Санкт-Петербургский государственный университет

Кафедра системного программирования

Расширение возможностей профилировщика данных Desbordante по работе с графовыми зависимостями

Черников Антон Александрович, группа 23.М04-мм

2 мая 2024 г.

Научный руководитель: ассистент кафедры ИАС Г.А. Чернышев

Санкт-Петербург
2024

Постановка задачи

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

- Разработать подсистему, позволяющую запускать валидатор графовых зависимостей из скриптов, написанных на языке программирования Python
- Создать скрипты-примеры работы валидатора графовых зависимостей на языке программирования Python
- Обеспечить возможность запускать валидатор графовых зависимостей через консоль путём реализации соответствующей подсистемы
- Выполнить обзор алгоритма поиска графовых функциональных зависимостей и описать его основные свойства

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, так как реализован на C++

Введение: функциональные зависимости

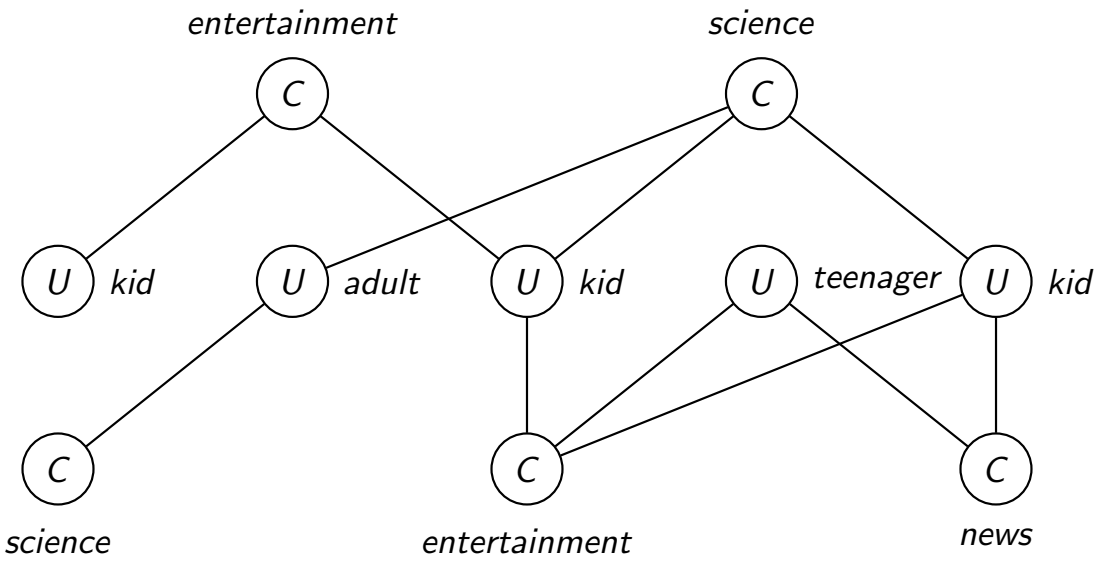
Определение: Отношение R удовлетворяет функциональной зависимости $X \to Y$ (где $X, Y \subset R$) тогда и только тогда, когда для любых кортежей $t_1,\ t_2 \in R$ выполняется: если $t_1[X] = t_2[X]$, то $t_1[Y] = t_2[Y]$.

Характеристики мобильных устройств

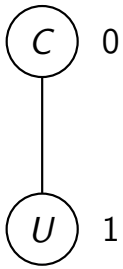
name	OS	memory	bluetooth_codec
Honor 20	Android	128 GB	SBC
iPhone 14	iOS	128 GB	AAC
Redmi Note 8t	Android	64 GB	SBC
Realme 8	Android	128 GB	SBC

Зависимость $\{OS \to bluetooth\_codec\}$ выполняется, а зависимость $\{OS \to memory\}$ — нет.

Введение: графовые функциональные зависимости



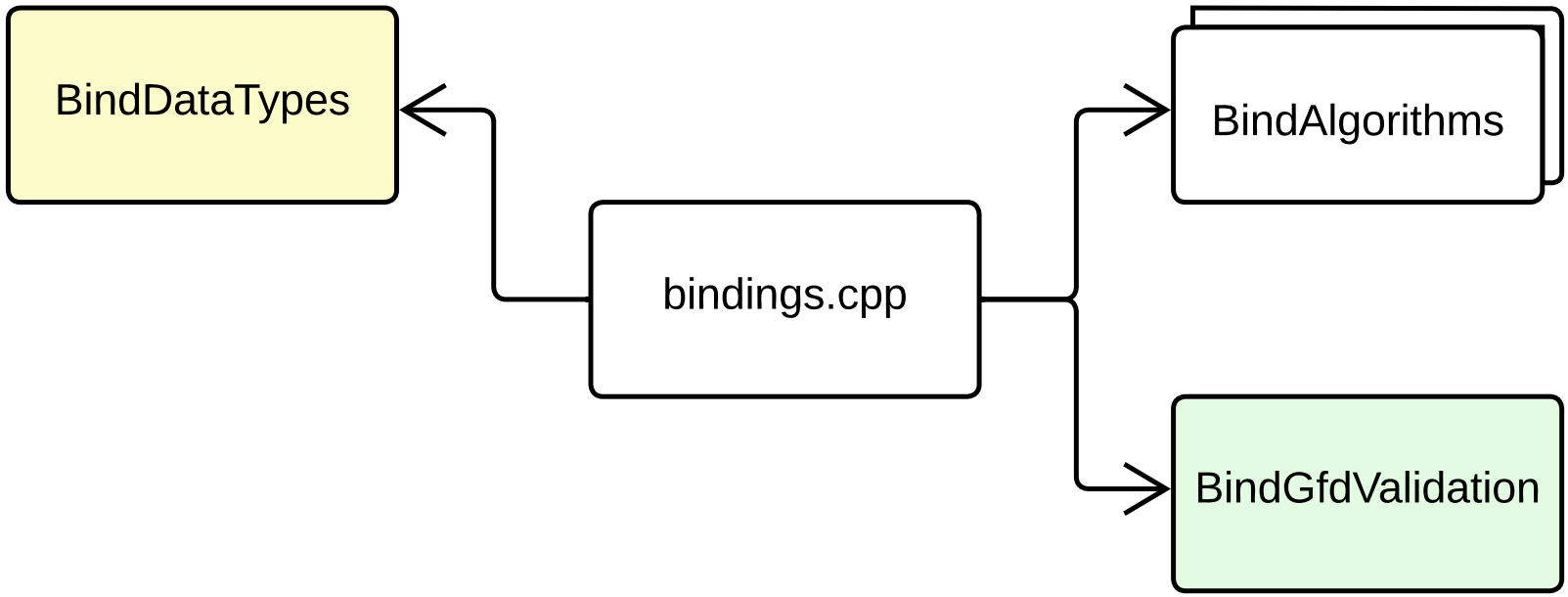
Связь каналов C с пользователями U
Атрибуты у C — $\{topic\}$, у U — $\{age\_group\}$



Графовая зависимость[1]
$\{0.topic = entertainment \rightarrow 1.age\_group = kid\}$

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

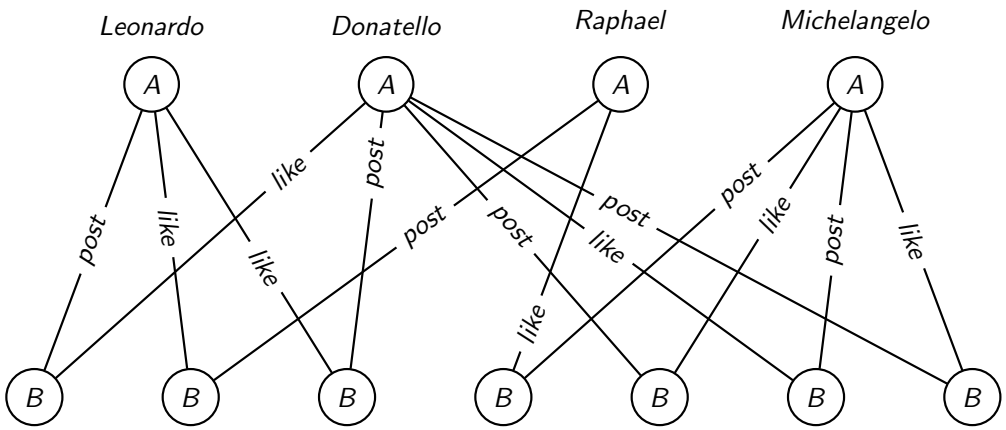
Схема привязок алгоритмов Desbordante к Python



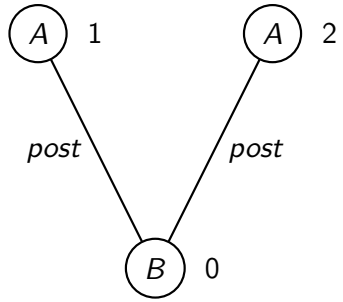
Примеры работы валидатора графовых зависимостей

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

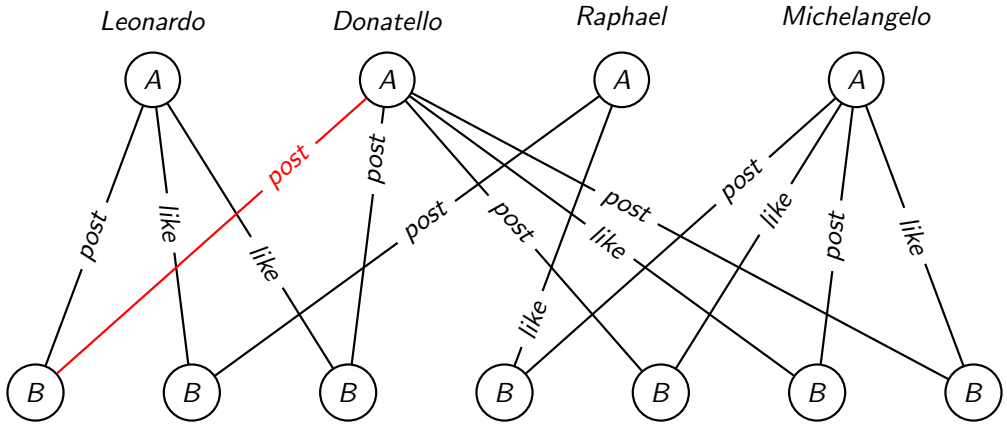
Примеры работы валидатора графовых зависимостей



Пример графа



Графовая зависимость
$\{\varnothing \to 1.name = 2.name\}$



Модифицированный граф

Примеры: вывод в консоль

```
The graph is depicted in the figure. The following abbreviations were used: A - account, B - blog. Vertices l
abeled A have a "name" attribute showing the nickname; vertices labeled B - "author", indicating who wrote th
e blog. The values of these attributes are labeled next to the vertices. The edges are also labeled as: "post
", which indicates who wrote the blog, and "like", which indicates approval by another person. In the drawing
, the edges are marked "post" in bold.

The dependency on the figure suggests that one blog cannot have two authors. That is, satisfiability of this
dependency ensures that there are no errors related to the number of authors in the data.

Let's check if this dependency holds.

Desbordante > GFD holds.

Well, GFD is really satisfied as expected.

Close the image window to continue.

Let's now modify the graph to see how the algorithm will behave in another case. The new graph is depicted in
 the new figure. Replaced edge label between the first left blog and Donatello account from "like" to "post".

Run algorithm:

Desbordante > GFD does not hold.

As you can see, the modified graph does not satisfy this dependency, indicating that it has errors.

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 алгоритмов проверки GFD существуют две уникальные опции: graph и gfd

Пример вызова валидатора через Python консоль:

```
$ python3 cli/cli.py --task=gfd_verification
--algo=gfd_verifier --graph=examples/graph.dot
--gfd=examples/gfd.dot
```

Алгоритм поиска графовых зависимостей[2]

- Не всегда известно, какой именно структурой обладает интересующая зависимость
- Полезным инструментом является автоматическая генерация графовых зависимостей на основе данного графа
- Разбор статьи составляет вторую, теоретическую, задачу данной работы

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

Алгоритм поиска графовых зависимостей

Входные параметры:

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

Алгоритм является итеративным. Описание i-ой итерации алгоритма:

- Генерация всевозможных паттернов с количеством вершин, равным i, на основе паттернов предыдущей итерации
- Выбор только тех сгенерированных паттернов, чьё количество вложений не меньше σ
- Генерация для каждого паттерна всевозможных правил литералов
- Последовательная проверка сгенерированных зависимостей на выполнимость

Результаты работы

- Разработана подсистема, позволяющая запускать валидатор графовых зависимостей из скриптов, написанных на языке программирования Python
- Созданы скрипты-примеры работы валидатора графовых зависимостей на языке программирования Python
- Обеспечена возможность запускать валидатор графовых зависимостей через консоль путём реализации соответствующей подсистемы
- Выполнен обзор алгоритма поиска графовых функциональных зависимостей и описаны его основные свойства

Код этой работы был принят, интегрирован и вошёл в релиз Desbordante 2.0.0[3].

[3] https://github.com/Desbordante/desbordante-core/pull/379