

## 第一章 经典图查询算法

本章将详细介绍几种经典的图查询算法。由于基本的图查询模式均为“过滤-验证”，所以要提高查询效率，只能从两方面优化。一是过滤阶段优化，二是验证阶段优化。众所周知，图是结构画的数据，目前并没有一个统一高效的索引机制可以实现较好的查询结果。因此大多数学者都会在过滤阶段选用不同的索引方式来得到更好的查询结果。而在验证阶段，由于图同构是个 NP-hard 问题，所以验证会消耗大量时间。如何快速检测同构，或将同构转为其他非复杂多项式的一半问题也是学者们关心的一个热点课题。本章介绍的几种查询方法均为过滤阶段的优化。

### 1.1 图精确查询算法

本节将详细介绍子图查询中的 *GraphGrep* 算法<sup>[1]</sup> 和 *gIndex* 算法<sup>[2]</sup>。

#### 1.1.1 GraphGrep 算法

2002 年 ShaSha 教授等人提出了 *GraphGrep* 算法。*GraphGrep* 算法是基于特征索引方法中的第一个经典算法，它采取典型的“过滤-验证”框架，*GraphGrep* 算法中只讨论过滤阶段。由于基于结构的算法中对图的顺序扫描代价太高，*GraphGrep* 提出基于路径的过滤方法，来减少候选集大小。

其算法的主要流程是这样的：

1. 构造索引: 首先将节点和边利用哈希存成两个二维表作为未来筛选用特征之一，然后枚举图数据库中所有长度不大于  $l_p$  的路径  $v_0, v_1, v_2, \dots, v_k, (k \leq l_p, \forall i \in [1, k-1], (v_i, v_{i+1}) \in E)$ ，然后将这些路径与图的关系存为一个二维表作为索引。表的每一行代表每一个图，每一列为一个路径，利用简单哈希确定路径对于行号，每一个单元格代表在该图包含该路径几条，如表1.1所示。
2. 解析查询: 如同图数据库，首先将节点和边利用哈希存成两个二维表，然后枚举查询中的所有长度不大于  $l_p$  的路径，哈希存在一个列表中。

Key	$g_1$	$g_2$	$g_3$
h(CA)	1	0	1
...			
h(ABAB)	2	2	0

表 1.1 GraphGrep 索引

### 3. 数据库过滤:

- (a) 利用节点信息过滤: 如果查询图中的某一节点在数据库中一图里未出现, 则此图一定不会包含查询图, 因此可以删去。
- (b) 利用边信息过滤: 同节点过滤, 如果某条边未出现, 则一定不会包含查询图, 可删去。
- (c) 利用路径个数进行过滤: 如果查询图中某一路径个数大于数据库中一图的此路径个数, 则此图一定不会包含查询图, 可以从候选集中删去。

### 4. 子图查询: 利用标号进行路径合成, 从候选集中删去所有未能成功合成的候选图。剩下的就是 GraphGrep 算法返回的候选集。后续再加以图同构验证即可得到最终子图查询结果。

## 参考文献

- [1] Rosalba Guigno and Dennis Shasha. Graphgrep : A fast and universal method for querying graphs. 2002.
- [2] Xifeng Yan, Philip S.Yu, and Jiawei Han. Graph indexing : A frequent structure-based approach. *ACM*, 2004.