

Graph Pattern Matching Challenge Report

컴퓨터공학부 2019-19519 최서린

컴퓨터공학부 2019-19330 허수현

1. Matching Order를 정하는 방법

먼저, root를 candidate size가 가장 작은 vertex를 고른다. 그 후, root를 기준으로 bfs를 돌려 DAG를 만들어 extendable vertices를 찾았다. 나온 vertices중에서 Candidate Size가 작은 vertex를 골라 그 vertex의 Candidate Set을 input으로 들어온 순서대로 골라 다음 extendable vertex를 찾아갔다. 모든 query_vertex가 매칭이 된 경우 답을 출력하도록 했다.

위 matching_order로 답이 안 나오는 경우가 있는 경우에는 matching order를 다시 구했다.

먼저, query의 전체 vertex 중 minimum degree를 가진 vertex를 initial vertex로 지정해준 후, reachableVertex 벡터를 정의하여 다음 vertex를 고르는 작업을 반복해주었다. 이때, reachableVertex에는 현재 선택된 vertex들로부터 edge가 하나라도 있는 unchosen vertex들이 저장되어 있도록 하였다.

다음 vertex를 고를 때에는, 네 가지의 케이스에 대하여 다르게 처리하였다.

Case 0) 먼저, reachable vertex가 없는 경우에는, unchosen vertex 중에서 minimum degree를 가진 vertex를 고르고 다시 위 과정을 반복해주었다.

Case 1) unchosen vertex 중 Candidate Size가 1인 vertex가 있다면, 그 vertex들 중 vertex_id가 가장 작은 vertex를 골라주었다.

Case 2, 3) 이외의 경우에는 chosen vertex들 중 neighbor를 가장 많이 갖는 unchosen vertex를 선정해주었다. 이때, 이러한 vertex가 하나라면 바로 선택해주고, 여러 개라면 그 중 minimum Candidate Size를 가진 vertex로 선택해주었다.

이렇게 완성한 matching order를 토대로 backtracking을 진행하여 answer를 출력하도록 하였다.

2. Environment

프로그램 실행 환경 및 구현 언어: Ubuntu, C++

3. How to run program

mkdir build

cd build

cmake ..

make

./main/program <data graph file> <query graph file> <candidate set file>

4. 각 프로그램별 걸린 시간 측정

input	시간	output 수	input	시간	output 수	input	시간	output 수
hprd_n1	0	96	human_n1	3	100000	yeast_n1	11	100000
hprd_n3	25	100000	human_n3	-	-	yeast_n3	21	100000
hprd_n5	12	32832	human_n5	7	100000	yeast_n5	34	100000
hprd_n8	43	100000	human_n8	11	100000	yeast_n8	53	100000
hprd_s1	0	504	human_s1	3	100000	yeast_s1	10	100000
hprd_s3	35	100000	human_s3	6	100000	yeast_s3	537	100000
hprd_s5	35	100000	human_s5	7	100000	yeast_s5	-	-
hprd_s8	44	100000	human_s8	-	-	yeast_s8	-	