

Методы и средства навигации при интерактивной визуализации структурированной информации

Золотухин Тимур Александрович НГУ, ФИТ

Научный руководитель: д.ф-м.н, профессор Касьянов В.Н.

Новосибирск, 2012

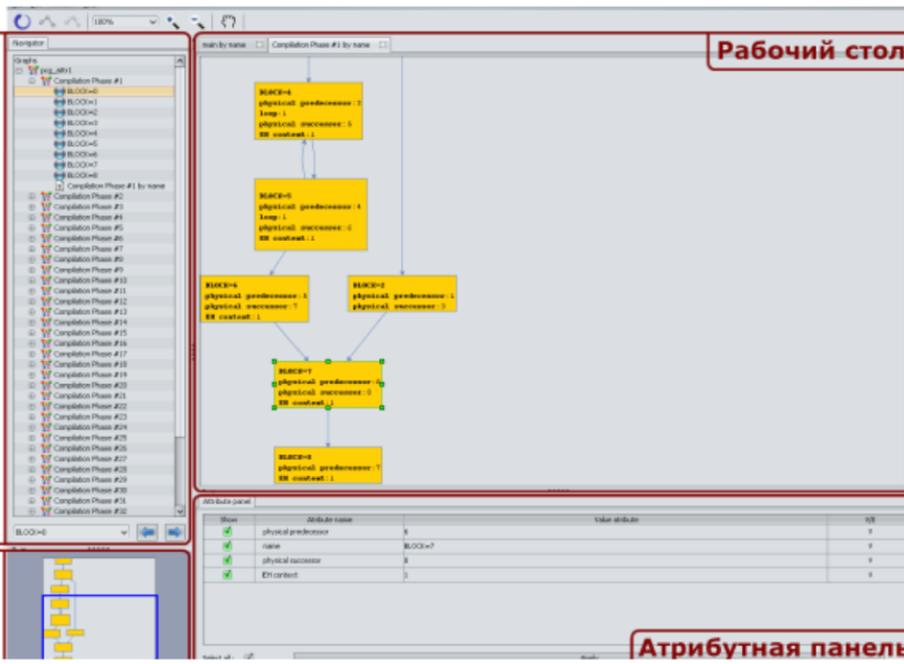
Происхождение задачи



- **Визуализация информации** - это процесс преобразования больших и сложных видов абстрактной информации в визуальную форму
- Для визуализации информации для данной задачи будем использовать графовые модели – иерархические атрибутированные графы
- Основные подзадачи при таком подходе:
 - Визуализация графовой модели
 - Навигация по графовой модели
- Существующие решения:
 - aiSee
 - yEd
 - Cytoscape

Описание системы и область применения

Visual Graph – кроссплатформенная, расширяемая система, которая предоставляет пользователю удобные средства визуализации атрибутированных иерархических графов с большим числом элементов, а так же навигацию по этим графам.



Область применения:

1. синтаксические деревья,
2. графы потоков управления,
3. графы вызовов



Тестовая эксплуатация:

1. Intel C compiler,
2. Dalvik VM

Средства навигации

main x graph_1 vs graph_2 x

node_1|node_1

node_2|node_2

node_3

node_4|node_4

node_5|node_5

node_6|node_6

node_7

node_8

Инструмент поиска максимального подграфа двух графов

Рабочий стол

Рабочая область

BLOCK=1
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=2
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=3
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=4
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=5
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=6
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=7
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

BLOCK=8
physical predecessor: 1
loop: 1
physical successor: 1
EH constant: 1

Навигационная панель

graph1

- Compilation Phase #1
- Compilation Phase #2
- Compilation Phase #3
- Compilation Phase #4
- Compilation Phase #5
- Compilation Phase #6
- Compilation Phase #7
- Compilation Phase #8
- Compilation Phase #9
- Compilation Phase #10
- Compilation Phase #11
- Compilation Phase #12
- Compilation Phase #13
- Compilation Phase #14
- Compilation Phase #15
- Compilation Phase #16
- Compilation Phase #17
- Compilation Phase #18
- Compilation Phase #19
- Compilation Phase #20
- Compilation Phase #21
- Compilation Phase #22
- Compilation Phase #23
- Compilation Phase #24
- Compilation Phase #25
- Compilation Phase #26
- Compilation Phase #27
- Compilation Phase #28
- Compilation Phase #29
- Compilation Phase #30
- Compilation Phase #31
- Compilation Phase #32

BLOCK=6



ПОЛЬЗОВАТЕЛЬ

Мини карта

Conditions for vertices:

((color green Delete) - +)

or

((color blue Delete)) - +

add condition

Задание условия для вершин

Conditions for edges:

add condition

Задание условия для ребер

Action

- Select in current tab
- Open in new tab
- Replace tab

Cancel Do

Attribute panel

Show	Attribute name	Value attribute	File
<input checked="" type="checkbox"/>	physical predecessor	1	?
<input checked="" type="checkbox"/>	name	BLOCK=7	?
<input checked="" type="checkbox"/>	physical successor	1	?
<input checked="" type="checkbox"/>	EH constant	1	?

Select all: Apply

Атрибутная панель

Разработка алгоритма нахождения максимального общего подграфа двух графов

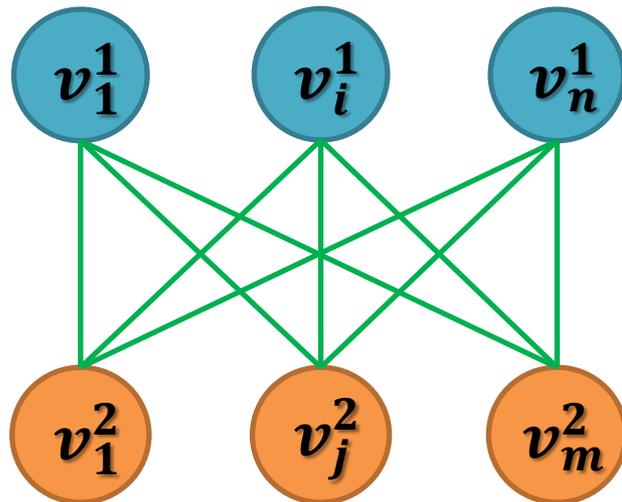
Шаг 1

$G_1(V_1, E_1)$

$G_2(V_2, E_2)$

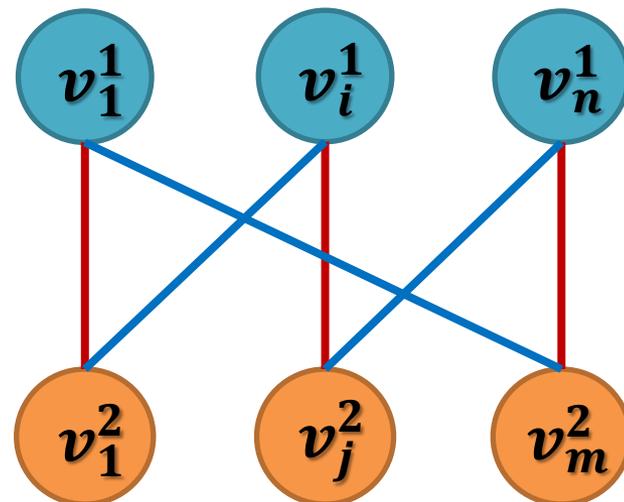
Вход

Шаг 2



Построение
двудольного графа

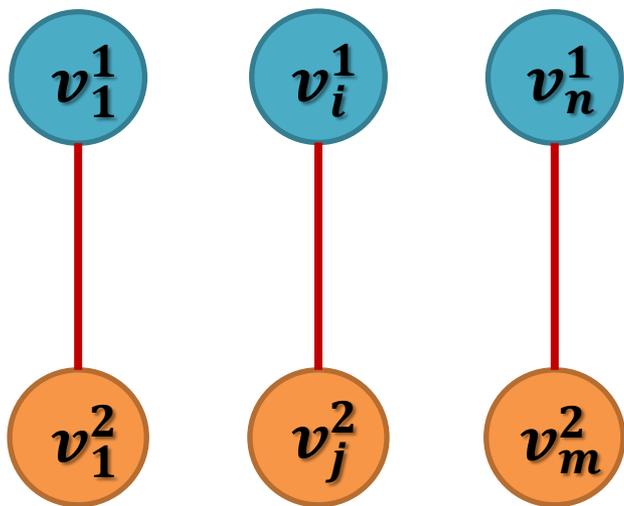
Шаг 3



Поиск максимального
паросочетания
максимального веса

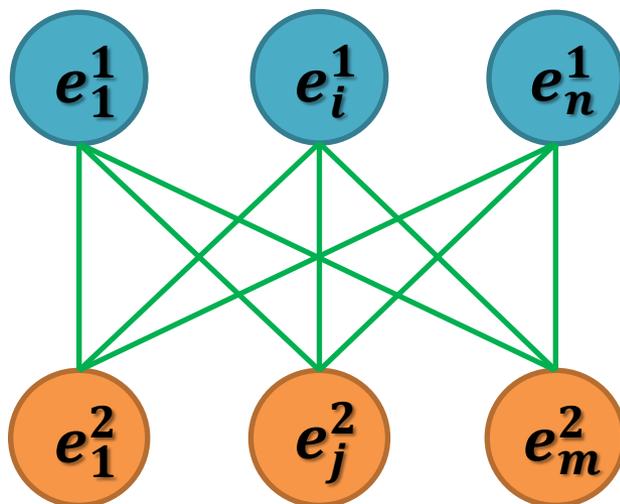
Разработка алгоритма нахождения максимального общего подграфа двух графов

Шаг 4

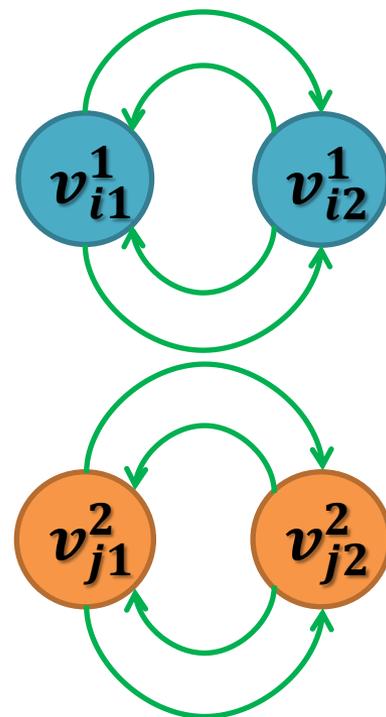


Для каждого паросочетания, полученного на шаге 3 применяем шаг 5

Шаг 5

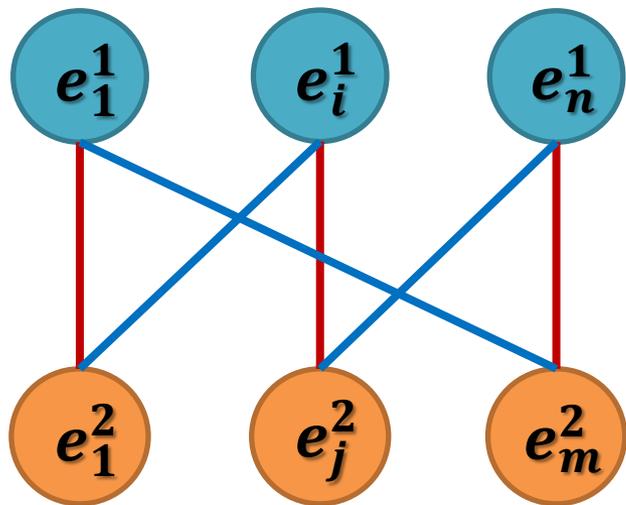


Построение
двудольного графа



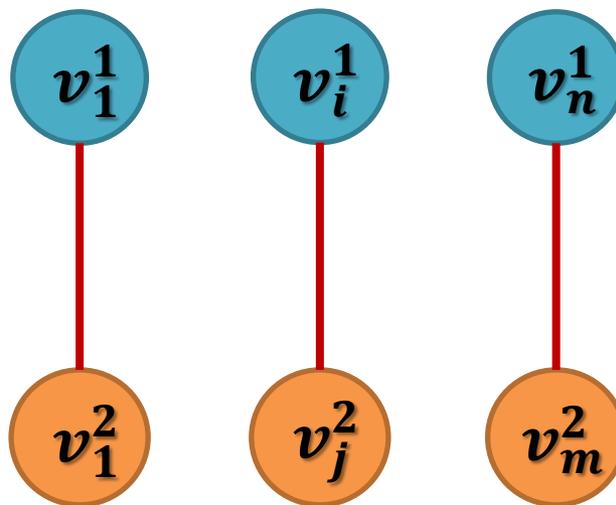
Разработка алгоритма нахождения максимального общего подграфа двух графов

Шаг 6



Поиск максимального паросочетания
максимального веса

Шаг 7



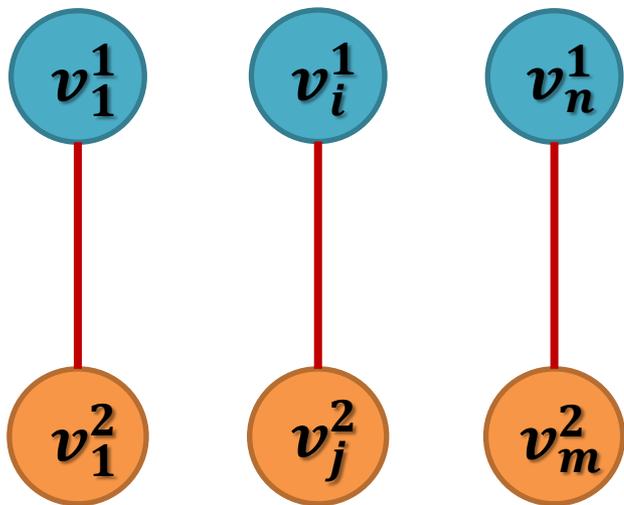
Получение веса паросочетания
(полученного на шаге 3) путем
суммирования максимальных весов,
полученных на шаге 6



M_i

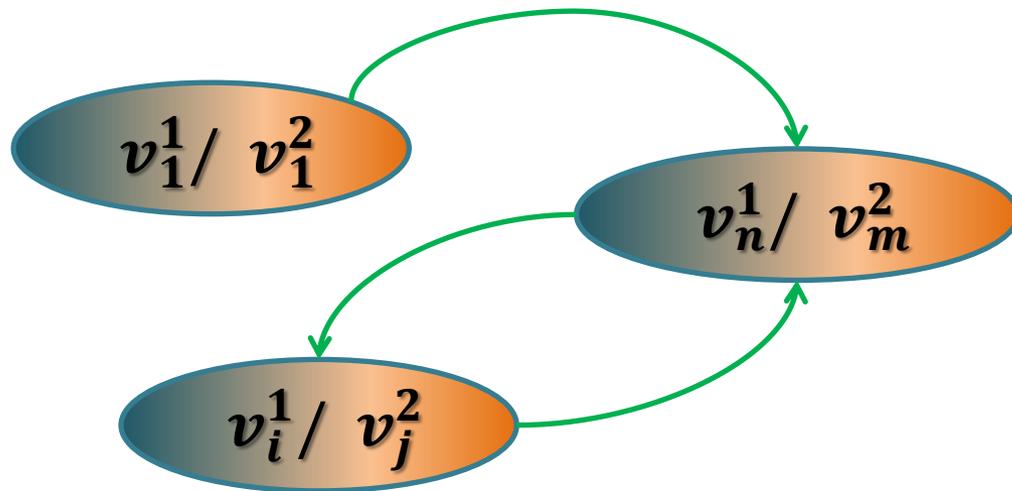
Разработка алгоритма нахождения максимального общего подграфа двух графов

Шаг 8



Выбираем
паросочетание с
максимальным весом
полученным на шаге 7

Шаг 9



Построение максимального общего
графа

Задача нахождения максимального общего подграфа двух графов



Задача нахождения максимального общего подграфа двух графов

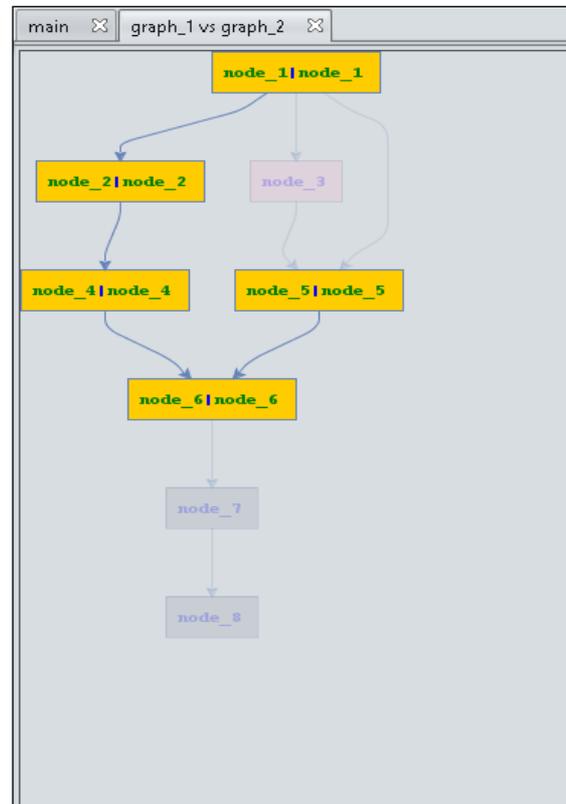
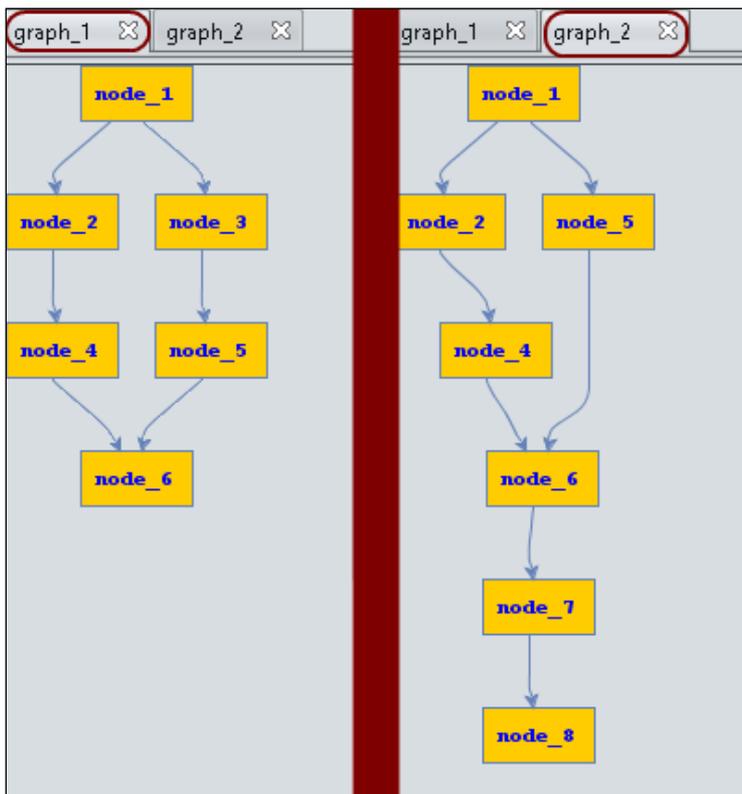


Задача нахождения максимального общего подграфа двух графов

Используется расстояние Хэмминга для определения
веса ребра в двудольном графе

$$d(v, u) = \sum a_i * \mu(p_{i1}, p_{i2}), \text{ где } \sum a_i = 1$$

Результаты и проблемы алгоритма



Алгоритм хорошо справляется с графами, элементы, которых содержат достаточно разнородные значения одних и тех же атрибутов и плохо применим в противном случае

Спасибо за внимание



Страница проекта:

<http://code.google.com/p/visualgraph/>