Алгоритмическая теория игр
Лекция 3

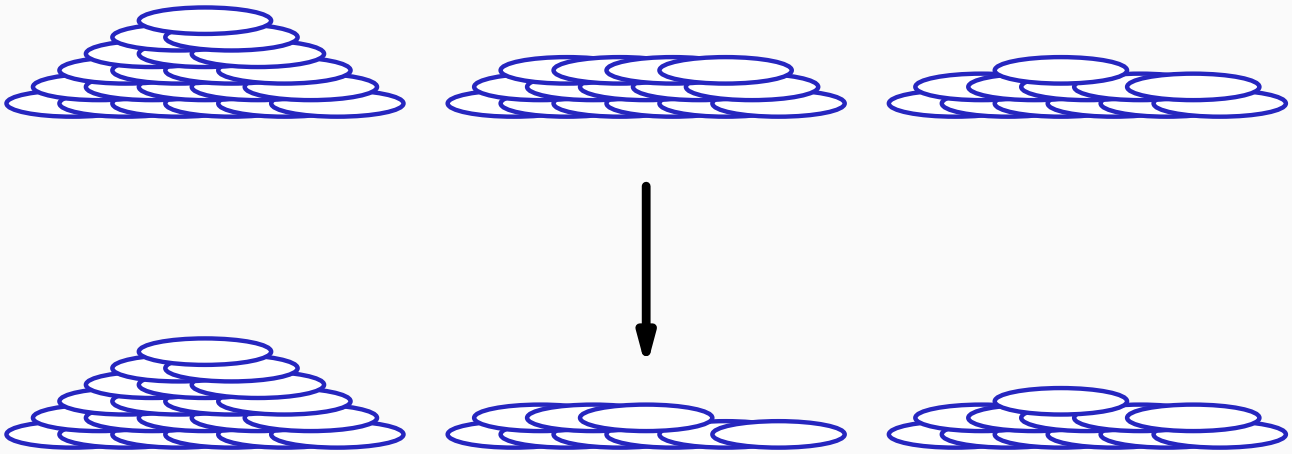
М.Н. Вялый

Лекции в Computer Science club (Санкт-Петербург), 2019

Ним и ним-функция

Ним

В этой игре есть три кучки камней. Игроки ходят по очереди. Игрок на любом ходе может взять любое количество камней из одной кучки, причём хотя бы один камень должен быть взят. Кто не может сделать ход, проиграл.



Ним: усложнения и упрощения

Несколько кучек

Позиция: набор положительных целых чисел. Единственная терминальная позиция 0.

Простейший вариант: ним с одной кучкой

Конечно, позиция 0 — это P-позиция; позиции $n > 0$ — это N-позиции.

Ним: усложнения и упрощения

Несколько кучек

Позиция: набор положительных целых чисел. Единственная терминальная позиция 0.

Простейший вариант: ним с одной кучкой

Конечно, позиция 0 — это P-позиция; позиции $n > 0$ — это N-позиции.

Ним-функция (функция Шпрага–Гранди)

Определение

Пусть в беспристрастной игре на графе G в терминальной позиции проигрывает тот, чья очередь хода. Функция $S\colon V \to \mathbb{N}$ определяется как

- если v — терминальная, то $S(v) = 0$;
- если v — нетерминальная, то

$$S(v) = \operatorname{mex}(\{S(u) : (vu) \in E\}),$$

$$\text{где} \quad \operatorname{mex}(X) = \min_{x \in \mathbb{N} \setminus X} x.$$

Примеры: $\operatorname{mex}() = 0$; $\operatorname{mex}(1, 2, 5) = 0$; $\operatorname{mex}(0, 1, 2, 3) = 4$.

Ним-функция с для нима с одной кучкой: $S(n) = n$.

Ним-функция (функция Шпрага–Гранди)

Определение

Пусть в беспристрастной игре на графе G в терминальной позиции проигрывает тот, чья очередь хода. Функция $S\colon V \to \mathbb{N}$ определяется как

- если v — терминальная, то $S(v) = 0$;
- если v — нетерминальная, то

$$S(v) = \operatorname{mex}(\{S(u) : (vu) \in E\}),$$

$$\text{где} \quad \operatorname{mex}(X) = \min_{x \in \mathbb{N} \setminus X} x.$$

Примеры: $\operatorname{mex}() = 0$; $\operatorname{mex}(1, 2, 5) = 0$; $\operatorname{mex}(0, 1, 2, 3) = 4$.

Ним-функция с для нима с одной кучкой: $S(n) = n$.

Ним-функция уточняет оценку позиции

Лемма

Пусть $S\colon V \mapsto \mathbb{N}$ — ним-функция. Тогда P-позиции — это в точности те позиции, для которых $S(v) = 0$. Остальные позиции являются N-позициями.

Доказательство

Обратная индукция. Основано на очевидном свойстве ним-функции:

(1) если среди аргументов есть 0, то $\mathrm{mex}(\cdot) > 0$;

(2) если все аргументы положительные, то $\mathrm{mex}(\cdot) = 0$.

Ним-функция уточняет оценку позиции

Лемма

Пусть $S\colon V \mapsto \mathbb{N}$ — ним-функция. Тогда P-позиции — это в точности те позиции, для которых $S(v) = 0$. Остальные позиции являются N-позициями.

Доказательство

Обратная индукция. Основано на очевидном свойстве ним-функции:

(1) если среди аргументов есть 0, то $\mathrm{mex}(\cdot) > 0$;

(2) если все аргументы положительные, то $\mathrm{mex}(\cdot) = 0$.

Сумма игр

Определение

Суммой игр $G_1 \oplus G_2$ называется игра, в которой игроки параллельно играют в игры G_1 и G_2, на каждом ходу игрок должен сделать ход в одной из игр.

Пример

Ним с k кучками камней является суммой k экземпляров нима с одной кучкой камней.

Теорема

$S_1 \colon V_1 \to \mathbb{N}$; $S_2 \colon V_2 \to \mathbb{N}$ — ним-функции игр G_1, G_2, в которых все терминальные позиции является P-позициями. Тогда

$$S(v_1, v_2) = S(v_1) \oplus S(v_2),$$

где $x \oplus y$ обозначает поразрядную сумму по модулю 2.

Сумма игр

Определение

Суммой игр $G_1 \oplus G_2$ называется игра, в которой игроки параллельно играют в игры G_1 и G_2, на каждом ходу игрок должен сделать ход в одной из игр.

Пример

Ним с k кучками камней является суммой k экземпляров нима с одной кучкой камней.

Теорема

$S_1 \colon V_1 \to \mathbb{N}$; $S_2 \colon V_2 \to \mathbb{N}$ — ним-функции игр G_1, G_2, в которых все терминальные позиции является P-позициями. Тогда

$$S(v_1, v_2) = S(v_1) \oplus S(v_2),$$

где $x \oplus y$ обозначает поразрядную сумму по модулю 2.

Сумма игр

Определение

Суммой игр $G_1 \oplus G_2$ называется игра, в которой игроки параллельно играют в игры G_1 и G_2, на каждом ходу игрок должен сделать ход в одной из игр.

Пример

Ним с k кучками камней является суммой k экземпляров нима с одной кучкой камней.

Теорема

$S_1 \colon V_1 \to \mathbb{N}$; $S_2 \colon V_2 \to \mathbb{N}$ — ним-функции игр G_1, G_2, в которых все терминальные позиции является P-позициями. Тогда

$$S(v_1, v_2) = S(v_1) \oplus S(v_2),$$

где $x \oplus y$ обозначает поразрядную сумму по модулю 2.

Свойства поразрядной суммы

Лемма о свойствах $x \oplus y$

симметричность: $x \oplus y = y \oplus x$;

инъективность: если $a \neq a'$, то $a \oplus x \neq a' \oplus x$ для любого x;

плотность: если $x < a \oplus b$, то либо $x = a' \oplus b$, $a' < a$; либо $x = a \oplus b'$, $b' < b$.

Доказательство

Свойства поразрядной суммы

Лемма о свойствах $x \oplus y$

симметричность: $x \oplus y = y \oplus x$;

инъективность: если $a \neq a'$, то $a \oplus x \neq a' \oplus x$ для любого x;

плотность: если $x < a \oplus b$, то либо $x = a' \oplus b$, $a' < a$; либо $x = a \oplus b'$, $b' < b$.

Доказательство

Симметричность очевидна из определения.

Свойства поразрядной суммы

Лемма о свойствах $x \oplus y$

симметричность: $x \oplus y = y \oplus x$;

инъективность: если $a \neq a'$, то $a \oplus x \neq a' \oplus x$ для любого x;

плотность: если $x < a \oplus b$, то либо $x = a' \oplus b$, $a' < a$; либо $x = a \oplus b'$, $b' < b$.

Доказательство

Инъективность: если поразрядно прибавить к двум разным двоичным строкам одну и ту же строку, полученные суммы будут различаться в тех же самых позициях, что и исходные строки.

Свойства поразрядной суммы

Лемма о свойствах $x \oplus y$

симметричность: $x \oplus y = y \oplus x$;

инъективность: если $a \neq a'$, то $a \oplus x \neq a' \oplus x$ для любого x;

плотность: если $x < a \oplus b$, то либо $x = a' \oplus b$, $a' < a$; либо $x = a \oplus b'$, $b' < b$.

Доказательство

Плотность. В самом старшем разряде, в котором x отличается от $a \oplus b$, в двоичной записи числа x стоит 0, а в двоичной записи $a \oplus b$ стоит 1.

Поэтому в двоичных записях чисел a, b в этом разряде ровно одна единица. Считаем без ограничения общности, что в этом разряде запись a содержит 1, а запись b содержит 0.

Свойства поразрядной суммы

Лемма о свойствах $x \oplus y$

симметричность: $x \oplus y = y \oplus x$;

инъективность: если $a \neq a'$, то $a \oplus x \neq a' \oplus x$ для любого x;

плотность: если $x < a \oplus b$, то либо $x = a' \oplus b$, $a' < a$; либо $x = a \oplus b'$, $b' < b$.

Доказательство

Плотность. (Слева — самый младший разряд.)

a:	a_1	1	a_2
b:	b_1	0	b_2
x:	x_1	0	x_2
a':	a_1	0	a'_2

$a_1 \oplus b_1 = x_1$ (по построению).

Полагаем $a'_2 = b_2 \oplus x_2$,

что равносильно $b_2 \oplus a'_2 = x_2$.

Тогда $a' < a$ и $a' \oplus b = x$.

Доказательство теоремы о сумме игр

Обратная индукция.

Для терминальных позиций $0 \oplus 0 = 0$.

Шаг индукции.

Доказательство теоремы о сумме игр

Обратная индукция.

Для терминальных позиций $0 \oplus 0 = 0$.

Шаг индукции.
$S(v_1) \oplus S(v_2)$ отличается от ним-функций всех позиций, куда можно сделать ход.

Доказательство теоремы о сумме игр

Обратная индукция.

Для терминальных позиций $0 \oplus 0 = 0$.

Шаг индукции.
$S(v_1) \oplus S(v_2)$ отличается от ним-функций всех позиций, куда можно сделать ход.

Пусть $(v_1, v_1') \in E(G_1)$; $(v_2, v_2') \in E(G_2)$.
По индуктивному предположению $S(v_1', v_2) = S(v_1') \oplus S(v_2)$;
$S(v_1, v_2') = S(v_1) \oplus S(v_2')$.
По определению ним-функции $S(v_1) \neq S(v_1')$; $S(v_2) \neq S(v_2')$.
Из инъективности поразрядной суммы получаем

$$S(v_1) \oplus S(v_2) \neq S(v_1') \oplus S(v_2); \quad S(v_1) \oplus S(v_2) \neq S(v_1) \oplus S(v_2');$$

Доказательство теоремы о сумме игр

Обратная индукция.

Для терминальных позиций $0 \oplus 0 = 0$.

Шаг индукции.
$S(v_1) \oplus S(v_2)$ — наименьшее из чисел, которые отличаются от ним-функций всех позиций, куда можно сделать ход.

Доказательство теоремы о сумме игр

Обратная индукция.

Для терминальных позиций $0 \oplus 0 = 0$.

Шаг индукции.
$S(v_1) \oplus S(v_2)$ — наименьшее из чисел, которые отличаются от ним-функций всех позиций, куда можно сделать ход.

Пусть $k < S(v_1) \oplus S(v_2)$. Из плотности поразрядной суммы: либо $k = s_1 \oplus S(v_2)$, $s_1 < S(v_1)$; либо $k = S(v_1) \oplus s_2$, $s_2 < S(v_2)$. Тогда $s_1 = S(v_1')$ в первом случае, а во втором $s_2 = S(v_2')$, причём в первом случае есть ход из v_1 в v_1', а во втором случае есть ход из v_2 в v_2'.
Поэтому $k = S(v_1') \oplus S(v_2)$ или $k = S(v_1) \oplus S(v_2')$.

Следствие для решения нима

Теорема

Существует полиномиальный алгоритм, который по двоичным записям количества камней в кучках решает игру ним.

Очень сильный результат: позиций много и длина партии может быть экспоненциально велика. Заранее неочевидна даже принадлежность PSPACE.

Утверждение

Существует полиномиальный алгоритм, который по двоичной записи n решает игру NODE KAYLES на графе-пути P_n.

Следствие для решения нима

Теорема

Существует полиномиальный алгоритм, который по двоичным записям количества камней в кучках решает игру ним.

Очень сильный результат: позиций много и длина партии может быть экспоненциально велика. Заранее неочевидна даже принадлежность PSPACE.

Утверждение

Существует полиномиальный алгоритм, который по двоичной записи n решает игру NODE KAYLES на графе-пути P_n.

Следствие для решения нима

Теорема

Существует полиномиальный алгоритм, который по двоичным записям количества камней в кучках решает игру ним.

Очень сильный результат: позиций много и длина партии может быть экспоненциально велика. Заранее неочевидна даже принадлежность PSPACE.

Утверждение

Существует полиномиальный алгоритм, который по двоичной записи n решает игру NODE KAYLES на графе-пути P_n.

Решение NODE KAYLES на P_n

Наблюдение

Игра на двух отрезках кеглей из n и m штук, разделённых хотя бы двумя пробелами, есть сумма игр на отрезке из n кеглей и отрезке из m кеглей.

Рекуррента

Для $K(n)$ — ним-функции NODE KAYLES на P_n — выполняются соотношения

$$K(n) = \operatorname{mex}\big(K(x) \oplus K(y) : x = n-2, y = 0 \ \vee \ x + y = n-3\big),$$

$$K(0) = 0, \ K(1) = 1, \ K(2) = 1.$$

Решение NODE KAYLES на P_n

Наблюдение

Игра на двух отрезках кеглей из n и m штук, разделённых хотя бы двумя пробелами, есть сумма игр на отрезке из n кеглей и отрезке из m кеглей.

Рекуррента

Для $K(n)$ — ним-функции NODE KAYLES на P_n — выполняются соотношения

$$K(n) = \operatorname{mex}\big(K(x) \oplus K(y) : x = n-2, y = 0 \ \vee \ x + y = n-3\big),$$

$$K(0) = 0, \ K(1) = 1, \ K(2) = 1.$$

Периодичность для NODE KAYLES на P_n

Результаты вычислений ним-функции $K(n)$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$K(n)$	0	1	1	2	0	3	1	1	0	3	3	2	2	4	0	5	2
$K(17+n)$	2	3	3	0	1	1	3	0	2	1	1	0	4	5	2	7	4
$K(34+n)$	0	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(51+n)$	2	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(68+n)$	8	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(85+n)$	9	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(102+n)$	8	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(119+n)$	9	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(136+n)$	8	1															

Задача

В предположении, что таблица составлена правильно, докажите периодичность $K(n)$ с предпериодом 68 и периодом 34.

Периодичность для NODE KAYLES на P_n

Результаты вычислений ним-функции $K(n)$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$K(n)$	0	1	1	2	0	3	1	1	0	3	3	2	2	4	0	5	2
$K(17+n)$	2	3	3	0	1	1	3	0	2	1	1	0	4	5	2	7	4
$K(34+n)$	0	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(51+n)$	2	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(68+n)$	8	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(85+n)$	9	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(102+n)$	8	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5
$K(119+n)$	9	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4
$K(136+n)$	8	1															

Задача

В предположении, что таблица составлена правильно, докажите периодичность $K(n)$ с предпериодом 68 и периодом 34.

Игры вычитания

Одномерная игра вычитания

Правила

Задано множество $\mathcal{D} \subseteq \mathbb{N} \setminus \{0\}$ разрешённых разностей.

Игра происходит с одной кучкой камней. За ход игрок обязан взять такое количество камней d, что $d \in D$.

Кто не может сделать ход, проиграл.

Игра с $\mathcal{D} = \{1\}$

Игра тривиальна, так как ход единственный в каждой позиции.

P-позиции — это чётные числа (камней в кучке); ним-функция: $\nu_{\mathcal{D}}(n) = n \bmod 2$.

Одномерная игра вычитания

Правила

Задано множество $\mathcal{D} \subseteq \mathbb{N} \setminus \{0\}$ разрешённых разностей.

Игра происходит с одной кучкой камней. За ход игрок обязан взять такое количество камней d, что $d \in D$.

Кто не может сделать ход, проиграл.

Игра с $\mathcal{D} = \{1\}$

Игра тривиальна, так как ход единственный в каждой позиции.

P-позиции — это чётные числа (камней в кучке); ним-функция: $\nu_{\mathcal{D}}(n) = n \bmod 2$.

Игра с $\mathcal{D} = \{1, 2\}$

Оценим несколько первых позиций ($\mathrm{val}(\mathcal{D}; n) = 0$ равносильно тому, что n является P-позицией):

n	0	1	2	3	4	5
$\mathrm{val}(\{1, 2\}; n)$	0	1	1	0	1	1

Утверждение

Далее последовательность оценок периодична.

Доказательство

Длина периода 3. Это больше величины разрешённых разностей.

Игра с $\mathcal{D} = \{1, 2\}$

Оценим несколько первых позиций ($\mathrm{val}(\mathcal{D}; n) = 0$ равносильно тому, что n является P-позицией):

n	0	1	2	3	4	5
$\mathrm{val}(\{1, 2\}; n)$	0	1	1	0	1	1

Утверждение

Далее последовательность оценок периодична.

Доказательство

Длина периода 3. Это больше величины разрешённых разностей.

Игра с $\mathcal{D} = \{1, 2\}$

Оценим несколько первых позиций ($\mathrm{val}(\mathcal{D}; n) = 0$ равносильно тому, что n является P-позицией):

n	0	1	2	3	4	5
$\mathrm{val}(\{1, 2\}; n)$	0	1	1	0	1	1

Утверждение

Далее последовательность оценок периодична.

Доказательство

Длина периода 3. Это больше величины разрешённых разностей.

Игра с $\mathcal{D} = \{2, 5\}$

Период 7:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathrm{val}(\{2, 5\}, n)$	0	0	1	1	0	1	1	0	0	1	1	0	1	1

Задача (периодичность выполняется не всегда)

Постройте пример такого множества $\mathcal{D}$, что оценки позиций в игре вычитания с множеством разрешённых разностей $\mathcal{D}$ не образуют периодическую последовательность (даже с предпериодом).

Игра с $\mathcal{D} = \{2, 5\}$

Период 7:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathrm{val}(\{2, 5\}, n)$	0	0	1	1	0	1	1	0	0	1	1	0	1	1

Задача (периодичность выполняется не всегда)

Постройте пример такого множества $\mathcal{D}$, что оценки позиций в игре вычитания с множеством разрешённых разностей $\mathcal{D}$ не образуют периодическую последовательность (даже с предпериодом).

Конечные множества разрешённых разностей

Теорема

Если множество $\mathcal{D}$ конечно, то последовательность $\mathrm{val}(\mathcal{D}, n)$ оценок в игре вычитания будет периодической. Длины периода и предпериода ограничены 2^A, где $A = \max(i : i \in \mathcal{D})$.

Основная идея доказательства

Поскольку на каждом ходе можно взять не более A камней, оценка позиции $\mathrm{val}(\mathcal{D}, n)$ зависит только от оценок позиций $\mathrm{val}(\mathcal{D}, n-1), \ldots, \mathrm{val}(\mathcal{D}, n-A)$.

Возможных наборов оценок не больше 2^A.

Конечные множества разрешённых разностей

Теорема

Если множество $\mathcal{D}$ конечно, то последовательность $\mathrm{val}(\mathcal{D}, n)$ оценок в игре вычитания будет периодической. Длины периода и предпериода ограничены 2^A, где $A = \max(i : i \in \mathcal{D})$.

Основная идея доказательства

Поскольку на каждом ходе можно взять не более A камней, оценка позиции $\mathrm{val}(\mathcal{D}, n)$ зависит только от оценок позиций $\mathrm{val}(\mathcal{D}, n-1), \ldots, \mathrm{val}(\mathcal{D}, n-A)$.

Возможных наборов оценок не больше 2^A.

Формальные подробности доказательства

Последовательность A-мерных булевых векторов v_i:

$$v_{i,j} = \mathrm{val}(\mathcal{D}, Ai + j).$$

Тогда $v_{i+1} = F(v_i)$, где $F\colon \mathbb{B}^A \to \mathbb{B}^A$ — отображение множества A-мерных булевых векторов в себя.

Как только $v_k = v_{k+T}$, эта последовательность векторов становится периодической. Но рано или поздно векторы в последовательности обязаны повториться: всего таких векторов не больше, чем 2^A.

Формальные подробности доказательства

Последовательность A-мерных булевых векторов v_i:

$$v_{i,j} = \mathrm{val}(\mathcal{D}, Ai + j).$$

Тогда $v_{i+1} = F(v_i)$, где $F\colon \mathbb{B}^A \to \mathbb{B}^A$ — отображение множества A-мерных булевых векторов в себя.

Как только $v_k = v_{k+T}$, эта последовательность векторов становится периодической. Но рано или поздно векторы в последовательности обязаны повториться: всего таких векторов не больше, чем 2^A.

Следствие из периодичности

Утверждение

Существует полиномиальный алгоритм решения игры вычитания с фиксированным конечным множеством $\mathcal{D}$ разрешённых разностей.

(Вход алгоритма — двоичная запись количества камней n в кучке; результат работы — оценка позиции.)

Алгоритм

1. Находим период p и предпериод ℓ последовательности оценок, строим таблицу значений функции оценки для первых $\ell + p$ чисел
   (время $O(1)$, так как множество $\mathcal{D}$ не зависит от входа).
2. Если $n > \ell$, то делим $n - \ell$ на p с остатком, находим в таблице периода значение оценки;
   в противном случае берём оценку из таблицы предпериода
   (время $O(\log n)$, так как $p = O(1)$).

Алгоритм

1. Находим период p и предпериод ℓ последовательности оценок, строим таблицу значений функции оценки для первых $\ell + p$ чисел
   (время $O(1)$, так как множество $\mathcal{D}$ не зависит от входа).
2. Если $n > \ell$, то делим $n - \ell$ на p с остатком, находим в таблице периода значение оценки;
   в противном случае берём оценку из таблицы предпериода
   (время $O(\log n)$, так как $p = O(1)$).

Алгоритм

1. Находим период p и предпериод ℓ последовательности оценок, строим таблицу значений функции оценки для первых $\ell + p$ чисел
   (время $O(1)$, так как множество $\mathcal{D}$ не зависит от входа).
2. Если $n > \ell$, то делим $n - \ell$ на p с остатком, находим в таблице периода значение оценки;
   в противном случае берём оценку из таблицы предпериода
   (время $O(\log n)$, так как $p = O(1)$).

Алгоритм

1. Находим период p и предпериод ℓ последовательности оценок, строим таблицу значений функции оценки для первых $\ell + p$ чисел
   (время $O(1)$, так как множество $\mathcal{D}$ не зависит от входа).
2. Если $n > \ell$, то делим $n - \ell$ на p с остатком, находим в таблице периода значение оценки;
   в противном случае берём оценку из таблицы предпериода
   (время $O(\log n)$, так как $p = O(1)$).

С конечными множествами разностей не всё просто

Открытый (?) вопрос

Существует ли общий полиномиальный алгоритм решения игр вычитания (входом является и множество разрешённых разностей, и число камней).

Вопрос не очевиден, даже когда разности заданы в унарной системе (то есть максимальная разность A линейно ограничена длиной входа).

С конечными множествами разностей не всё просто

Открытый (?) вопрос

Существует ли общий полиномиальный алгоритм решения игр вычитания (входом является и множество разрешённых разностей, и число камней).

Вопрос не очевиден, даже когда разности заданы в унарной системе (то есть максимальная разность A линейно ограничена длиной входа).

Многомерные игры вычитания

Правила игры

Есть несколько кучек камней. То есть, позиция — это вектор

$$x = (x_1, \ldots, x_d) \in \mathbb{N}^d.$$

Ход возможен из позиции x в позицию $x - a$, $a \in \mathcal{D}$ (множество разрешённых разностей).

Предполагаем, что $|\mathcal{D}| < \infty$ и для любого $a \in \mathcal{D}$

$$\sum_{i=1}^{d} a_i > 0$$

(это гарантирует конечность игры из любой позиции).

Неформально, на каждом ходе в какие-то кучки могут добавляться камни, но общее количество камней обязано убывать.

Правила игры

Есть несколько кучек камней. То есть, позиция — это вектор

$$x = (x_1, \ldots, x_d) \in \mathbb{N}^d.$$

Ход возможен из позиции x в позицию $x - a$, $a \in \mathcal{D}$ (множество разрешённых разностей).

Предполагаем, что $|\mathcal{D}| < \infty$ и для любого $a \in \mathcal{D}$

$$\sum_{i=1}^{d} a_i > 0$$

(это гарантирует конечность игры из любой позиции).

Неформально, на каждом ходе в какие-то кучки могут добавляться камни, но общее количество камней обязано убывать.

Простейший алгоритм

Чтобы оценить позицию x, находим всю таблицу оценок позиций с $\leqslant N$ камнями в кучке, где

$$N = \sum_{i=1}^{d} x_i,$$

пользуясь разбором с конца.

Всего таких позиций

$$\binom{N+d-1}{d-1}.$$

Если количества камней заданы в двоичной системе, то эта величина $\Omega(2^n)$, где n — длина входа.

Вопрос: можно ли быстрее?

Ответ: для *некоторых* игр существенно быстрее нельзя.

Простейший алгоритм

Чтобы оценить позицию x, находим всю таблицу оценок позиций с $\leqslant N$ камнями в кучке, где

$$N = \sum_{i=1}^{d} x_i,$$

пользуясь разбором с конца.

Всего таких позиций

$$\binom{N+d-1}{d-1}.$$

Если количества камней заданы в двоичной системе, то эта величина $\Omega(2^n)$, где n — длина входа.

Вопрос: можно ли быстрее?

Ответ: для *некоторых* игр существенно быстрее нельзя.

Простейший алгоритм

Чтобы оценить позицию x, находим всю таблицу оценок позиций с $\leqslant N$ камнями в кучке, где

$$N = \sum_{i=1}^{d} x_i,$$

пользуясь разбором с конца.

Всего таких позиций

$$\binom{N+d-1}{d-1}.$$

Если количества камней заданы в двоичной системе, то эта величина $\Omega(2^n)$, где n — длина входа.

Вопрос: можно ли быстрее?

Ответ: для *некоторых* игр существенно быстрее нельзя.

Простейший алгоритм

Чтобы оценить позицию x, находим всю таблицу оценок позиций с $\leqslant N$ камнями в кучке, где

$$N = \sum_{i=1}^{d} x_i,$$

пользуясь разбором с конца.

Всего таких позиций

$$\binom{N+d-1}{d-1}.$$

Если количества камней заданы в двоичной системе, то эта величина $\Omega(2^n)$, где n — длина входа.

Вопрос: можно ли быстрее?

Ответ: для некоторых игр существенно быстрее нельзя.

Простейший алгоритм

Чтобы оценить позицию x, находим всю таблицу оценок позиций с $\leqslant N$ камнями в кучке, где

$$N = \sum_{i=1}^{d} x_i,$$

пользуясь разбором с конца.

Всего таких позиций

$$\binom{N+d-1}{d-1}.$$

Если количества камней заданы в двоичной системе, то эта величина $\Omega(2^n)$, где n — длина входа.

Вопрос: можно ли быстрее?

Ответ: для некоторых игр существенно быстрее нельзя.

Трудность игр вычитания (неограниченная размерность)

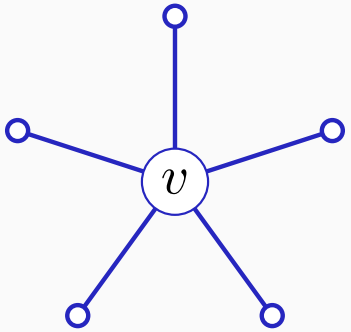
Решение игры NODE KAYLES (PSPACE-полна) сводится к решению игры вычитания.

G — граф игры NODE KAYLES, количество рёбер равно d.

Разрешённые разности: $\mathcal{D} = \{a^{(v)} : v \in V(G)\}$, где

$$a_e^{(v)} = \begin{cases} 1, & \text{если вершина } v \text{ является концом ребра } e, \\ 0, & \text{в противном случае.} \end{cases}$$

Звезда $a^{(v)}$:



Начальная позиция: $\mathbb{1}$ (все координаты равны 1).

Позиции в построенной игре вычитания

Поскольку отрицательные количества камней в кучках невозможны, это попросту подмножества рёбер графа.

После вычитания $a^{(v)}$ исчезают инцидентные v рёбра. Поэтому в дальнейшем вычитание $a^{(u)}$, где $(u, v) \in E(G)$, невозможно.

Если текущая позиция

$$\mathbb{1} - \sum_{v \in X} a^{(v)}$$

и вершина u не связана ребром ни с одной вершиной из множества X, то вычитание вектора $a^{(u)}$ возможно.

Вывод

Граф позиций и ходов для этой игры вычитания совпадает с графом позиций и ходов NODE KAYLES на графе G.

Позиции в построенной игре вычитания

Поскольку отрицательные количества камней в кучках невозможны, это попросту подмножества рёбер графа.

После вычитания $a^{(v)}$ исчезают инцидентные v рёбра. Поэтому в дальнейшем вычитание $a^{(u)}$, где $(u, v) \in E(G)$, невозможно.

Если текущая позиция

$$\mathbb{1} - \sum_{v \in X} a^{(v)}$$

и вершина u не связана ребром ни с одной вершиной из множества X, то вычитание вектора $a^{(u)}$ возможно.

Вывод

Граф позиций и ходов для этой игры вычитания совпадает с графом позиций и ходов NODE KAYLES на графе G.

Позиции в построенной игре вычитания

Поскольку отрицательные количества камней в кучках невозможны, это попросту подмножества рёбер графа.

После вычитания $a^{(v)}$ исчезают инцидентные v рёбра. Поэтому в дальнейшем вычитание $a^{(u)}$, где $(u, v) \in E(G)$, невозможно.

Если текущая позиция

$$\mathbb{1} - \sum_{v \in X} a^{(v)}$$

и вершина u не связана ребром ни с одной вершиной из множества X, то вычитание вектора $a^{(u)}$ возможно.

Вывод

Граф позиций и ходов для этой игры вычитания совпадает с графом позиций и ходов NODE KAYLES на графе G.

Позиции в построенной игре вычитания

Поскольку отрицательные количества камней в кучках невозможны, это попросту подмножества рёбер графа.

После вычитания $a^{(v)}$ исчезают инцидентные v рёбра. Поэтому в дальнейшем вычитание $a^{(u)}$, где $(u, v) \in E(G)$, невозможно.

Если текущая позиция

$$\mathbb{1} - \sum_{v \in X} a^{(v)}$$

и вершина u не связана ребром ни с одной вершиной из множества X, то вычитание вектора $a^{(u)}$ возможно.

Вывод

Граф позиций и ходов для этой игры вычитания совпадает с графом позиций и ходов NODE KAYLES на графе G.

Игры вычитания в размерности $O(1)$

Решение таких игр также трудно. Доказательство основано на связи между играми вычитания и клеточными автоматами, открытой U. Larsson'ом и J. Wästlund'ом.

U. Larsson and W. J. Wästlund. From heaps of matches to the limits of computability. *The Electronic Journal of Combinatorics*, 20(3):#P41, 2013.

Клеточные автоматы

Одномерные клеточные автоматы (КА)

КА похож на МТ. Но теперь функция переходов применяется во всех ячейках одновременно.

Определение

Клеточный автомат (КА) C задаётся алфавитом A, размером окрестности r и функцией переходов $\delta\colon A^{2r+1} \to A$.

Конфигурация КА: $c\colon \mathbb{Z} \to A$ (бесконечная в обе стороны лента, в ячейках записаны символы алфавита).

Изменение конфигурации за такт работы:

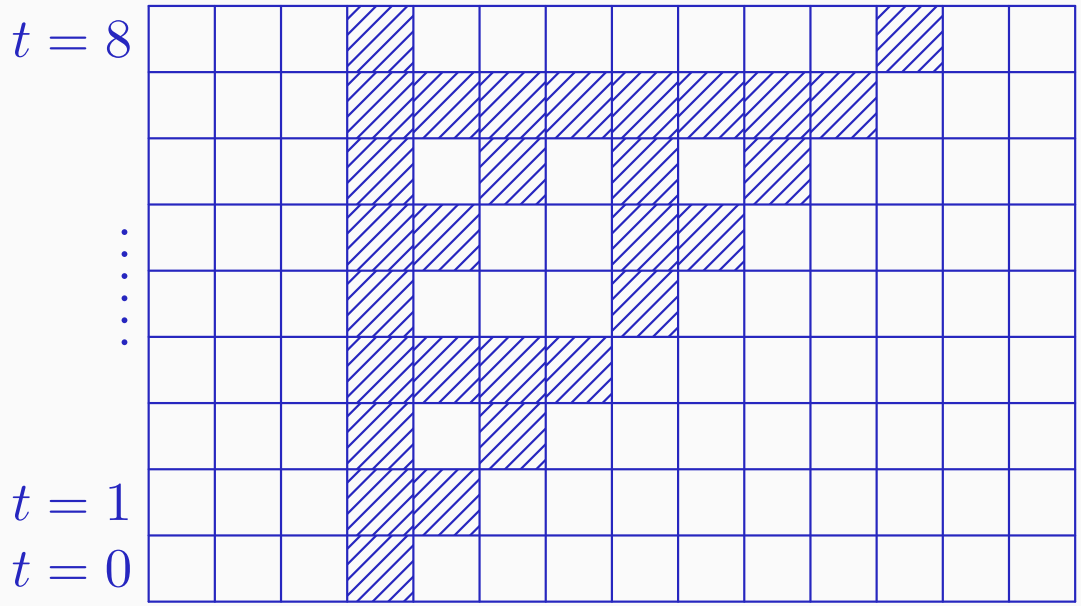
$$c'(x) = \delta\big(c(x-r), c(x-r+1), \ldots, c(x), \ldots, c(x+r-1), c(x+r)\big).$$

Пустой символ Λ: $\delta(\Lambda, \ldots, \Lambda) = \Lambda$.

Пример двоичного КА (автомат Паскаля)

$A = \{0, 1\}$; $r = 1$; $\delta(x, y, z) = 1 \oplus x \oplus y$.

$\Lambda = 1$ (здесь и далее будет такое соглашение для двоичных КА)

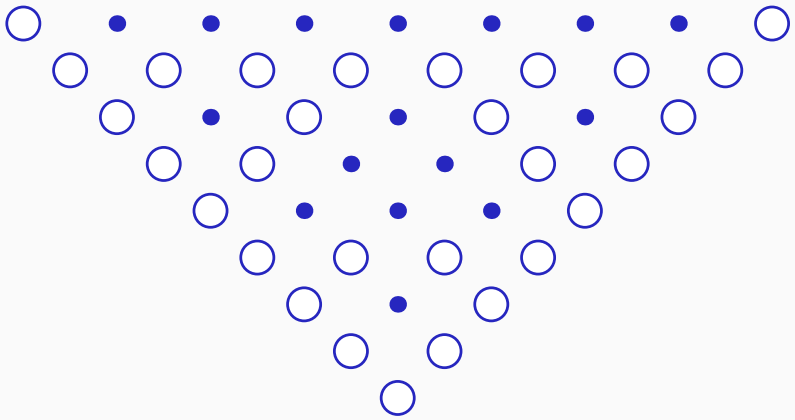


Нули заштрихованы, в остальных клетках стоят 1.

Автомат Паскаля и биномиальные коэффициенты

Диаграмма работы автомата Паскаля — это по сути инвертированная таблица чётности биномиальных коэффициентов.

Сдвинем ячейки так, чтобы новое состояние выражалось через состояния ближайших соседей снизу (слева и справа).



Нули изображены кружочками.

Свойства автомата Паскаля

Задача (непериодичность конфигураций)

Докажите, что вдоль любой прямой

$$\{(x,t) : x = a_1 + b_1\lambda, t = a_2 + b_2\lambda,\ a_i, b_j > 0,\ b_1 < b_2,\ \lambda \in \mathbb{N}\}$$

последовательность конфигураций автомата Паскаля непериодична.

(Начальная конфигурация содержит ровно один 0.)

Задача (эффективное вычисление конфигураций)

Докажите, что существует полиномиальный алгоритм, который по двоичным записям x, t находит $c(x,t)$: состояние ячейки x в момент времени t работы автомата Паскаля.

(Начальная конфигурация содержит ровно один 0.)

Свойства автомата Паскаля

Задача (непериодичность конфигураций)

Докажите, что вдоль любой прямой

$$\{(x,t) : x = a_1 + b_1\lambda, t = a_2 + b_2\lambda,\ a_i, b_j > 0,\ b_1 < b_2,\ \lambda \in \mathbb{N}\}$$

последовательность конфигураций автомата Паскаля непериодична.

(Начальная конфигурация содержит ровно один 0.)

Задача (эффективное вычисление конфигураций)

Докажите, что существует полиномиальный алгоритм, который по двоичным записям x, t находит $c(x,t)$: состояние ячейки x в момент времени t работы автомата Паскаля.

(Начальная конфигурация содержит ровно один 0.)

Моделирование работы МТ на КА

T ячеек   *T* ячеек

<table>
<tr><td>…</td><td>$\langle \Lambda, \Lambda \rangle$</td><td>$\langle q_0, w_1 \rangle$</td><td>$\langle \Lambda, w_2 \rangle$</td><td>…</td><td>…</td><td>$\langle \Lambda, w_n \rangle$</td><td>$\langle \Lambda, \Lambda \rangle$</td><td>…</td></tr>
<tr><td colspan="9">………………………………………………………………</td></tr>
<tr><td colspan="9">⋮………………⋮………………⋮………………⋮</td></tr>
<tr><td colspan="2">… …</td><td>$\langle \Lambda, c \rangle$</td><td>…</td><td>$\text{cell}_{-1}$</td><td>$\text{cell}_0$</td><td>$\text{cell}_1$</td><td colspan="2">… …</td></tr>
<tr><td colspan="2">… …</td><td>$\langle \Lambda, c \rangle$</td><td colspan="2">… …</td><td>cell</td><td colspan="3">… …</td></tr>
<tr><td colspan="9">⋮………………⋮………………⋮………………⋮</td></tr>
<tr><td colspan="9">………………………………………………………………</td></tr>
<tr><td colspan="3">… …</td><td>$\langle q_f, a \rangle$</td><td colspan="5">… … …</td></tr>
</table>

Моделирование работы МТ на КА (продолжение)

Работа МТ $M = (A, \Lambda, Q, q_0, Q_f, \delta)$ моделируется КА с алфавитом $(Q \cup \{\Lambda\}) \times A$, размером окрестности 1 и модифицированной функцией переходов Δ.

Свойство стабильности

$\Delta((\Lambda, \Lambda), (\Lambda, \Lambda), (\Lambda, \Lambda)) = (\Lambda, \Lambda)$: пустое значение первой компоненты в паре означает отсутствие головки в данной ячейке ленты МТ, поэтому символ в этой ячейке не изменяется.

Моделирование работы МТ на КА (продолжение)

Работа МТ $M = (A, \Lambda, Q, q_0, Q_f, \delta)$ моделируется КА с алфавитом $(Q \cup \{\Lambda\}) \times A$, размером окрестности 1 и модифицированной функцией переходов Δ.

Свойство стабильности

$\Delta((\Lambda, \Lambda), (\Lambda, \Lambda), (\Lambda, \Lambda)) = (\Lambda, \Lambda)$: пустое значение первой компоненты в паре означает отсутствие головки в данной ячейке ленты МТ, поэтому символ в этой ячейке не изменяется.

Моделирование работы КА на двоичном КА

Работа КА $C = (A, r, \delta)$ моделируется работой двоичного КА $C' = (\{0, 1\}, r', \delta')$ с увеличенным размером окрестности

$$r' = (r + 1)(|A| + 2).$$

Идея моделирования

Кодируем символы алфавита A двоичными словами.

Однако

функция переходов клеточного автомата одна и та же для всех ячеек, независимо от их места в кодировке.

Нужно выбирать кодировку с учётом этого обстоятельства.

Моделирование работы КА на двоичном КА

Работа КА $C = (A, r, \delta)$ моделируется работой двоичного КА $C' = (\{0, 1\}, r', \delta')$ с увеличенным размером окрестности

$$r' = (r + 1)(|A| + 2).$$

Идея моделирования

Кодируем символы алфавита A двоичными словами.

Однако

функция переходов клеточного автомата одна и та же для всех ячеек, независимо от их места в кодировке.

Нужно выбирать кодировку с учётом этого обстоятельства.

Моделирование работы КА на двоичном КА: кодировка

Без ограничения общности $A = \{0, 1, \ldots, \ell - 1\}$, пустой символ $\Lambda = 0$.

Кодировка

$$a \mapsto \varphi(a) = 1^{1+\ell-a} 0^a 1.$$

Так как $\varphi(0) = 1^{\ell+2}$, пустой символ двоичного КА обязан равняться 1.

Лемма

Пусть $\varphi(c)$ — посимвольная кодировка конфигурации c КА C. Тогда по $(r+1)(\ell+2)$-окрестности любого бита в $\varphi(c)$ однозначно восстанавливается r-окрестность того символа c, коду которого принадлежит данный бит.

Моделирование работы КА на двоичном КА: кодировка

Без ограничения общности $A = \{0, 1, \ldots, \ell - 1\}$, пустой символ $\Lambda = 0$.

Кодировка

$$a \mapsto \varphi(a) = 1^{1+\ell-a} 0^a 1.$$

Так как $\varphi(0) = 1^{\ell+2}$, пустой символ двоичного КА обязан равняться 1.

Лемма

Пусть $\varphi(c)$ — посимвольная кодировка конфигурации c КА C. Тогда по $(r+1)(\ell+2)$-окрестности любого бита в $\varphi(c)$ однозначно восстанавливается r-окрестность того символа c, коду которого принадлежит данный бит.

Доказательство леммы (наблюдение)

Для краткости $L = \ell + 2$ — длина кода символа.

Наблюдение за пустотой

Если в r-окрестности символа из конфигурации c есть хотя бы один ненулевой символ, то в $(r+1)L$-окрестности любого бита из кодировки этого символа есть нулевой.

Действительно, максимально возможное расстояния между битами кодов символов из r-окрестности равно $L - 1 + (r-1)L + L - 1 = (r+1)L - 2$.

Доказательство леммы (наблюдение)

Для краткости $L = \ell + 2$ — длина кода символа.

Наблюдение за пустотой

Если в r-окрестности символа из конфигурации c есть хотя бы один ненулевой символ, то в $(r+1)L$-окрестности любого бита из кодировки этого символа есть нулевой.

Действительно, максимально возможное расстояния между битами кодов символов из r-окрестности равно $L - 1 + (r-1)L + L - 1 = (r+1)L - 2$.

Доказательство леммы (процедура восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1+\ell-a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

Доказательство леммы (процедура восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1+\ell-a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

Доказательство леммы (процедура восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1+\ell-a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

Доказательство леммы (процедура восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1+\ell-a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

Доказательство леммы (процедура восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1+\ell-a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

Доказательство леммы (корректность восстановления)

1. Ищем 0 в $(r+1)L$-окрестности данного бита b в конфигурации $\varphi(c)$.
2. Если нулей нет, то считаем, что вся окрестность состоит из одних нулей.

Корректность этого шага следует из наблюдения за пустотой.

Доказательство леммы (корректность восстановления)

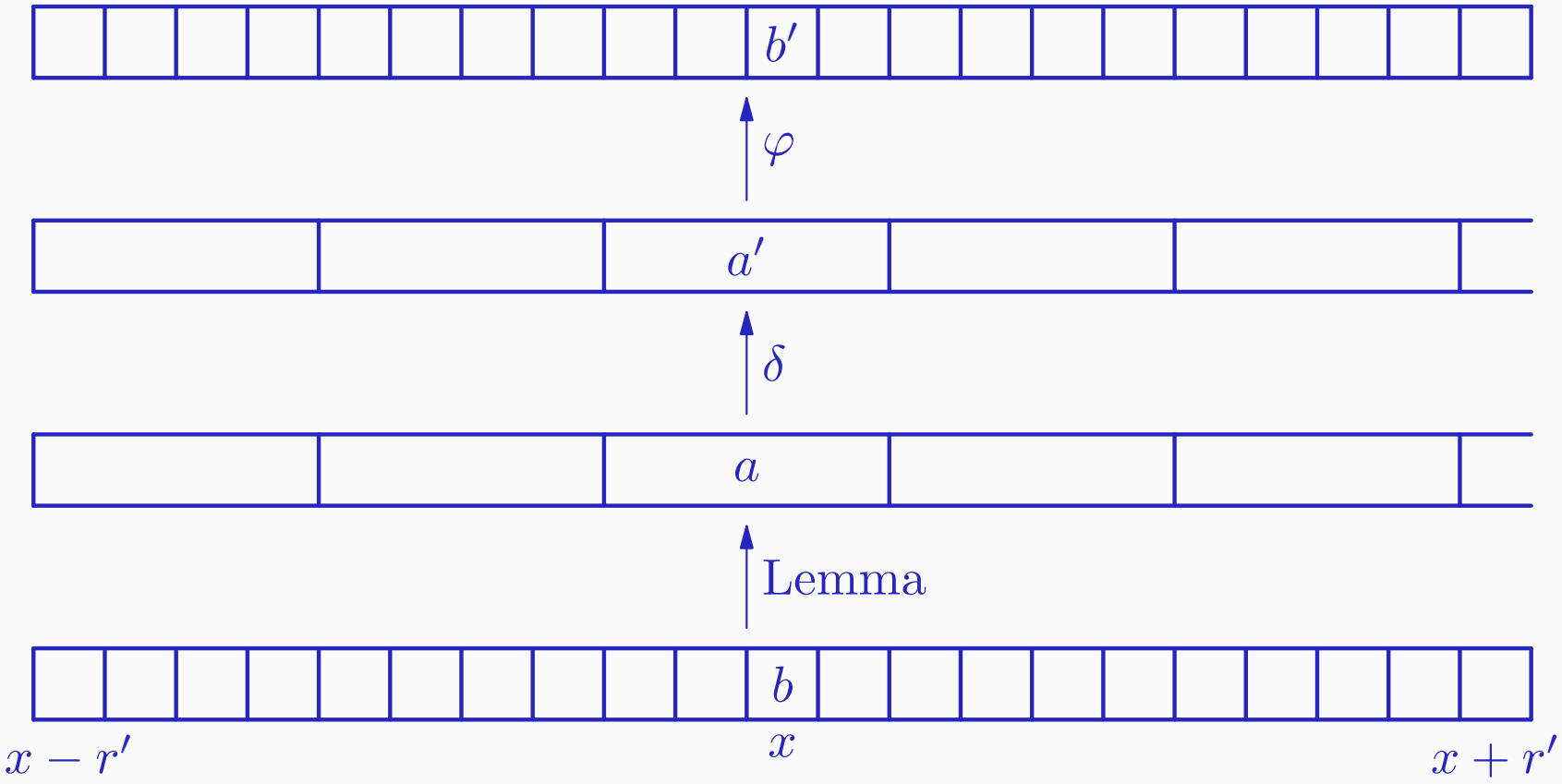
3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1 + \ell - a$ единиц слева и одну справа. Получаем код символа.
   Корректность: в коде конфигурации любой максимальной серии из a нулей предшествует не менее $1 + \ell - a$ единиц (входящих в код символа a, которому принадлежит эта серия).

Доказательство леммы (корректность восстановления)

3. В противном случае находим ближайший к исходному биту b ноль и максимальную серию 0^a нулей, которая его содержит.
4. Добавляем к этой серии $1 + \ell - a$ единиц слева и одну справа. Получаем код символа.
5. Так как кодировка равномерная (длины всех символов одинаковы), восстанавливаем коды всех символов r-окрестности символа из конфигурации c, коду которого принадлежит бит b.

   Корректность: коды тех символов, которые лежат в искомой r-окрестности, попадают в $(r+1)L$-окрестность исходного бита b.

Моделирование работы КА на двоичном КА: функция δ'



Соответствие начальных конфигураций

МТ $\to$ КА $\to$ двоичный КА

Утверждение

Можно добиться, чтобы начальной конфигурации МТ при работе на пустом входе отвечала начальная конфигурация $\ldots 11011 \ldots$ двоичного КА.

Доказательство

Переход от МТ к КА: $(\Lambda, \Lambda) \mapsto 0$; $(q_0, \Lambda) \mapsto 1$.

Переход к двоичному КА: $0 \mapsto 1^{|A|+2}$; $1 \mapsto 1^{|A|}01$.

Здесь A — алфавит МТ, q_0 — начальное состояние МТ, а Λ — пустой символ.

Соответствие начальных конфигураций

МТ $\to$ КА $\to$ двоичный КА

Утверждение

Можно добиться, чтобы начальной конфигурации МТ при работе на пустом входе отвечала начальная конфигурация $\ldots 11011 \ldots$ двоичного КА.

Доказательство

Переход от МТ к КА: $(\Lambda, \Lambda) \mapsto 0$; $(q_0, \Lambda) \mapsto 1$.

Переход к двоичному КА: $0 \mapsto 1^{|A|+2}$; $1 \mapsto 1^{|A|}01$.

Здесь A — алфавит МТ, q_0 — начальное состояние МТ, а Λ — пустой символ.