PWPP не содержится в PLS или в PPA

Федорович Арсений

Физтех-школа прикладной математики и информатики
МФТИ

Октябрь 2025

Под руководством Мусатова Д. В.
МФТИ

Содержание

Введение

Принципы существования, позволяющие определять задачи поиска:

1. Принцип Дирихле: в отображении $f\colon [1, N+1] \to [1, N]$ есть коллизия.
2. В ориентированном ациклическом графе есть сток
3. Лемма о рукопожатиях: если в графе есть одна вершина нечётной степени, то есть и другая
4. Если орграфе есть одна несбалансированная вершина, то есть и другая
5. Теорема Рамсея: в графе на $N = 2^{2n}$ вершинах найдётся либо клика размера n, либо независимое множество размера n.
6. Теорема Мантеля: в графе на N вершинах с $> \frac{N^2}{4}$ ребрами есть треугольник

И так далее.

Введение

Пусть $N = 2^n$, где n — размер входа x. Для определения классов PWPP, PPA и PLS опишем полные задачи в них.

PWPP: Weak Pigeon. Дана $f\colon [1, 2N] \to [1, N]$. Найти коллизию.

PPA: Lonely. Дан неориентированный граф на $2N$ вершинах, одна вершина изолирована, остальные образуют пары, за исключением как минимум ещё одной изолированной вершины. Найти другую изолированную вершину.

PLS: Sink-of-DAG. Дан ациклический ориентированный граф на N вершинах, для любой вершины u можно вычислить её единственного потомка v и функцию потенциала, возрастающую вдоль ребра (u, v). Найти сток.

Мы будем использовать другую PLS-полную задачу — Iter. Дана $f\colon [1, N] \to [1, N]$, такая что $f(v) \geq v$ для всех v. Решениями задачи Iter являются:

1. 1, если $f(1) = 1$;
2. Такое $v \in [1, N]$, что $v < f(v)$ и $f(f(v)) = f(v)$.

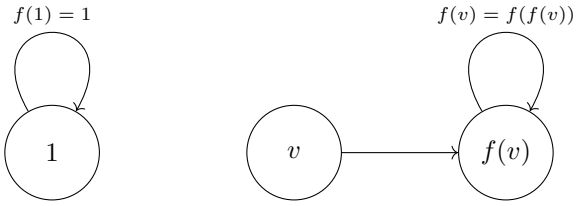


Рис. 1: Решения задачи Iter

Модель

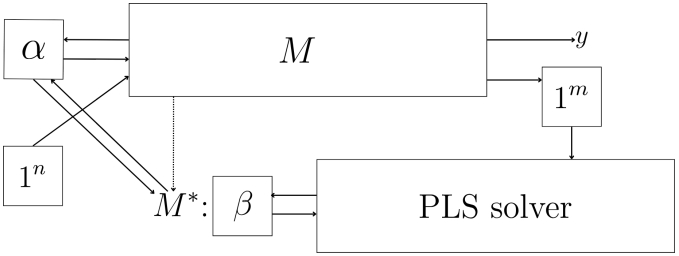


Рис. 2: Модель

Пусть f из определения задачи Weak Pigeon задаётся оракулом α: для любого $v \in [1, 2N]$ $\alpha(v) = f(v)$.

Машина M выдает вход 1^m, где $m = \text{poly}(n)$, для решателя задачи Iter, а также предоставляет PLS-решателю доступ к другой машине M^*, которая эмулирует β для задачи Iter.

Решатель задачи Iter в процессе работы задает вопросы о значении f из задачи Iter на каком-то $c \in [1, N]$. Для этого он делает запрос к оракулу β и узнает $\beta(c)$.

Любой запрос $\beta(c)$ к оракулу β состоит из последовательности запросов к оракулу α и может быть представлен в виде дерева $T(c)$. Любое дерево $T(c)$ имеет полиномиальную глубину, ограниченную временем работы машины M^*, которое, в свою очередь, ограничено временем работы машины M, а машина M работает за полиномиальное время.

Модель

Пример дерева $T(c)$ для какого-то $c \in [1, N]$. Для каждого c оно свое. В дальнейшем мы будем рассматривать такие деревья для всех c, то есть $T(1), \ldots, T(N)$.

Запрос $\beta(c)$ в данном случае начинается с вопроса $\alpha(1) =?$. В зависимости от значения $\alpha(1)$ задается еще один вопрос к α, например $\alpha(3) =?$ или $\alpha(2) =?$ и так далее.

В листе находится посчитанное на определенной ветви вычислений значение $\beta(c)$.

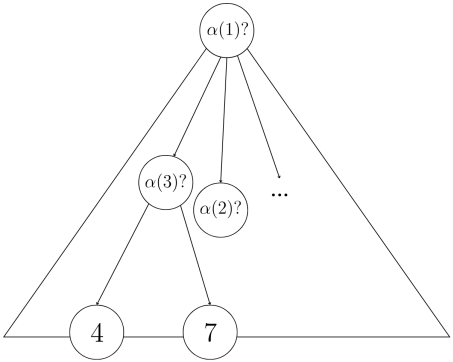


Рис. 3: Пример дерева $T(c)$

PWPP не содержится в PLS

Цель — показать, что Weak Pigeon не сводится к Iter. Рассмотрим деревья $T(c)$ для всех c. Мы удаляем те ветви $T(c)$, которые: (i) раскрывают ответ на задачу Weak Pigeon, то есть коллизию; (ii) являются самопротиворечивыми — содержат два различных ответа на один и тот же запрос $\alpha(v)$; или (iii) противоречат текущему частичному определению α. При этом каждая вершина в $T(c)$ будет все еще иметь много исходящих рёбер.

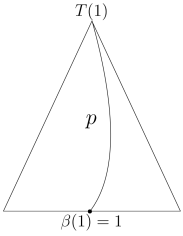
Пути p и p' называются согласованными, если ответы на одинаковые запросы к α у них совпадают.

Возможны два случая:

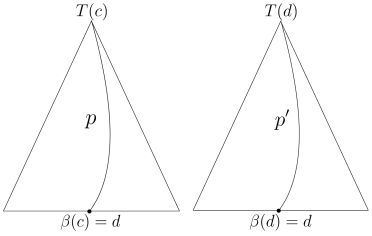
Случай 1. Выполняется хотя бы одно из условий:

1. В $T(1)$ есть путь p, заканчивающийся листом с меткой 1; или
2. Существуют два согласованных пути: p в $T(c)$ и p' в $T(d)$ при $c < d$, такие что оба пути заканчиваются листом с меткой d.

Случай 2. Ни одно из условий не выполняется, что приводит к противоречию.



(a) Дерево $T(1)$



(b) Деревья $T(c)$ и $T(d)$

В Случае 1 мы расширяем текущее частичное определение α, включая в него запросы из p и p'. Тем самым, мы задаем еще полиномиальное число вопросов к α.

Итого, мы смогли построить α так, что мы не раскрываем коллизию в Weak Pigeon, однако найдено решение на задачу Iter, то есть:

1. 1, если $\beta(1) = 1$;
2. Такое $c \in [1, N]$, что $c < \beta(c) = d$ и $\beta(d) = \beta(\beta(c)) = \beta(c) = d$.

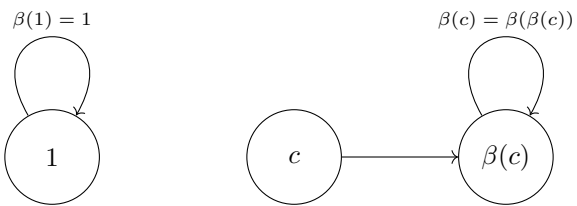


Рис. 5: Найденные решения задачи Iter (из Случая 1)

PWPP не содержится в PPA

Теперь покажем, что Weak Pigeon не сводится к задаче Lonely. Теперь у нас имеются деревья для задачи Lonely, то есть для $c \in [1, 2N]$ заданы деревья $T(c)$. Каждый лист дерева $T(c)$ помечен либо c', либо символом $\varnothing$, что означает, что $\beta(c) = c'$ (то есть c соединена с c') или $\beta(c) = \varnothing$ (то есть c — одинокая вершина) соответственно.

Мы удаляем ветви дерева $T(c)$ так же, как и в предыдущем случае.

Случай 1. Существует дерево $T(c)$ с листом, помеченным $\varnothing$, что означает, что c — одинокая вершина. Как и в случае с PLS, мы добавляем в α запросы вдоль пути p.

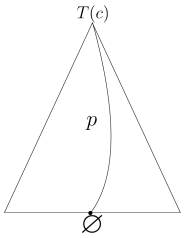


Рис. 6: Согласно пути p c — одинокая вершина

Случай 2. Не существует дерева $T(c)$ с листом, помеченным $\varnothing$. Мы строим другую последовательность деревьев $\mathcal{T}^*$.

Nullstellensatz Degree Lower Bound

Опишем систему $\mathcal{PHP}_N^{2N}$ состоит из полиномиальных уравнений относительно переменных $x_{i,j}$, где $i \in [1, 2N]$, $j \in [1, N]$. Переменные $x_{i,j}$ означают наличие или отсутствие ребра из i в j, то есть $x_{i,j} = 1 \Leftrightarrow f(i) = j$.

1. $\sum_{j=1}^{N} x_{i,j} - 1 = 0, \quad \forall i \in [1, 2N].$
2. $x_{i,j} x_{i,k} = 0, \quad \forall i \in [1, 2N],\ \forall j \neq k \in [1, N].$
3. $x_{i,k} x_{j,k} = 0, \quad \forall k \in [1, N],\ \forall i \neq j \in [1, 2N].$

Пусть $\mathcal{PHP}_N^{2N} = \{Q_i(\bar{x}) = 0\}$. Обозначим через l максимальную глубину деревьев в $\mathcal{T}^*$.

Если семейство $\mathcal{T}^*$ существует, то существуют и многочлены P_i степени $\leq l - 1$ такие, что

$$\sum_i P_i(\bar{x}) Q_i(\bar{x}) = 1.$$

Цель состоит в том, чтобы построить отображение, называемое **дизайном**, и применить его к обеим частям этого тождества, что приведёт к противоречию.

k-дизайном называется линейное отображение $D \colon \mathbb{F}_2[x_1, x_2, \ldots]_{\leq k} \to \mathbb{F}_2$ такое, что:

1. $D(1) = 1$;
2. $D(PQ_i) = 0$ для любого P, такого что $\deg(P) + \deg(Q_i) \leq k$.

Альтернативные способы доказательства несводимости

Можно построить формулу, утверждающую что у задачи нет решения. Тогда чтобы доказать, что задача не лежит в классе, достаточно показать, что у формулы нет короткого опровержения. В PPA и PPAD это можно сделать, построив дизайн. В некоторых случаях (напр. для PPP) можно строить псевдоматожидания.

Показаны некоторые разделения между системами доказательств, из которых следуют разделения для классов.

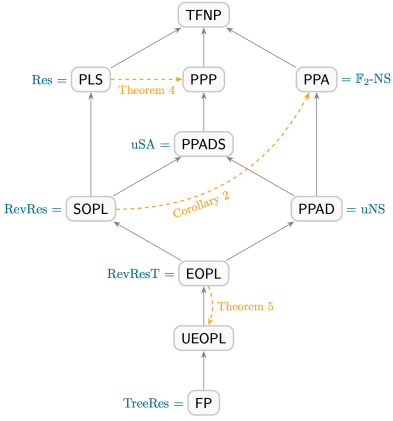


Рис. 7: Диаграмма некоторых классов в TFNP. Стрелка $A \to B$ означает $A \subseteq B$. Стрелка $A \dashrightarrow B$ означает $A \not\subseteq B$ относительно какого-то оракула. Изображены не все вложения. Голубым обозначены соответствующие классам пропозициональные системы доказательств

Список литературы

Paul Beame et al.

The Relative Complexity of NP Search Problems.
Journal of Computer and System Sciences, 57(1):3--19, 1998.
ISSN 0022-0000.
doi: 10.1006/jcss.1998.1575.
URL https://www.sciencedirect.com/science/article/pii/S0022000098915756.

Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao.

Separations in proof complexity and tfnp.
J. ACM, 71(4), July 2024.
doi: 10.1145/3663758.
URL https://doi.org/10.1145/3663758.

Tsuyoshi Morioka.

Classification of search problems and their definability in bounded arithmetic, mar 2001.