2018-12430 Kim Miseung

2018-10036 Park Yoonsoo

2021년 6월 8일

Graph Pattern Matching Challenge

1. Matching order

1) Choosing an Extendable vertex : Candidate-size order

Partial Embedding **M**에서 extendable vertices 중 가장 먼저 **M**에 추가할 vertex **u**를 고를 때에는 priority queue를 사용하여 해당 extendable vertex의 candidate size가 제일 작은 순으로 방문할 수 있도록 하였다.

2) Choosing an Extendable candidate : Candidate which has less degree in data graph

1)을 통해 Partial Embedding **M**에 새롭게 추가할 vertex **u**를 고른 후에, **u**의 extendable candidate들 중에서는 각 extendable candidate들의 data graph에서의 degree가 적은 순으로 수행하였다.

Extendable vertex **u**의 extendable candidate **v**가 Date graph에서의 degree가 적을수록, 나중에 **u**의 child vertex들이 Partial embedding에 extendable vertex가 될 때에 child vertex의 candidate들이 extendable candidate **v**가 Candidate Set에서 **v**와 연결될 확률이 낮아져 제외되는 extendable candidate이 많아져서 backtracking할 때에 고려해야하는 경우의 수가 줄어들어 시간을 줄일 수 있다.

2. Backtracking

Backtracking에 필요한 visited 와 partial Embedding **M** 을 각각 배열과 Vertex pair의 벡터로 구현했다. **M**은 기본적으로는 stack처럼 작동하나 출력을 위해 vector로 구현했다.

Backtracking 용도로 재귀 호출을 하기 위해 FindPartialEmbedding 함수를 구현했다. 함수 내부는 다음과 같이 세 파트로 구성되어 있다.

1) Stdout

먼저, Embedding을 찾은 경우 바로 출력하기 위해 내부에 출력문을 구현했다. 이 때 Embedding을 찾았다는 기준은 Partial Embedding **M**의 크기가 query 의 크기와 동일하다는 것이다. stdout으로 출력한다. 만약 Partial Embedding이 matching되지 않는 경우 재귀함수는 이 출력문에 이르지 못하고 탈출하게 되어 출력되지 않는다.

2) Initial condition

M의 크기가 0인 경우, 아무 vertex도 들어오지 않은 초기 상태로 이때 query graph의 DAG에서 구했던 root를 처음에 넣어준다. root의 ExtendableCandidate들을 앞선 순서로 차례로 방문한다. 이때, 해당 candidate을 방문했다고 표시하기 위해 _visited_에서 해당 vertex를 Index로 하는 부분을 1로 바꾸어준다. 그리고 재귀적으로 이 vertex를 포함한 Partial Embedding **M**을 매개변수로 하는 _FindPartialEmbedding_을 호출한다. 여기서 재귀적으로 쌓인 FindPartialEmbedding 함수들이 끝나고 다시 해당 코드의 다음 줄로 오게 되면 _visited_의 해당 index를 다시 0으로 바꾸고 다음 extendable candidate에 대해 같은 과정을 반복적으로 수행한다.

3) Recursive call

2)와 유사한 과정을 이제 중간 단계에서 재귀적으로 수행하는 부분이다. 이 때도 마찬가지로 현재 vertex에서 가능한 extendable candidates를 기존에 정한 순서로 방문하고 복사한 **M**에 새로 방문한 vertex를 삽입한 **newM**을 매개변수로 다시 재귀함수를 호출한다. 마찬가지로 모든 재귀함수들이 끝나고 원래 지점으로 돌아오게 되면 visited/의 해당 index를 다시 0으로 바꾸고 반복 수행한다.

3.Enviroment

1) 머신 스펙

- OS: macOS Big Sur
- CPU: 2.4 GHz 쿼드 코어 Intel Core i5
- Mem: 256GB

2) 컴파일러/인터프리터

Apple clang version 12.0.0 (clang-1200.0.32.21)

3) 실행방식

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

주어진 프로그램 방식과 동일하게 실행할 수 있다.