

Graph Pattern Matching Challenge Report

2019-11730 신현지

2019-12917 이규진

목차

1. How our program chooses Matching order.
2. How our program performs backtracking.
3. Code detail.
4. Environment we run our program.
5. How to compile and execute our program.

1.

저희는 Adaptive Matching Order 중 Candidate-size order 방식을 택했습니다. 이 방식을 택한 이유는 우선, 최대한 많은 Matching order를 출력해야하는 이번 과제의 형식을 고려했습니다. Candidate-size order는 항상 $|Cm(u)|$ 가 최소인 u 를 선택하기 때문에 마지막 candidate를 고르기 직전까지 시간이 비교적 빠를 것이라 생각했습니다. 두 번째로, 마지막 candidate의 후보군의 숫자가 가장 많을 것이며 이 방식이라면 몰려 있는 Matching order를 빠르게 출력할 수 있을 것이라 생각했습니다.

2.

Backtracking을 시작하기 전, 다음과 같은 자료구조를 선언하고 초기화합니다.

- 1) parents : $P(u) = \{u_{p,1}, u_{p,2}, \dots\}$, u 의 부모 vertex의 vector를 반환.
- 2) childs : $C(u) = \{u_{c,1}, u_{c,2}, \dots\}$, u 의 자식 vertex의 vector를 반환.
- 3) Cm : $Cm(u) = \{v_1, v_2, \dots\}$, u 의 candidate set 중 가능한 vertex들의 집합을 반환.
- 4) Cm_queue : $|Cm(u)|$ 에 따라 Min-Heap 형태로 저장.
- 5) M : $M(u) = v$, u 에 대응되는 vertex를 저장 및 반환.
- 6) M_search: M에 이미 할당된 v 를 찾을 때, 빠르게 찾기 위한 자료구조.

Backtracking($v, u, M, M_search \dots$)을 요약하자면 다음과 같습니다.

- $M(u) := v$
- $M_search.insert(v)$
- calculate Cm and Cm_queue
- if(Cm_queue has element /* 다음 실행할 u 가 존재 */) {
- Cm_queue에서 다음 실행할 u_next 를 pop한다.
- for each candidate in $Cm(u_next)$ {
- candidate가 M에 존재하면 pass한다. /* M_search를 이용 */
- Backtracking(candidate, $u_next, M, M_search, Cm_queue, data, cs$);
- }
- }

3.

위의 자료구조는 각각 다음의 형식으로 생성되었습니다.

- 1) parents : vector<vector<Vertex>>
- 2) childs : vector<vector<Vertex>>
- 3) Cm : vector<vector<Vertex>>
- 4) Cm_queue : priority_queue<pair<size_t, Vertex>,
vector<pair<size_t, Vertex>>,
Cm_pair_compare>

이때, Cm_pair_compare은 다음과 같다.

```
struct Cm_pair_compare{  
    bool operator()(pair<size_t, Vertex> a, pair<size_t, Vertex> b){  
        return a.first > b.first;  
    }  
};
```

- 5) M : vector<Vertex>

- 6) M_search: vector<Vertex>

이외의 함수들은 graph.h / candidate_set.h 에서 제공하는 함수를 응용하였습니다.

4.

Linux : Linux version 4.4.0-17763-Microsoft (Microsoft@Microsoft.com) (gcc version 5.4.0 (GCC)) #1432-Microsoft Mon Aug 18 18:18:00 PST 2020

cmake : version 3.16.3

gcc : gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0

5.

기존 skeleton code의 실행방식과 동일합니다.

- mkdir build
- cd build
- cmake ..
- make
- ./main/program <data graph file> <query graph file> <candidate set file>