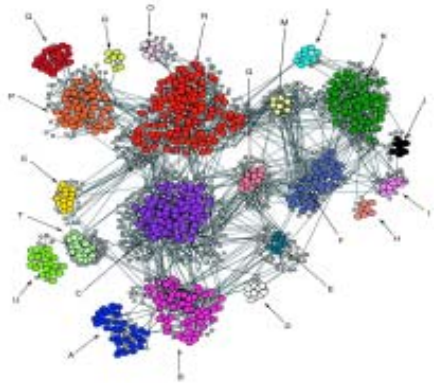
Subgraph Isomorphic Algorithms

제8회 소프웨어무결점연구센터 워크숍

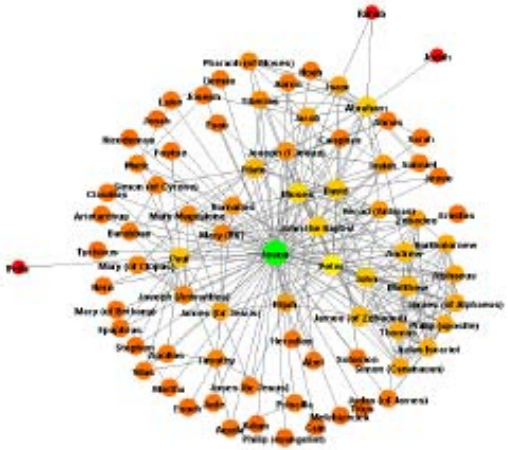
경북대학교 데이터베이스 연구실
이진수

Graphs are Everywhere

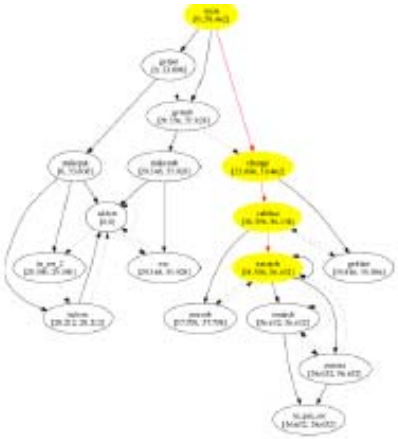


Magwene *et al. Genome Biology* 2004 5:R100

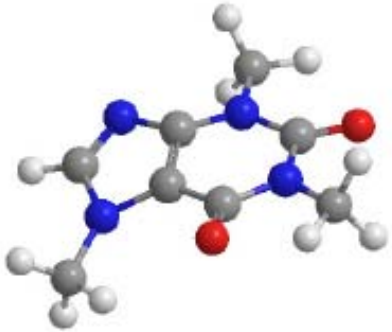
Co-expression Network



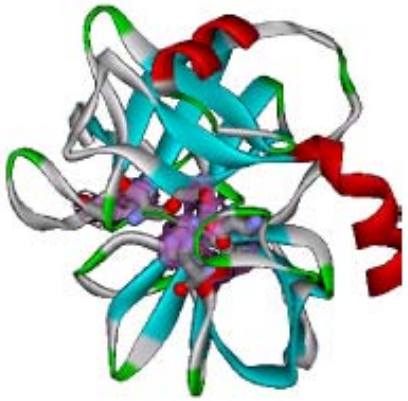
Social Network



Program Flow



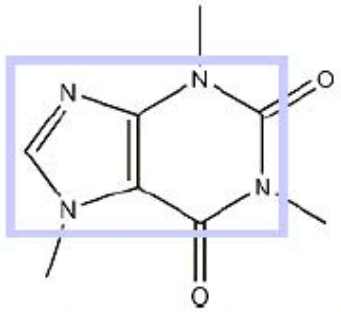
Chemical Compound



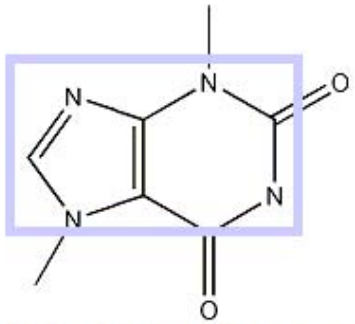
Protein Structure

Example 1: Frequent Subgraphs

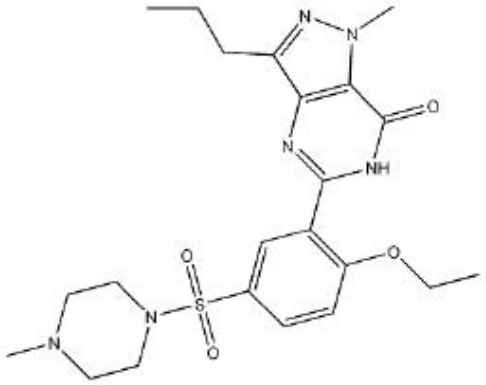
CHEMICAL COMPOUNDS



(a) caffeine



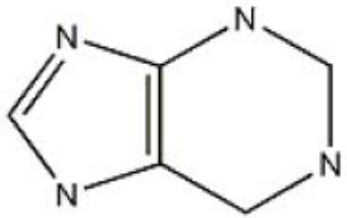
(b) diurobromine



(c) viagra

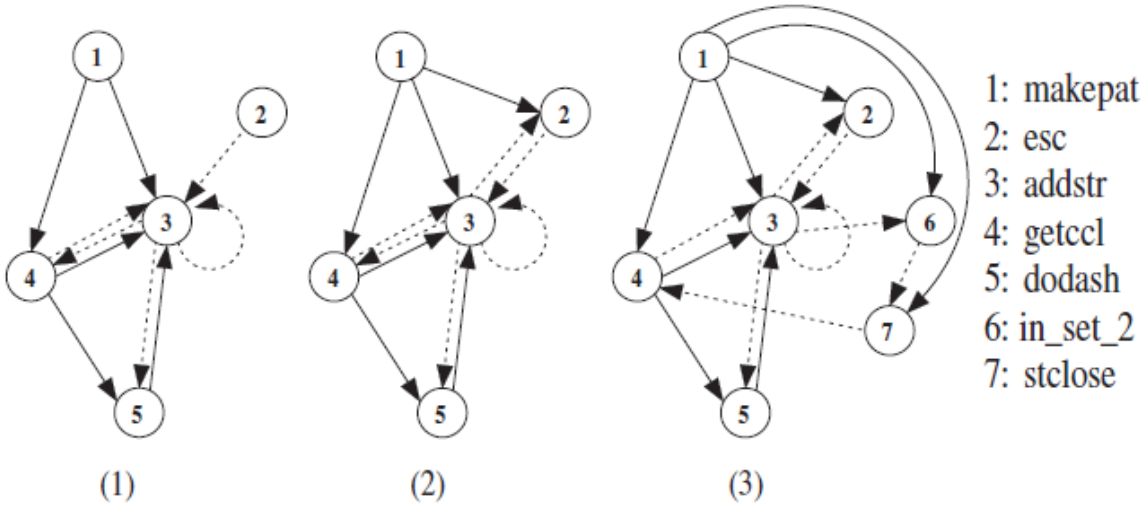
...

FREQUENT SUBGRAPH



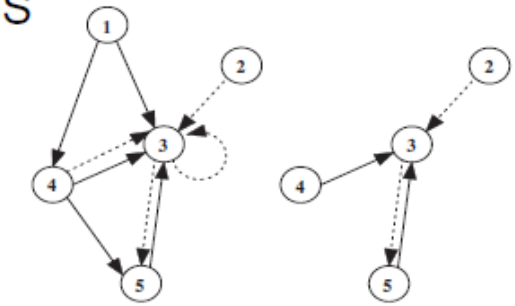
Example 2: Frequent Subgraphs

PROGRAM CALL GRAPHS



(1) (2) (3)

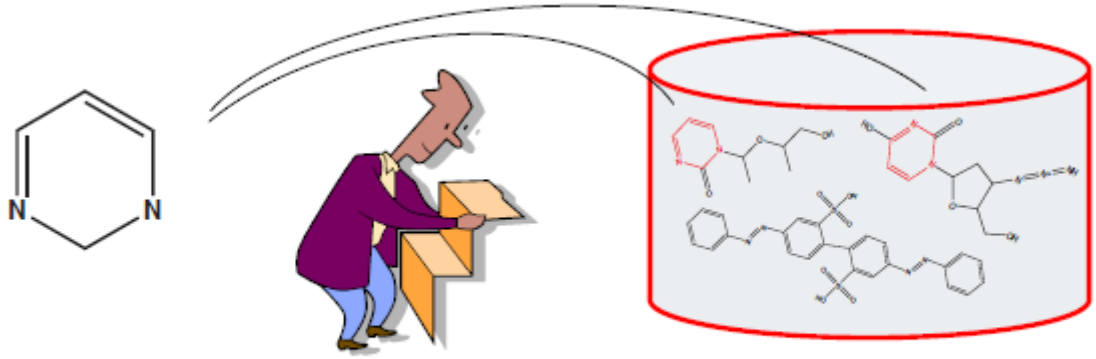
FREQUENT SUBGRAPHS
(MIN SUPPORT IS 2)



(1) (2)

Graph Searching

- Find all of the graphs in a database that contain the query graph

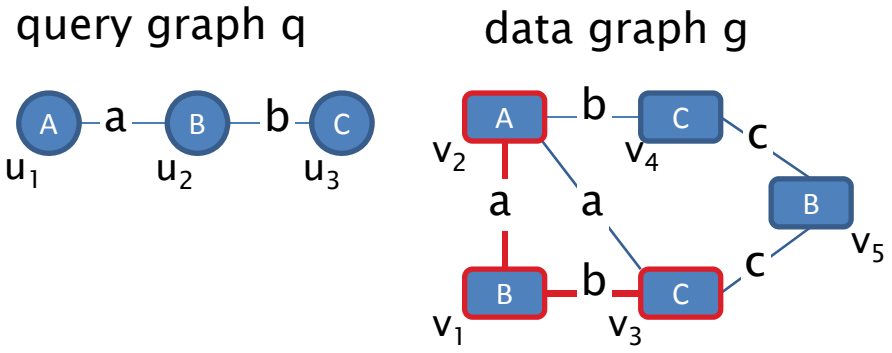


query graph

graph database

Problem Definition: Subgraph Isomorphism

- Graph g = {V, E, ∑, L}
  - V(g): a set of vertices
  - E(g): a set of edges, E(g) ⊆ V(g)×V(g)
  - ∑: a set of all labels
  - L: labeling function, L(v) ∈ ∑ and L(e) ∈ ∑ (v ∈ V(g), e ∈ E(g))
- Query graph q is **subgraph-isomorphic** to data graph g, if there is an **injective function** M : V(q) → V(g), such that :
  1. ∀u∈V(q) : L(u) ⊆ L(M(u))
  2. ∀(u,u')∈E(q) : (M(u),M(u'))∈E(g)
  3. ∀(u,u')∈E(q) : L(u,u') = L(M(u),M(u'))



Existing Subgraph Isomorphic Algorithms

- Popular algorithms:
  - Ullman's algorithm, ACM Journal, 1976
  - VF2 algorithm, Trans. on Patt. Analysis and Machine Intel., 2004

- Recently published algorithms:
  - GraphQL algorithm, SIGMOD 2008
  - QuickSI algorithm, PVLDB 2008
  - GADDI algorithm, EDBT 2009
  - SPath algorithm, VLDB 2010

General Search Algorithm

- 기존 검색 알고리즘을 일반화하여 표현
- 주요 서브 루틴
  - FilterCandidates
    - 각 query vertex와 matching이 될 수 있는 candidate data vertex set을 검색
  - NextQueryVertex
    - Matching을 찾을 다음 query vertex를 선택
  - RefineCandidates
    - 검색 상태 정보를 사용하여 candidate set을 refine
  - IsJoinable
    - Matching된 data vertex가 유요한 matching인지를 검사

Algorithm 1 GENERICQUERYPROC

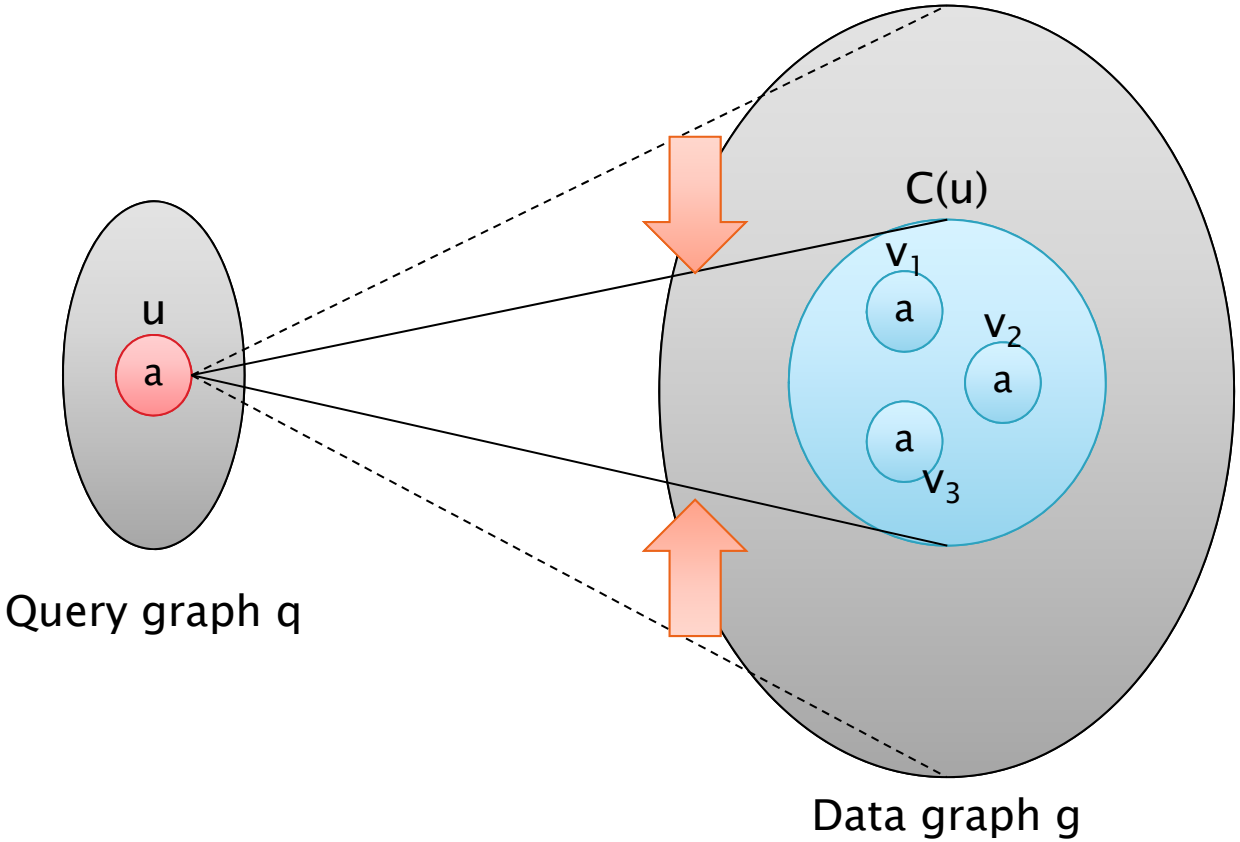
Input: query graph q
Input: data graph g
Output: all subgraph isomorphisms of q in g

```
1: M := ∅;
2: for each u ∈ V(q) do
3:    C(u) := FILTERCANDIDATES (q, g, u, ...);
      [[ ∀v ∈ C(u)((v ∈ V(g)) ∧ (L(u) ⊆ L(v))) ]]
4:    if C(u) = ∅ then
5:       return;
6: SUBGRAPHSEARCH (q, g, M, ...);
```

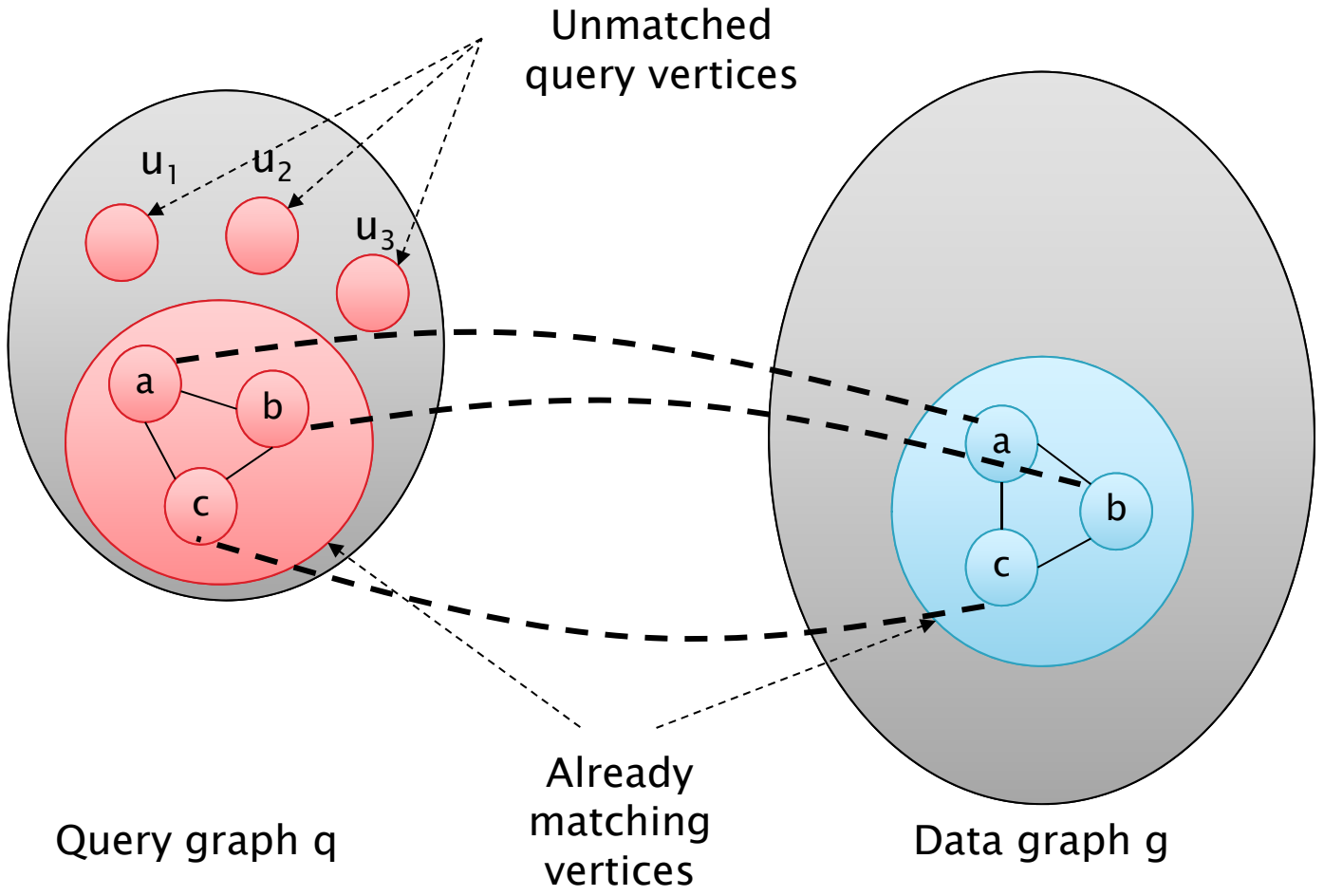
Subroutine SUBGRAPHSEARCH $(q, g, M, \ldots)$

```
1: if |M| = |V(q)| then
2:    report M;
3: else
4:    u := NEXTQUERYVERTEX (...);
      [[ u ∈ V(q) ∧ ∀(u', v) ∈ M(u' ≠ u) ]]
5:    C_R := REFINECANDIDATES (M, u, C(u), ...);
      [[ C_R ⊆ C(u) ]]
6:    for each v ∈ C_R such that v is not yet matched do
7:       if ISJOINABLE (q, g, M, u, v, ...) then
8:          [[ ∀(u', v') ∈ M((u, u') ∈ E(q) ⟹
                 (v, v') ∈ E(g) ∧ L(u, u') = L(v, v')) ]]
9:          UPDATESTATE (M, u, v, ...);
            [[ (u, v) ∈ M ]]
10:         SUBGRAPHSEARCH (q, g, M, ...);
11:         RESTORESTATE (M, u, v, ...);
            [[ (u, v) ∉ M ]]
```

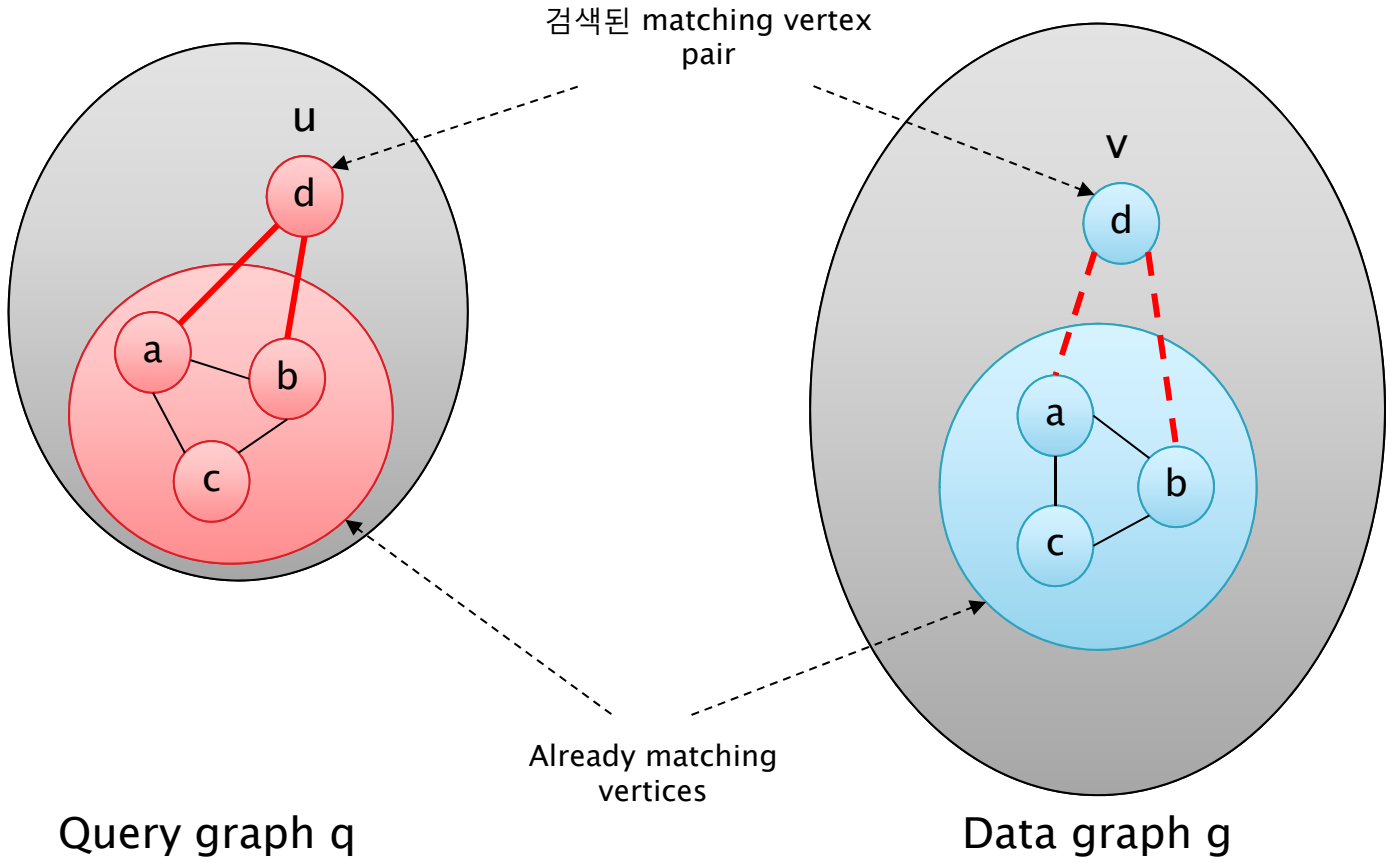
Filtering Candidates



Selecting Next Query Vertex



Joinable Test



Ullman's Algorithm

- NextQueryVertex
  - 매칭 되지 않은 query vertex들 중에서 random하게 선택
- RefineCandidate
  - Vertex의 degree정보를 활용
  - Deg(u) ≤ Deg(v) (u ∈ V(q), v ∈ C(u))를 만족하지 않는 data graph vertex v를 u의 candidate set C(u)에서 제거

GraphQL Algorithm

- FilterCandidates
  - Vertex의 neighborhood정보를 활용하여 filtering 효과를 높임
  - 모든 vertex의 neighborhood vertex들에 나타난 label의 빈도수를 활용
- NextQueryVertex
  - |C(u)|의 크기가 작은 순으로 query graph vertex를 선택함
  - RDBMS의 heuristic join order selection과 유사한 방법으로 search space의 크기를 줄이는 효과를 가짐

Thank You