Санкт-Петербургский государственный университет

Кафедра системного программирования

# Расширение возможностей профилировщика данных Desbordante по работе с графовыми зависимостями

Черников Антон Александрович, группа 23.М04-мм

3 октября 2024 г.

**Научный руководитель:** ассистент кафедры ИАС Г.А. Чернышев

Санкт-Петербург
2024

# Введение: Desbordante

Desbordante — это инструмент для профилирования данных, разрабатываемый группой студентов под руководством Г. А. Чернышева.

- Способен обнаруживать закономерности в данных с использованием различных алгоритмов
- Весь проект имеет открытый исходный код
- Высокопроизводителен

# Введение: функциональные зависимости

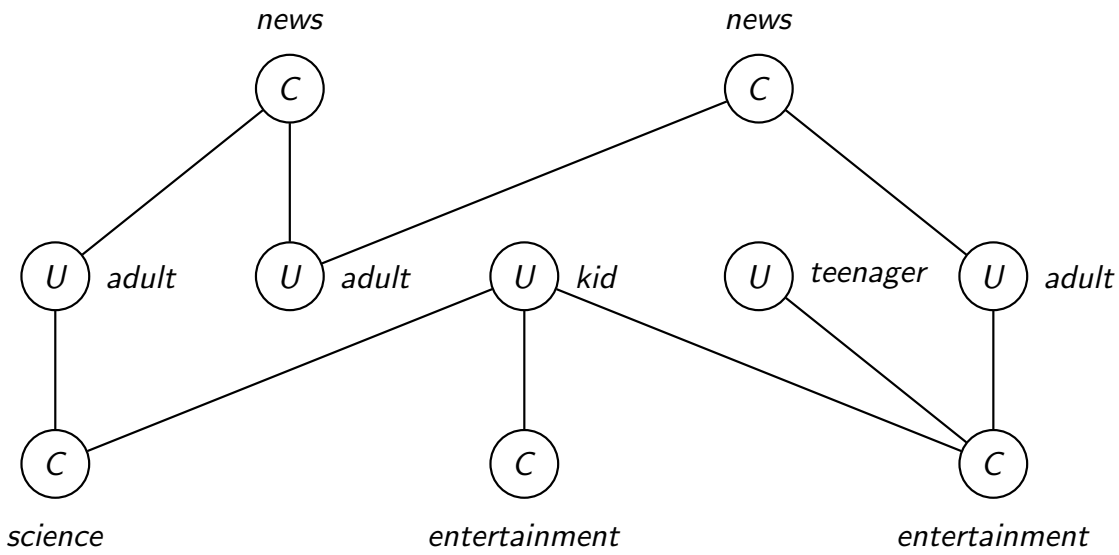
Определение: Отношение $R$ удовлетворяет функциональной зависимости $X \to Y$ (где $X, Y \subset R$) тогда и только тогда, когда для любых кортежей $t_1,\ t_2 \in R$ выполняется: если $t_1[X] = t_2[X]$, то $t_1[Y] = t_2[Y]$.

Данные о студентах и их оценках

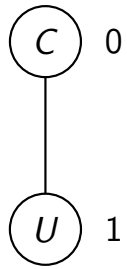
| ID | Name | Course | Grade |
|---|---|---|---|
| 1 | Alice | Math | A |
| 2 | Bob | Math | B |
| 3 | Charlie | Science | A |
| 1 | Alice | Science | B |
| 2 | Bob | Science | A |

Зависимость $\{ID \to Name\}$ выполняется, а зависимость $\{Name \to Course\}$ — нет.

# Введение: графовые функциональные зависимости




Связь каналов $C$ с пользователями $U$
Атрибуты у $C$ — $\{topic\}$, у $U$ — $\{age\_group\}$




$\{0.topic = news\} \rightarrow \{1.age\_group = adult\}$

Графовая зависимость[1]

[1] "Functional Dependencies for Graphs", Wenfei Fan, Yinghui Wu, Jingbo Xu, SIGMOD'16

# Предыдущие работы

- Реализован алгоритм **проверки** графовых зависимостей в трёх вариациях:
  - Наивный
  - Базовый
  - Эффективный
- Реализована возможность запускать их на Python и через консоль, созданы примеры работы алгоритмов
- Выполнен обзор алгоритма **поиска** графовых зависимостей

Следующий этап — реализация изученного алгоритма **поиска**.

# Постановка задачи

**Целью** работы является расширение инструментария Desbordante для работы с графовыми зависимостями.
Для достижения этой цели были поставлены следующие **задачи**:

- Реализовать алгоритм поиска графовых функциональных зависимостей и произвести тестирование
- Реализовать возможность запускать алгоритм поиска графовых зависимостей из скриптов, написанных на языке программирования Python
- Создать скрипты-примеры работы алгоритма поиска графовых зависимостей на языке программирования Python
- Обеспечить возможность запускать алгоритм поиска графовых зависимостей через консоль путём реализации соответствующей подсистемы

# Алгоритм поиска графовых зависимостей[2]

Входные параметры:

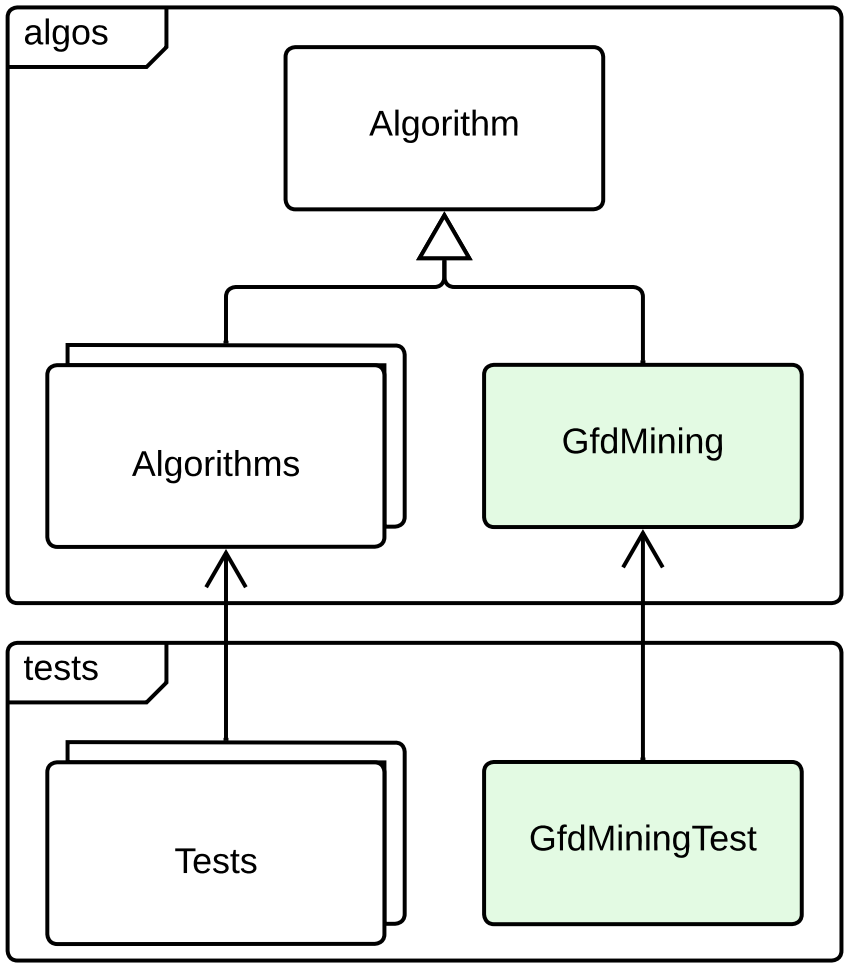
- $G$ — граф
- $k$ — максимальное количество вершин, которое ожидается от паттернов найденных зависимостей
- $\sigma$ — сколько вложений в граф минимально должен иметь паттерн найденной зависимости

Алгоритм является итеративным. Описание одной итерации:

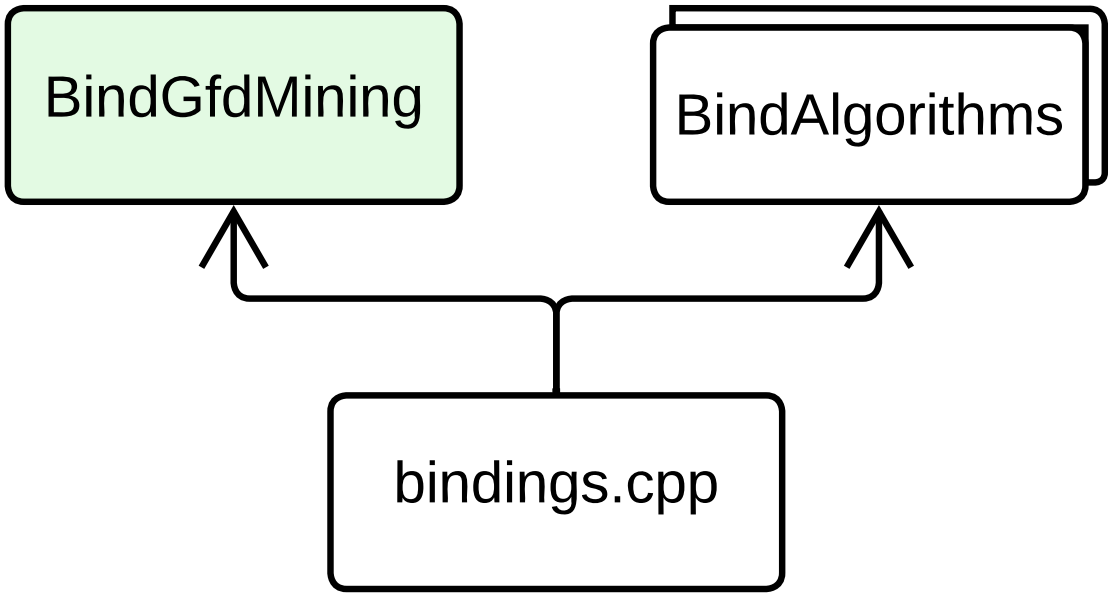
- Горизонтальная генерация: создание всевозможных правил литералов, проверка
- Вертикальная генерация: создание нового кластера паттернов путём добавления ребра
- Фильтрация паттернов, количество вложений которых больше $\sigma$

[2] "Discovering Graph Functional Dependencies", Wenfei Fan, Chunming Hu, Xueli Liu, Ping Lu, SIGMOD'18

# Диаграмма классов для реализованного алгоритма



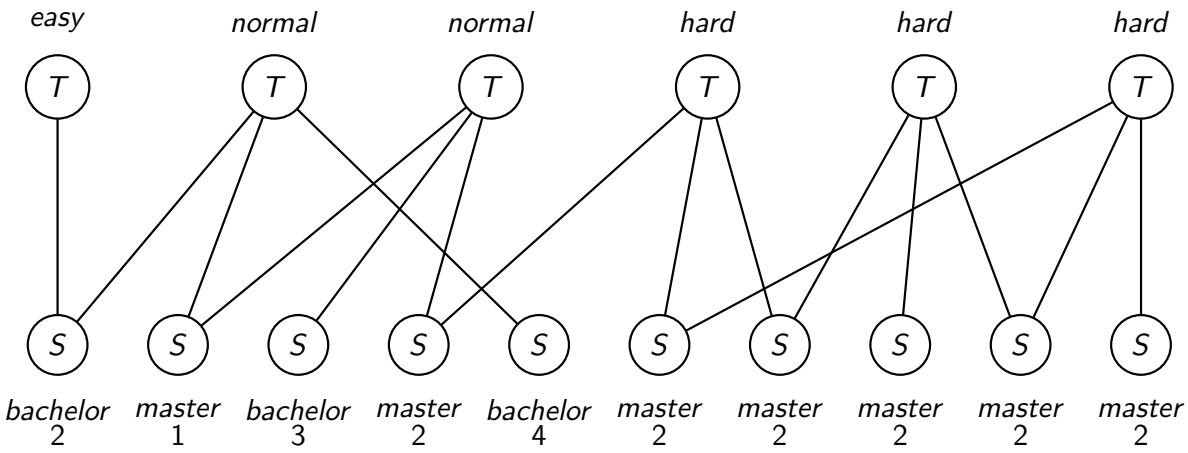
# Схема привязок алгоритмов Desbordante к Python



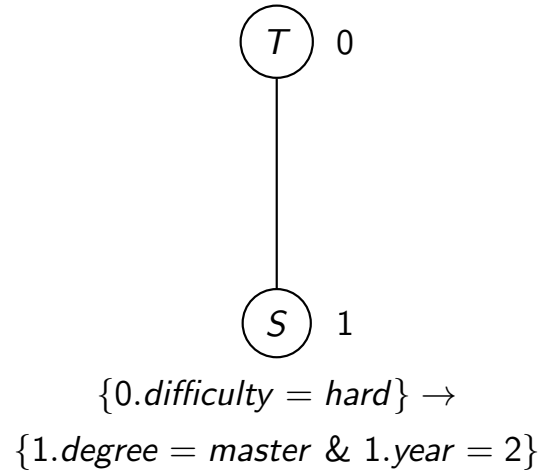
# Примеры работы алгоритма поиска графовых зависимостей

- В Desbordante нет технического писателя
- Отсутствует документация по пользованию алгоритмами
- Существует необходимость снабжать каждый алгоритм примером на Python

# Примеры работы алгоритма поиска графовых зависимостей




Пример графа




Графовая зависимость

# Примеры: вывод в консоль

```
Figure provides an example of a graph. The following abbreviations were used here: T - task, S -
 student. The vertices with the T-label have the attributes "name" and "difficulty", the vertice
s with the S-label have the "name", "degree" and "year" attributes, which indicate the student's
 name, level of education and year. The values of these attributes are signed next to the vertic
es, except for the name, since it is not informative.

Let's run the algorithm. We'll specify 2 as the k parameter to look for patterns with no more th
an two vertices, and we'll specify 3 as the sigma to exclude rare dependencies.

Desbordante > Mined GFDs: 1

The dependency found indicates that only second-year master's students are working on the hard t
ask.

Close the image window to finish.
```

# Python CLI

- Взаимодействие с python-модулем desbordante производится с помощью модуля Click
- Командная строка имеет набор опций: help, task, algo
- Для алгоритма поиска GFD были сделаны три уникальные опции: graph, k и sigma

Пример вызова алгоритма поиска через Python консоль:

```
$ desbordante --task=gfd_mining --algo=gfd_miner
--graph=examples/graph.dot --gfd_k=3 --gfd_sigma=10
```

# Тестирование

Для апробации алгоритма было добавлено 5 тестов на небольших графах в среднем имеющих 13 вершин и 19 рёбер.
Тесты направлены на:

- Выявление минимальных графовых зависимостей
- Обработку графов, не имеющих зависимостей
- Получение графовых зависимостей с заключениями, имеющими более одного литерала
- Корректную работу на синтетических данных

# Результаты работы

- Реализован алгоритм поиска графовых функциональных зависимостей и произведено тестирование
- Реализована возможность запускать алгоритм поиска графовых зависимостей из скриптов, написанных на языке программирования Python
- Созданы скрипты-примеры работы алгоритма поиска графовых зависимостей на языке программирования Python
- Обеспечена возможность запускать алгоритм поиска графовых зависимостей через консоль путём реализации соответствующей подсистемы

Исходный код доступен на GitHub[3][4].

[3] https://github.com/Desbordante/desbordante-core/pull/465
[4] https://github.com/Desbordante/desbordante-cli/pull/5