

Algorithm HW2: Graph Pattern Matching Challenge

2015-16506 이호배

2016-17357 김지완

1. How our program chooses a matching order and performs backtracking

Algorithm

PrintAllMatches(*data, query, candidate set*):

```
Build CandidateSetQue;  
Embedding  $\leftarrow \phi$ ;  
mark  $\leftarrow$  All False;  
BacktrackMatches(data, query, CandidateSetQue, Embedding, mark);
```

BacktrackMatches(*data, query, CandidateSetQue, Embedding, mark*):

```
if  $|Embedding| = |V(query)|$   
    printEmbedding(Embedding);  
else  
    u, CandidateSpace(u)  
         $\leftarrow$  GetExtendableVertex(data, query, CandidateSetQue, Embedding, mark);  
    for v in CandidateSpace(v)  
        Embedding(u)  $\leftarrow v$ ; mark[v]  $\leftarrow$  True;  
        BacktrackMatches(data, query, CandidateSetQue, Embedding, mark);  
        mark[v]  $\leftarrow$  False;
```

GetExtendableVertex(*data, query, CandidateSetQue, Embedding, mark*):

```
u  $\leftarrow$  CandidateSetQue.pop;  
Fix u's CandidateSpace;  
if CandidateSpace is empty  
    return UNEXTENDABLE;  
else  
    Fix query's all vertice' CandidateSpace;  
    if available: return u, CandidateSpace;  
    else: return UNEXTENDABLE;
```

Pattern Matching을 위한 자료구조 *CandidateSetQue*를 새로 만들어 진행한다. *CandidateSetQue*는 $\langle \langle \text{Vertex, Weight} \rangle, \langle \text{Size_t, CandidateSpace} \rangle \rangle$ 로 이루어진 2중 Pair 구조이며 *Size_t*는 *Candidate*의 총 숫자, *Weight*에는 Vertex의 Label 정보와 *query*에서 Vertex Connection 관련 정보를 가중치로 계산하여 집어넣는다.

mapping할 Vertex u 선정 시, *CandidateSpace*의 크기를 기준으로 가장 작은 Vertex로 선택한다. 만약 *CandidateSpace*의 크기가 같을 경우, 계산해 놓은 Weight를 비교해 선택한다. 선택한 u 의 *CandidateSpace*를 *Embedding*과 *mark*정보를 바탕으로 다시 가공해, 분기 수를 최소화 한다. 만약 선택된 u 와 그에 따른 mapping값 v 로 인해 *query*의 어떤 Vertex u' 의 *CandidateSpace*가 empty가 된다면, 해당 mapping은 잘못된 mapping으로 간주하고 분기를 더 진행하지 않고 빠져 나간다.

추가적으로 *CandidateSpace* 도 제공되는 그대로가 아닌 Weight 을 사용하여 mapping시킬 *Candidate* 선택하는 순서도 바뀌보았으나 획기적으로 성능이 향상되지 않아 최종 알고리즘에 반영하지 않았다.

2. How to run our program

Machine Spec

OS: CentOS

CPU: Intel® Core™ i7-7500 CPU @ 2.70GHz

Mem: 4.00 GB

Language & Compiler

C++ / g++ 6.3.0, 9.2.0.

Windows 로컬 환경에서 문제없이 build되는 것을 확인 후, Linux 서버에서도 build 하여 run하는 것까지 수행했다. 컴파일 방법은 과제의 README.md 파일에서 명시한 방법과 같다.

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