

알고리즘

Graph pattern matching challenge 보고서

홍영서, 윤성원

백트래킹 과정 설명

```
void Backtrack::SubgraphSearch(const Graph &data, const Graph &query,
                               const CandidateSet &cs, std::vector<std::pair<Vertex, Vertex>> match){
    if(match.size()==query.GetNumVertices()){
        PrintAllMatches(query, match); 1
    } else {
        Vertex nextQueryVertex = NextQueryVertex(query, cs); 2

        std::vector<Vertex> candidates = GetCandidates(cs, nextQueryVertex);

        for(Vertex v : candidates){
            std::pair<Vertex, Vertex> pair = {nextQueryVertex, v};
            if(IsExtendable(pair, match, data, query, cs)==true){ 3
                match.push_back(pair);

                UpdateState(query, nextQueryVertex); 4
                SubgraphSearch(data, query, cs, match);
                match.pop_back();

                RestoreState(query, nextQueryVertex);
            }
        }
    }
}
```

1. 매칭이 완료된 (u,v)를 원소로하는 vector(match)를 받아 함수가 시작될 때 vector의 원소 개수와 query의 vertex 개수를 비교하여 같으면 전체 매칭이 끝났다는 의미이므로 출력한다. 같지 않으면 남은 매칭을 진행한다.

2. Matching order - NextQueryVertex 함수를 이용해 query graph에서 다음에 매칭될 vertex를 찾는다.

이 때 함수 작동 방식은 다음과 같다.

- 1) 아직 매칭되지 않은 vertex중에 parent들은 모두 매칭된 vertex들을 찾는다.
- 2) 그 중 candidate 수가 가장 적은 vertex를 택한다.

3. 매칭될 vertex의 각 candidate들에 대해 IsExtendable 함수를 이용해 매칭 가능한지 확인한다.

```

bool Backtrack::IsExtendable(std::pair<Vertex, Vertex> pair, std::vector<std::pair<Vertex, Vertex>> match,
                             const Graph &data, const Graph &query, const CandidateSet &cs){
    for(auto it=match.begin(); it!=match.end(); it++){
        std::pair<Vertex, Vertex> p = *it;
        if(pair.second == p.second) return false; // 중복 여부 확인 1
    }
    std::vector<Vertex> parentID = query.GetParentID(pair.first);
    for(size_t i=0; i<parentID.size(); i++){
        for(size_t j=0; j<match.size(); j++){
            if(match[j].first==parentID[i]) {
                if(!data.IsNeighbor(match[j].second, pair.second)) return false;
                break;
            }
        }
    }
    return true;
}

```

이 때 함수 작동 방식은 다음과 같다.

- 1) 이미 매칭되었으면 false를 리턴한다. (match vector에서 확인한다.)
- 2) 매칭된 parent들이 그 candidate를 가리키고있는지 확인한다.
 - 이 때 각 parent들을 global variable로 선언한 map에 저장하고있어 그것을 이용한다.

4. 매칭이 가능한 candidate들은 match vector에 넣고 recursive하게 백트래킹을 진행한다.

실행 환경과 실행 방법 설명

1. 실행 환경 - window, c++
2. 실행 방법 - cmake로 컴파일 후
program.exe [data graph] [query graph] [candidate] 로 실행한다.