# Problem Statement

CS6740 : Searching & Indexing in Large Datasets

## 1 Statement

Let $G(V, E)$ be a labelled graph, such that $V$ is the set of vertices of the graph , $E$ is the set of all edges $e \in V \times V$ and $l$ be a function mapping which maps every vertex and edge to a label.

For two labelled graphs $g_1$ and $g_2$, subgraph isomorphism is an injective function $f : V(g_1) \to V(g_2)$ such that

- $\forall v \in V(g_1),\ \ l_1(v) = l_2(f(v))$
- $\forall (u, v) \in E(g_1), (f(u), f(v)) \in E(g_2)$ and $l_1(u, v) = l_2(f(u), f(v))$

where $l_1$ and $l_2$ are labeling functions of graph $g_1$ and $g_2$ respectively.

In brief, if there exists a mapping such that every vertex of a graph $g_1$ is mapped to a vertex from a graph $g_2$ , and for every edge $e_1(u_1, v_1)$ of graph $g_1$ there exists at least an edge $e_2(l(u_1), l(v_1))$ in $g_2$, it can be said that $g_1$ is contained by $g_2$. If $g_1$ is subgraph of $g_2$, $g_2$ is called supergraph of $g_1$.

Given a graph database D consisting labelled graphs $g_1, g_2, \ldots, g_n$, the goal is to find the subset of D consisting the graphs that are contained in the given query graph. This can be also be stated as a supergraph search problem because we want to find which of the database graphs have query graph as their supergraph.

This can be applied to many real-life scenarios like

- In organic compounds, different groups have unique characteristics of their own. e.g. groups like $-OH$ , $-CHO$ and many other groups ( some of them are much more complex and may contain cycles ) have their unique characteristics. So, given an unknown compound ( query graph in our case), if we can find which of these groups ( a whole database of graphs ) are present in it, we may be able to deduce some of the characteristics of the given compound.
- In bio-informatics, some molecular structures may suggest the possibility of some particular diseases, so given a test sample, you may want to

find which of those structures are present in the patient, that can help predicting the disease.

- In the web, malicious activities can be converted into a database of malicious patterns ( graphs ), and given a graph of few users, you may want to check whether there exist such patterns in the graph to flag those users and reduce such activities.

## 2 Implementation

The naive implementation would be to pick up every graph from the database and check whether it is subgraph of the query graph.

**Require:** graph database D, query graph q
  $count = 0$
  **while** there is a graph g in database D **do**
    **if** $g \subseteq q$ **then**
      $count = count + 1$
    **end if**
  **end while**
  **return** count

Subgraph isomorphism itself being a non-polynomial in nature, this naive solution does not scale well. Given a database of graphs and a query graph, the task is to implement a program which does graph containment query as fast as possible. The task will be evaluated based on the correctness and execution time of the program.

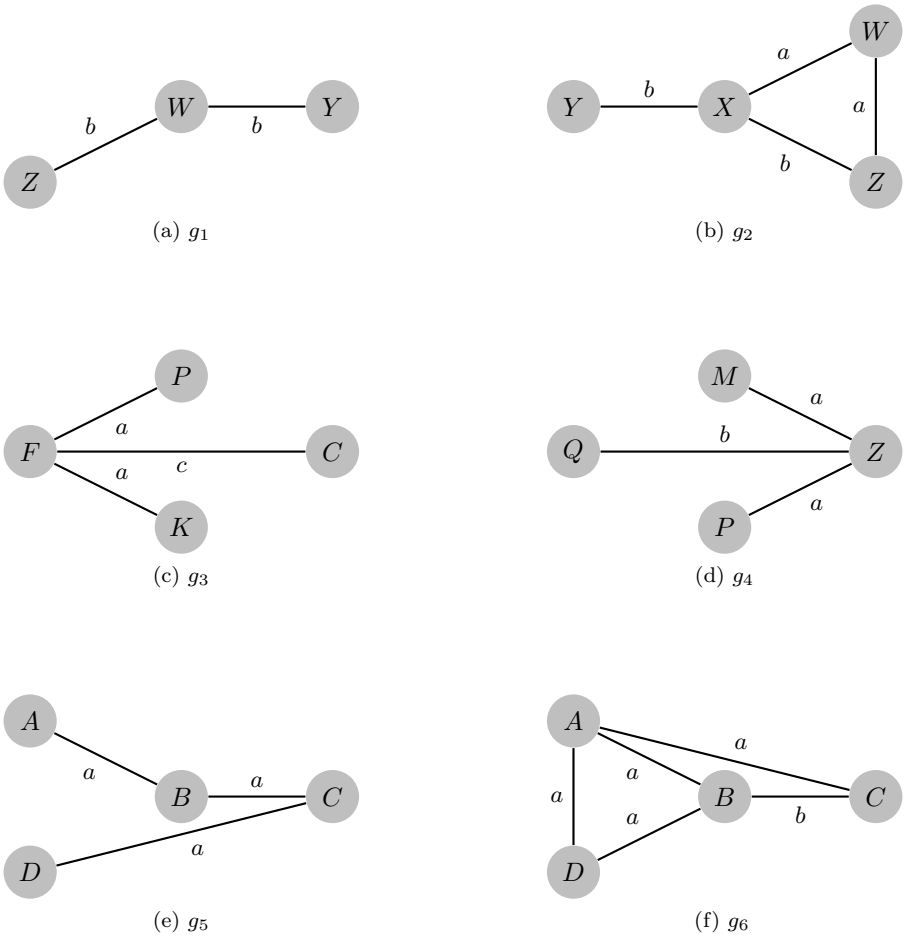
There has been some work done in this area to make it faster. You may want to go through some of these papers before implementation.

**You may use libraries for checking the subgraph isomorphism.**

Note: You may use any programming language (yes, if you have your own language that compiles, you may use it too), but it is a known fact that some languages like Python can be 10-100 times slower than C/C++.

# 3 Example

From the given graphs in dataset (Figure 1), only $g_4$ is contained by query graph (Figure 2).




(a) $g_1$ (b) $g_2$

(c) $g_3$ (d) $g_4$

(e) $g_5$ (f) $g_6$

Figure 1: Graphs in Database

Note: Multiple nodes in the same graph can also have same labels just the same way as edges do. e.g. in chemical compounds, the atoms which are depicted as nodes can have similar labels ( almost every complex organic compound has more than 1 carbon or hydrogen molecule).

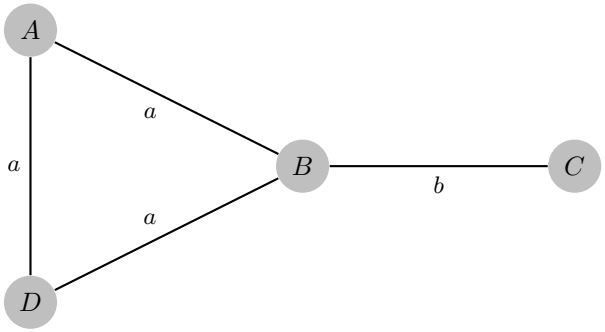



Figure 2: Query graph