Graphs are very important in many applications which rely on networks and connections.[1] Ability to get relevant matches from the graph is generally time consuming[1]. In case, the node of the graph contains a set of attributes generally node to node calculations are costly[1].

Preprocessing steps[1]

- Formation of tree structure
- Construction of inverted lists
- Horizontal Pruning
- Vertical Pruning
- Structural pruning

Matching steps[1]

- Finding dominating set and convert to graph
- Finding match for dominating graph
- Find match for entire graph from dominating graph match

Experiments

- Time taken(s) vs Size of data graph
- Time taken(ms) vs size of query Graph

Results

The results below are for a large dataset

<table>
<thead>
<tr>
<th>Size of query graph</th>
<th>time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9119 ms</td>
</tr>
<tr>
<td>6</td>
<td>17439 ms</td>
</tr>
</tbody>
</table>

A number of factors affect the time to get the answer
- These include size of dominating set - smaller a dominating set lesser need to do similarity matching
- The number of distinct elements[1], due to size of offline structure being proportional to number of elements.

Conclusion-Future work

The preprocessing as well as finding matches for dominating set makes subgraph matching very efficient[1].

References