

Graph Pattern Matching Report

2017-19835 김성훈

2017-14044 김재웅

1. 개발 환경

OS: Ubuntu 20.04.1 LTS, 5.4.72-microsoft-standard-WSL2

2. 실행 방법

cmake / make 후 run.sh 실행 - 각 쿼리마다 1분 시간 제한동안 실행된 후, 출력 파일을 output 폴더에 생성하고 줄 수를 출력

3. 알고리즘

- DAG 변환

주어진 query graph를 DAG로 변환한다. 먼저 query의 vertex 중 (해당하는 candidate set의 크기) / (vertex의 degree)가 가장 작은 vertex를 root로 정한다. 이후 DAG에 vertex를 하나씩 추가하는데, 이미 추가된 vertex에 인접한 vertex들 중에서 (해당하는 candidate set의 크기) / (vertex의 degree - (이미 추가된 neighbor vertex의 수))가 최소인 vertex부터 추가한다. query graph의 인접한 vertex에 대해 먼저 추가된 vertex가 나중에 추가된 vertex의 parent가 되도록 한다. 기존 Graph 클래스에 BuildDAG 함수 및 관련 멤버 변수를 추가하는 방법으로 구현했다.

- Backtracking

변환한 DAG를 이용해 Backtracking한다. query 그래프의 vertex를 u , data graph/candidate set의 vertex를 v 로 이름 붙였다. 또, i 번째의 u 를 매칭시키고 $(i+1)$ 번째의 매칭을 탐색하는 시점을 “레벨 i ”라고 부른다.

한 u 를 매칭할 때, u 에 매칭될 수 있는 v 들 중 하나를 고른다. 이후, u 의 children 중 다음 u 로 선정될 수 있는 vertex들을 binary set ‘extendNext’에 저장한다. u 에 매칭될 수 있는 v 의 수는 candidate set에서의 v 의 개수에서 이미 매칭된 parent v 와 매칭될 수 없는 경우, 혹은 v 자체가 이미 매칭된 vertex인 경우를 제외한 수이다.

extendNext set은 ‘ u 에 매칭될 수 있는 v 의 수’를 첫번째 키로 사용하기 때문에 매칭될 수 있는 v 의 수가 가장 작은 u 를 $O(\log n)$ 으로 고를 수 있다. extendNext set이 비어 있지 않다면, 그 중 가장 앞에 있는 노드를 다음 level에 매칭할 u 로 고른다.

중간에 더 이상 다음 level로 extend할 수 없는 경우, 현재 레벨에서 매칭될 수 있는 다음 v 를 고른다. 현재 레벨에서 매칭할 수 있는 v 가 더 없는 경우, 이전 레벨로 돌아가 이전 레벨에서 매칭될 수 있는 다음 v 를 고른다. 모든 레벨에서 v 가 매칭될 경우 결과를 출력하고, 다음 매칭을 이어서 찾는다.

메모리를 최적으로 사용하기 위해, 각 레벨에서 extendNext의 상태 변화를 stack 방식으로 저장하여 레벨이 내려갈 때 기존 상태를 복원할 수 있도록 했다.