Introduction:

- Billions of nodes, effect on performance
- Querying: subgraph matching
- Goal: To implement and analyze subgraph matching algorithms
- VF2 and GraphQL

Motivation:

- Framework – Prof Carlos Rivero
- Ullman’s subgraph matching algorithm, not efficient.
- Two state of art algorithms VF2 and GraphQL
- VF2 – only exact Matching

Datasets:

- The datasets used are Biological datasets.
- Molecules, Contact Maps and Proteins.
- Data parser

Algorithms:

- VF2
  - Input: [querygraph,subgraph]perations
  - Output: [set of all mappings] and [set of feasible mates]
  - Algorithm:
    - Step 1: Call Search Order()
    - Step 2: Call Check()
    - Step 3: Call Feasible Mates()
    - Step 4: Call Search Order()
    - Step 5: Call DFS Search()

Examples / Implementation:

GraphQl:

- Query: [dataGraph, queryGraph]
- Search Order:
  - DFS Search()
  - Check()
  - Step 1: [dataGraph, queryGraph]
  - Step 2: [dataGraph, queryGraph]
  - Step 3: [dataGraph, queryGraph]
  - Step 4: [dataGraph, queryGraph]
  - Step 5: [dataGraph, queryGraph]

Results:

- VF2 vs GraphQL – Nodes
- GraphQl Query size vs Algorithm
- VF2 vs GraphQL – Datasets vs Algorithms

Conclusion:

- Performance depends
- Framework – a better comparison between subgraph matching algorithms
- Focus towards join order, pruning

References:

2. Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento: A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs.