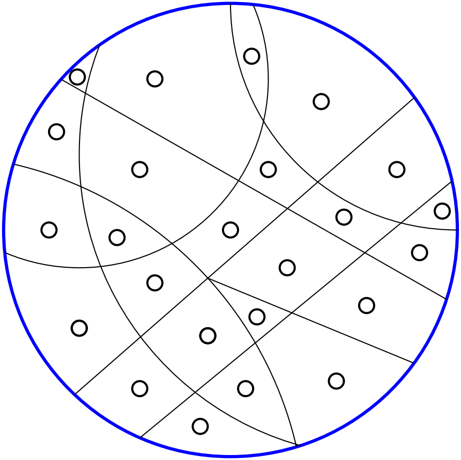
Вопрос

Какого минимального количества цветов гарантированно хватит для правильной раскраски любой карты?

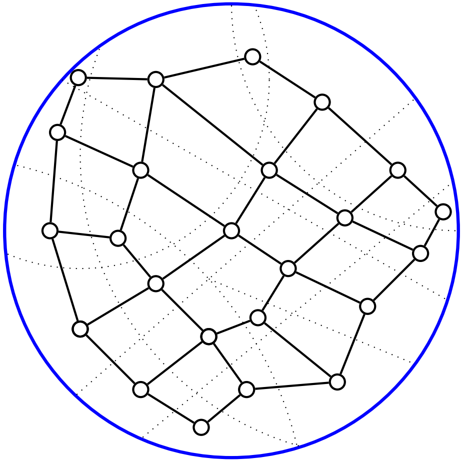
Раскраска карты



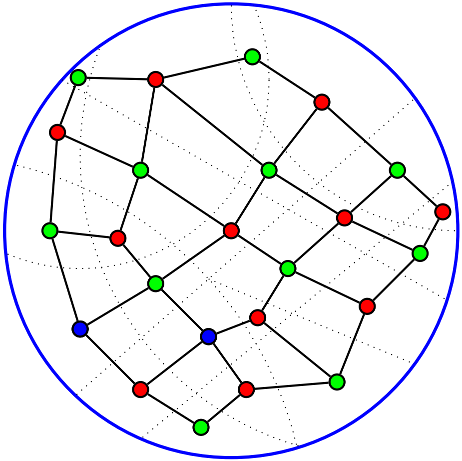
Раскраска карты



Раскраска карты



Раскраска карты



Определение

Степенью вершины простого графа называется число смежных (соседних) с ней вершин.

Формула Эйлера

Пусть v — число вершин плоской карты, e — число ее ребер, а f — число граней (включая одну неограниченную). Тогда

$$v + f = e + 2.$$

Теорема о пяти красках

Определение

Степенью вершины простого графа называется число смежных (соседних) с ней вершин.

Формула Эйлера

Пусть v — число вершин плоской карты, e — число ее ребер, а f — число граней (включая одну неограниченную). Тогда

$$v + f = e + 2.$$

Следствие

Любой планарный граф содержит вершину степени не больше 5.

Теорема о пяти красках

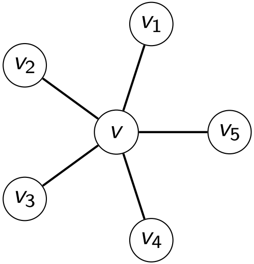
Теорема

Вершины любого планарного графа можно правильно раскрасить в пять цветов.

Теорема о пяти красках

Теорема

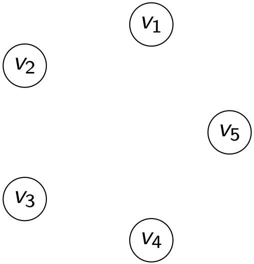
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

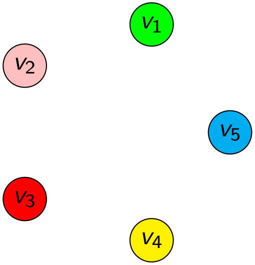
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

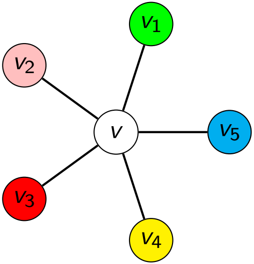
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

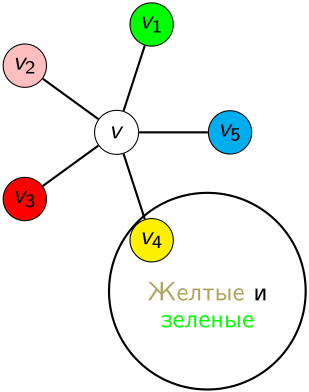
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

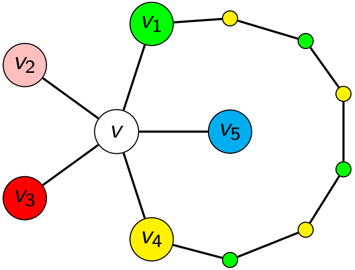
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

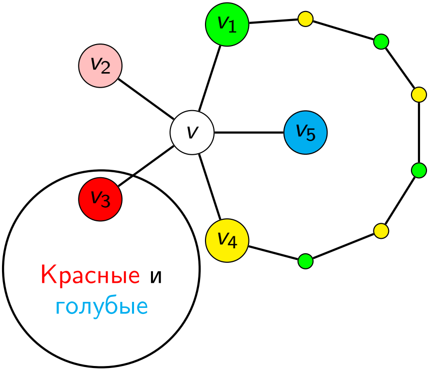
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



Теорема о пяти красках

Теорема

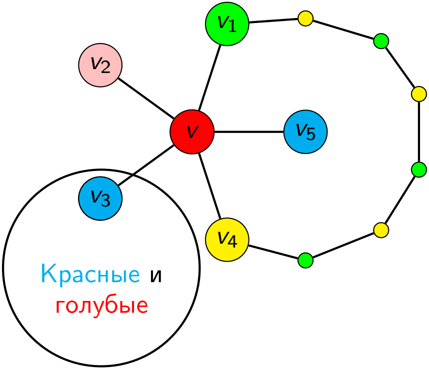
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



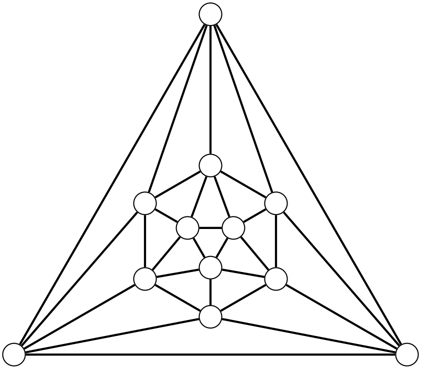
Теорема о пяти красках

Теорема

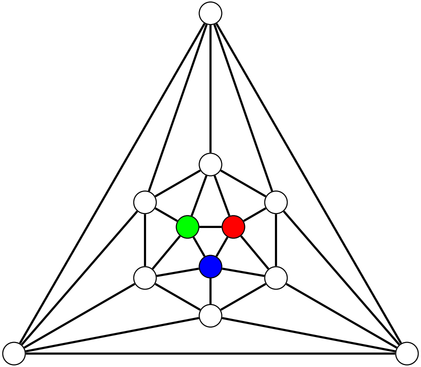
Вершины любого планарного графа можно правильно раскрасить в пять цветов.



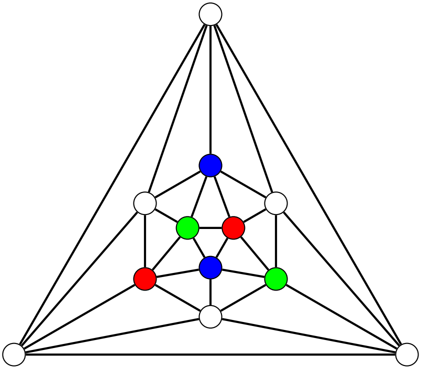
Пример планарного графа со степенями > 4



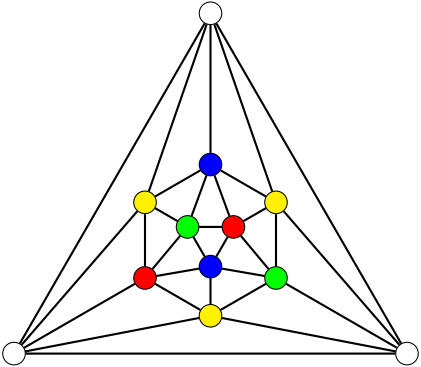
Пример планарного графа со степенями > 4



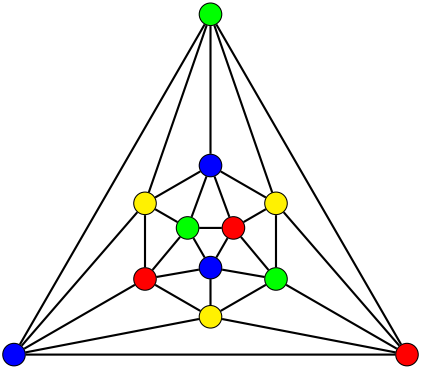
Пример планарного графа со степенями > 4



Пример планарного графа со степенями > 4



Пример планарного графа со степенями > 4



Теорема о четырех красках

Теорема

Для любого планарного графа существует правильная раскраска вершин в четыре цвета.

Теорема о четырех красках

Теорема

Для любого планарного графа существует правильная раскраска вершин в четыре цвета.

- K. Appel, W. Haken (1977), "Every Planar Map is Four Colorable. I. Discharging", Illinois Journal of Mathematics 21 (3): 429–490.
- K. Appel, W. Haken, J. Koch (1977), "Every Planar Map is Four Colorable. II. Reducibility", Illinois Journal of Mathematics 21 (3): 491–567.
- K. Appel, W. Haken (October 1977), "Solution of the Four Color Map Problem", Scientific American 237 (4): 108–121.

Теорема

Для любого планарного графа существует правильная раскраска вершин в четыре цвета.

- K. Appel, W. Haken (1977), "Every Planar Map is Four Colorable. I. Discharging", Illinois Journal of Mathematics 21 (3): 429–490.
- K. Appel, W. Haken, J. Koch (1977), "Every Planar Map is Four Colorable. II. Reducibility", Illinois Journal of Mathematics 21 (3): 491–567.
- K. Appel, W. Haken (October 1977), "Solution of the Four Color Map Problem", Scientific American 237 (4): 108–121.
- K. Appel, W. Haken (1989), "Every Planar Map is Four-Colorable", Contemporary Mathematics 98, With the collaboration of J. Koch., Providence, RI: American Mathematical Society, 741 pp.

Хроматическое число плоскости

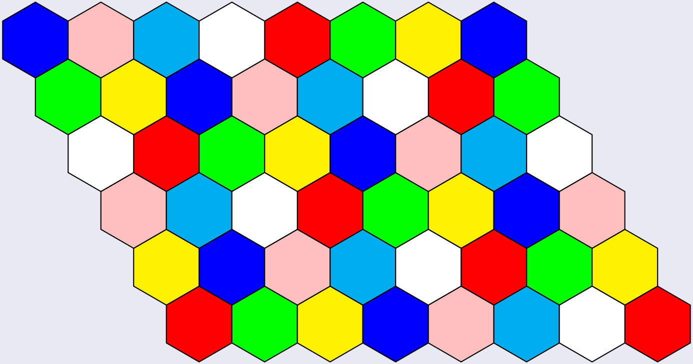
Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

Семи красок точно достаточно:

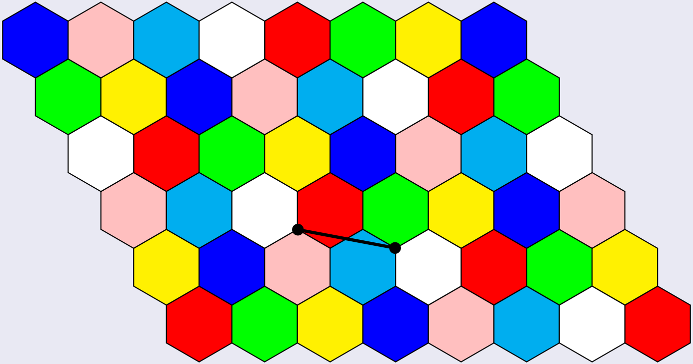


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

Семи красок точно достаточно:

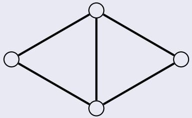


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

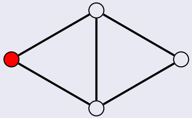


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

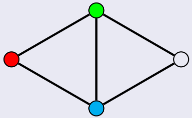
А трех красок мало:



Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

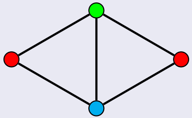


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

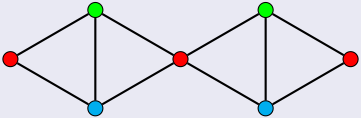


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

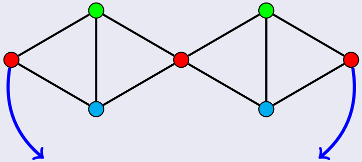


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

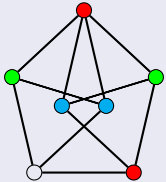


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

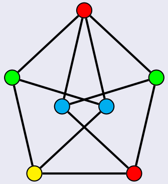
А трех красок мало:



Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:

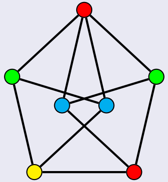


Хроматическое число плоскости

Вопрос

С использованием какого минимального количества красок можно так раскрасить плоскость, чтобы любые две точки, находящиеся на расстоянии 1, были разного цвета?

А трех красок мало:



Оказывается, четырех красок тоже мало:

Построен пример графа с 1585 вершинами, подтверждающий это утверждение.

Aubrey de Grey, "The chromatic number of the plane is at least 5" (11.04.2018)
https://arxiv.org/pdf/1804.02385.pdf

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) =$

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) =$

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

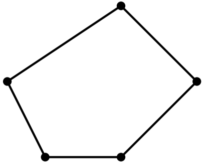
- $f(3) = 3$;
- $f(4) = 5$;

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

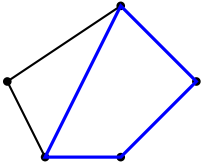
- $f(3) = 3$;
- $f(4) = 5$;



Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) = 5$;

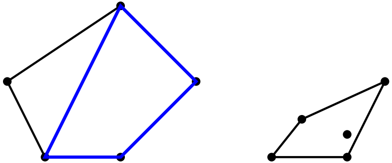


Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

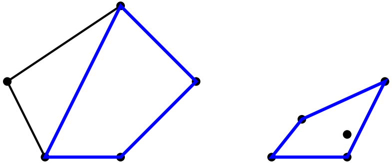
- $f(3) = 3$;
- $f(4) = 5$;



Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

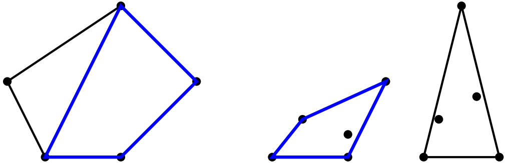
- $f(3) = 3$;
- $f(4) = 5$;



Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

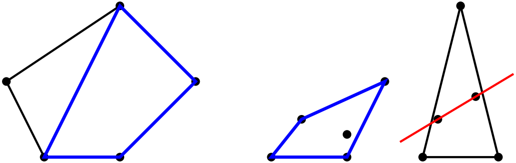
- $f(3) = 3$;
- $f(4) = 5$;



Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

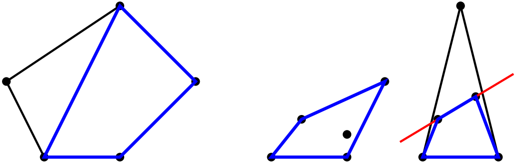
- $f(3) = 3$;
- $f(4) = 5$;



Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) = 5$;



Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

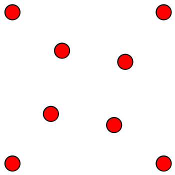
- $f(3) = 3$;
- $f(4) = 5$;
- $f(5) =$

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) = 5$;
- $f(5) = 9$;



Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) = 5$;
- $f(5) = 9$;
- $f(N) \geqslant 1 + 2^{N-2}$;

Задача со счастливым концом

Вопрос (Erdős, 1933)

Каково минимальное значение $f(N)$ такое, что среди любых $f(N)$ точек на плоскости (из которых никакие три не лежат на одной прямой) гарантированно найдется выпуклый N-угольник?

- $f(3) = 3$;
- $f(4) = 5$;
- $f(5) = 9$;
- $f(N) \geqslant 1 + 2^{N-2}$;
- $f(6) = 17$.
  G. Szekeres, L. Peters (2006), "Computer solution to the 17-point Erdős-Szekeres problem", ANZIAM Journal T. 48 (02): 151–164.

Диаметр графа состояний кубика Рубика

Диаметр графа состояний кубика Рубика

Теорема (2010)

Диаметр графа состояний кубика Рубика в метрике FTM равен 20.

T. Rokicki, H. Kociemba, M. Davidson, and J. Dethridge, около 35 лет машинного времени

Диаметр графа состояний кубика Рубика

Теорема (2010)

Диаметр графа состояний кубика Рубика в метрике FTM равен 20.

T. Rokicki, H. Kociemba, M. Davidson, and J. Dethridge, **около 35 лет машинного времени**

Теорема (2014)

Диаметр графа состояний кубика Рубика в метрике QTM равен 26.

T. Rokicki, M. Davidson, **около 29 лет машинного времени**

Диаметр графа состояний кубика Рубика

Теорема (2010)

Диаметр графа состояний кубика Рубика в метрике FTM равен 20.

T. Rokicki, H. Kociemba, M. Davidson, and J. Dethridge, около 35 лет машинного времени

Теорема (2014)

Диаметр графа состояний кубика Рубика в метрике QTM равен 26.

T. Rokicki, M. Davidson, около 29 лет машинного времени http://www.cube20.org/

Теорема

Не существует однозначно разрешимой головоломки “Судоку” с 16 подсказками.

Gary McGuire, Bastian Tugemann, Gilles Civario (2012), “There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration”, http://arxiv.org/abs/1201.0749, **почти год реального времени, кластер из 320 6-ядерных Xeon E5650.**

			8		1			
							4	3
5								
				7		8		
						1		
	2			3				
6							7	5
		3	4					
			2			6		

Теорема

Не существует однозначно разрешимой головоломки “Судоку” с 16 подсказками.

Gary McGuire, Bastian Tugemann, Gilles Civario (2012), “There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration”, http://arxiv.org/abs/1201.0749, **почти год реального времени, кластер из 320 6-ядерных Xeon E5650**.

			8		1			
							4	3
5								
				7		8		
						1		
	2			3				
6							7	5
		3	4					
			2			6		

Вопрос

Можно ли доверять компьютерным доказательствам?

Многостадийные задачи и стандартная нижняя оценка

Постановка

- Даны множества машин и работ;
- Каждая работа состоит из операций;
- Каждая операция выполняется на определённой машине за заданное время (длительность);
- Операции одной работы/на одной машине не могут выполняться одновременно.

Многостадийные задачи и стандартная нижняя оценка

Постановка

- Даны множества машин и работ;
- Каждая работа состоит из операций;
- Каждая операция выполняется на определённой машине за заданное время (длительность);
- Операции одной работы/на одной машине не могут выполняться одновременно.

Стандартная нижняя оценка

- ℓ_i — нагрузка i-й машины, сумма длительностей её операций.
- $\ell_{\max} = \max\limits_i \ell_i$ — максимальная нагрузка.
- d_j — длина j—й работы, сумма длительностей её операций.
- $d_{\max} = \max\limits_j d_j$ — максимальная длина работы

$$C^*_{\max} \geqslant \bar{C} = \{\ell_{\max}, d_{\max}\}.$$

Локализация оптимумов: напоминание

Задача

Для данного класса примеров $\mathcal{I}$ найти наименьшее значение ρ такое, что оптимумы всех примеров из I лежат в интервале $[\bar{C}, \rho\bar{C}]$.

Иными словами:

1. Доказать, что $\forall I \in \mathcal{I} \;\; C^*_{\max}(I) \leqslant \rho\bar{C}$, и
2. показать, что существует пример $\tilde{I} \in \mathcal{I}$ такой, что $C^*_{\max}(\tilde{I}) = \rho\bar{C}$.

Иными словами: найти

$$\rho = \sup_{I \in \mathcal{I}} \frac{C^*_{\max}(I)}{\bar{C}(I)}.$$

Локализация оптимумов: напоминание

Задача

Для данного класса примеров $\mathcal{I}$ найти наименьшее значение ρ такое, что оптимумы всех примеров из I лежат в интервале $[\bar{C}, \rho\bar{C}]$.

Иными словами:

1. Доказать, что $\forall I \in \mathcal{I} \;\; C^*_{\max}(I) \leqslant \rho\bar{C}$, и
2. показать, что существует пример $\tilde{I} \in \mathcal{I}$ такой, что $C^*_{\max}(\tilde{I}) = \rho\bar{C}$.

Иными словами: найти

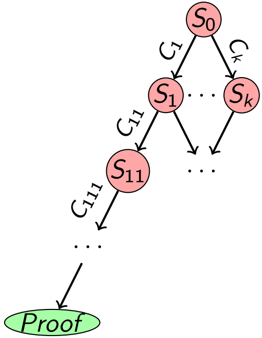
$$\rho = \sup_{I \in \mathcal{I}} \frac{C^*_{\max}(I)}{\bar{C}(I)}.$$

Цель компьютеризированного доказательства — подтвердить, что

$$\forall I \in \mathcal{I} \;\; C^*_{\max}(I) \leqslant \rho\bar{C}, \qquad (*)$$

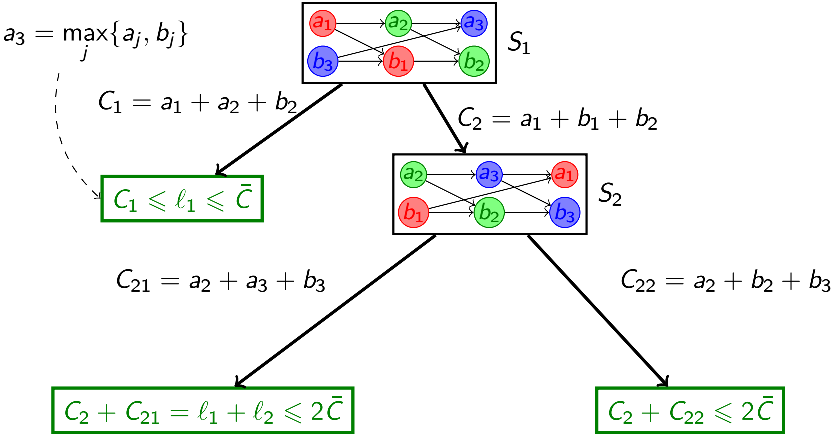
если ρ известно (или предполагается), или найти ρ и критический пример $\tilde{I}$ (попутно доказав (*)).

Структура доказательства



- Вершины соответствуют подмножествам примеров;
- в красных вершинах — схемы расписаний
- выходящие из них дуги — варианты нетривиальных критических путей в схемах. Каждый путь — выражение, сумма длительностей некоторых операций;
- в висячих (зеленых) вершинах — доказательства, что для критического примера из рассматриваемого подмножества оптимум не превосходит $\rho\bar{C}$.

Простой пример $(O2|n=3|C_{\max})$



Алгоритм построения

Гипотеза: Для любого примера (с заданными ограничениями) существует расписание S с длиной $C_{\max}(S) \leqslant \rho^* \bar{C}$.

- Имеется набор схем $\mathcal{S} = \{S_0, \ldots, S_N\}$.
- Для каждой схемы S_k описан набор вариантов нетривиальных критических путей $\mathcal{P}(S_k)$.

1. Корень дерева: $S := S_0$. Текущий набор путей: $Q = \emptyset$.
2. Для каждого $P \in \mathcal{P}(S)$:
    1. $Q := Q \cup \{P\}$.
    2. Находим критический пример $\tilde{I} = \arg\max\limits_{I} \min\limits_{P \in Q} P(I)$.
    3. Если $\min\limits_{P \in Q} P(\tilde{I}) \leqslant \rho^*$, $Q := Q \setminus \{P\}$ и переходим к следующему P.
    4. Иначе
        1. Находим в $\mathcal{S}$ улучшающую схему S для примера $\tilde{I}$, переходим на Шаг 2.
        2. Если улучшающей схемы не существует, то STOP, вывод критического примера.
3. Доказано!

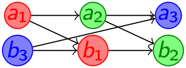
Гипотеза: Для любого примера (с заданными ограничениями) существует расписание S с длиной $C_{\max}(S) \leqslant \rho^* \bar{C}$.

- Имеется набор схем $\mathcal{S} = \{S_0, \ldots, S_N\}$.
- Для каждой схемы S_k описан набор вариантов нетривиальных критических путей $\mathcal{P}(S_k)$.

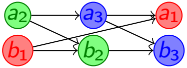
1. Корень дерева: $S := S_0$. Текущий набор путей: $Q = \emptyset$.
2. **это Шаг 2!** Для каждого $P \in \mathcal{P}(S)$:
    1. $Q := Q \cup \{P\}$.
    2. Находим критический пример $\tilde{I} = \arg\max_{I} \min_{P \in Q} P(I)$.
    3. Если $\min_{P \in Q} P(\tilde{I}) \leqslant \rho^*$, $Q := Q \setminus \{P\}$ и переходим к следующему P.
    4. Иначе
        1. Находим в $\mathcal{S}$ улучшающую схему S для примера $\tilde{I}$, переходим на Шаг 2.
        2. Если улучшающей схемы не существует, то STOP, вывод критического примера.
3. Доказано!

Технические особенности. Схемы расписаний.

- Для алгоритма достаточно про каждую схему знать набор вариантов путей для него (т.е. схема описывается набором путей).
- Схем может потребоваться много, хочется их описывать компактно.
- Идея с перестановками:
  - Две схемы, которые мы использовали в примере, эквивалентны:



$\Rightarrow \pi^J = (2, 3, 1) \Rightarrow$



  - Варианты критических путей в них тоже эквивалентны:

$$\begin{cases} a_1 + a_2 + b_2, \\ a_1 + b_1 + b_2 \end{cases} \Rightarrow \pi^J = (2, 3, 1) \Rightarrow \begin{cases} a_2 + a_3 + b_3, \\ a_2 + b_2 + b_3 \end{cases}$$

  - Если описать допустимые перестановки работ и машин, то из одной схемы (шаблона) можно получить сразу много.
  - Достаточно описывать попарно неэквивалентные схемы, и общие наборы допустимых перестановок.

Технические особенности. ЗЛП.

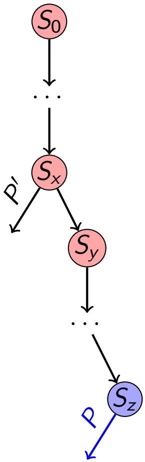
1. Для доказательства (в висячих вершинах) нужно точное решение ЗЛП (на максимум): нам необходима гарантия, что оптимум ЗЛП не превосходит ρ^*.
2. Возможные подходы:
   - Решать ЗЛП точно (в рациональных числах!). Это замедляет процесс.
   - Решать ЗЛП в действительных числах (float), но для висячих вершин проверять в рациональных.
   - Решать ЗЛП в действительных числах. Для висячих вершин запоминать решение двойственной задачи. Округлить это решение до рациональных значений. Проверить, что полученное решение является допустимым для двойственной ЗЛП с рациональными коэффициентами, и что значение двойственной целевой функции не превосходит ρ^*.

Оптимизация процесса

Сокращение дерева доказательства

- Упорядочить длительности операций / длины работ / нагрузки машин без ограничения общности.
- Разбить на подзадачи (по числу работ, по суммарной нагрузке примера, по каким-то свойствам).
- При выборе улучшающей схемы:
  - Учитывать количество вариантов критических путей
  - Учитывать среднюю длину некритических путей на критическом примере
- “Теорема Зяблицкого”.

- Упорядочить длительности операций / длины работ / нагрузки машин без ограничения общности.
- Разбить на подзадачи (по числу работ, по суммарной нагрузке примера, по каким-то свойствам).
- При выборе улучшающей схемы:
    - Учитывать количество вариантов критических путей
    - Учитывать среднюю длину некритических путей на критическом примере
- "Теорема Зяблицкого".



Входные файлы: описание модели


```
{    "M": 2,
     "J": 3,
     "params": 0,
     "missed_vertexes": "",
     "bounds": [
          [ "Lmax", "<", "1", "Lmax" ],
          [ "Dmax", "<", "1", "Dmax" ]
     ],
     "expressions": [],
     "hypothesis": "1",
     "objective_augmentation": "",
     "improvement_flag": true,
     "Machines": [
          [ 1, 2 ]
     ],
     "Jobs": [
          [ 1, 2, 3 ]
     ]}
```

Входные файлы: пример описания шаблона

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "a1 b1",
      "b2 a2",
      "b3 a3"
    ]
  ]
}
```

Входные файлы: пример описания шаблона

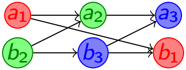
```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "a1 b1",
      "b2 a2",
      "b3 a3"
    ]
  ]
}
```

$$a_1 \to a_2 \to a_3,$$
$$b_2 \to b_3 \to b_1,$$
$$a_1 \to b_1,$$
$$b_2 \to a_2,$$
$$b_3 \to a_3.$$

Входные файлы: пример описания шаблона

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "a1 b1",
      "b2 a2",
      "b3 a3"
    ]
  ]
}
```

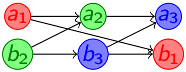
$$a_1 \to a_2 \to a_3,$$
$$b_2 \to b_3 \to b_1,$$
$$a_1 \to b_1,$$
$$b_2 \to a_2,$$
$$b_3 \to a_3.$$



Входные файлы: пример описания шаблона

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "a1 b1",
      "b2 a2",
      "b3 a3"
    ]
  ]
}
```

$$
\begin{aligned}
&a_1 \to a_2 \to a_3,\\
&b_2 \to b_3 \to b_1,\\
&a_1 \to b_1,\\
&b_2 \to a_2,\\
&b_3 \to a_3.
\end{aligned}
$$



$$
\begin{cases}
P_0 : b_2 + a_2 + a_3,\\
P_1 : b_2 + b_3 + b_1
\end{cases}
$$

Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

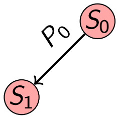
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$

S_0

Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

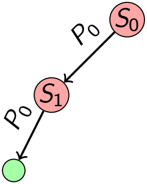
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

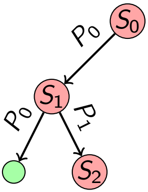
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

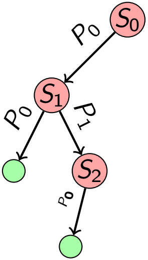
$$
\begin{aligned}
S_0 &= S_{(0,0,0)} \\
S_1 &= S_{(0,1,2)} \\
S_2 &= S_{(0,0,4)}
\end{aligned}
$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

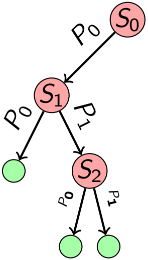
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

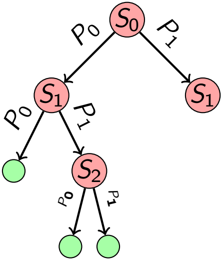
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

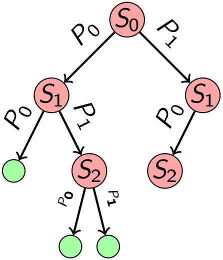
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

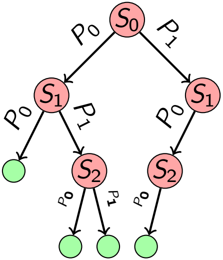
$$\begin{aligned} S_0 &= S_{(0,0,0)} \\ S_1 &= S_{(0,1,2)} \\ S_2 &= S_{(0,0,4)} \end{aligned}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

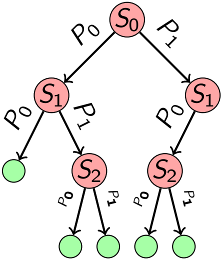
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

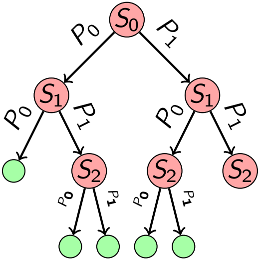
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

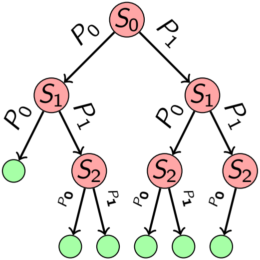
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

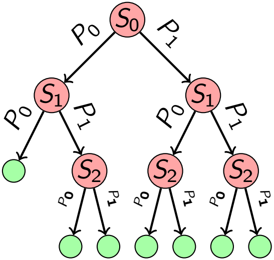
$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$



Выходной файл: дерево доказательства

```
0 (0 1 2 )
0 0 [L1=1/2 L2=1/2 P_1=1/2 P_2=1/2 ]
0 1 (0 0 4 )
0 1 0 [L1=2/3 L2=1/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
0 1 1 [L1=1/2 L2=1/2 D3=0 P_2=1/2 P_3=1/2 ]
1 (0 1 2 )
1 0 (0 0 4 )
1 0 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 0 1 [L1=1/3 L2=2/3 P_1=1/3 P_2=1/3 P_3=1/3 ]
1 1 (0 0 4 )
1 1 0 [L1=1/2 L2=1/2 D3=0 P_1=1/2 P_3=1/2 ]
1 1 1 [L1=1/2 L2=1/2 P_2=1/2 P_3=1/2 ]
```

$$S_0 = S_{(0,0,0)}$$
$$S_1 = S_{(0,1,2)}$$
$$S_2 = S_{(0,0,4)}$$

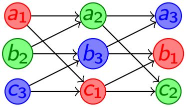


Еще пример

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "c3 c1 c2",
      "a1 c1 b1",
      "b2 a2 c2",
      "c3 b3 a3"
    ]
  ]
}
```

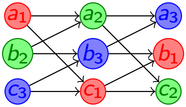
Еще пример

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "c3 c1 c2",
      "a1 c1 b1",
      "b2 a2 c2",
      "c3 b3 a3"
    ]
  ]
}
```



Еще пример

```
{
  "base": [
    [
      "a1 a2 a3",
      "b2 b3 b1",
      "c3 c1 c2",
      "a1 c1 b1",
      "b2 a2 c2",
      "c3 b3 a3"
    ]
  ]
}
```



Пример: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}$