# Graph Diff

**文件修改记录**

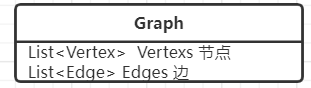
|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **内容** | **修改人** | **修改日期** |
| 1.0 | 创建 | 田开贺 | 2022-03-30 |
| 1.1 | 修改 | 田开贺 | 2022-03-31 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 前提：

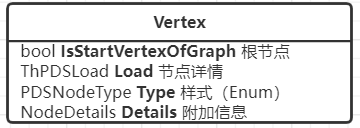
1. 电力系统图Graph形式为QuikGraph。 <https://github.com/KeRNeLith/QuikGraph>
2. 此图是有向图，但不是连通图，可能存在多个连通图组合成的一个Graph
3. 目前的想法是：实现图的Diff算法，参数为两个Graph，返回值为两个图Diff后的结果，也是一个Graph，我们希望返回的Graph可以通过一些附加属性(Tag)，来标识这个”节点”或者”边”的变化，目前的想法是每条边和每个节点都新增一个Tag属性，此Tag只用来标识本身(节点/边)
4. 目前讨论Diff后的结果有五种大类型：（增加，删除，移位，变化和未变化)。变化里面又细分（定性：属性的改变，定量：属性值的变化）
5. Tag内的diff结果是可以叠加的，例如一个节点可以被移位，且属性被改变

补充：

Graph的结构如下如所示，一个图其实就是一堆顶点和一堆边



其中节点的属性有以下这些



其中需要关注的仅有四点： ID(string),描述(string),功率(double)，是否具有消防属性(bool)) (这四个在上图找不全是因为有的属性在Load内部)

这四个属性又进行了分级：

ID：非常重要的属性，是判断两个节点是否相同的凭证，如果两个图比较同一

节点，两个节点的ID相同，可以认为他们是一个节点，如果ID不同，那么要认为旧的节点被**删除**了，新的节点被**创建**了，如果同ID的节点在不同的图下的”位置”不一致，那么认为这个节点被**移位**了。

功率 & 消防属性：比较重要的属性，如果两个图比较同一节点，两个节点的ID相同，功率 & 消防属性不同，那么认为这个节点的属性被改变了, 消防属性属于定性变化，功率值改变属于定量变化。

描述：不那么重要的属性，如果两个图的同一节点，只有描述不同，那么新节点的Node不标注为变化，只是把这个属性进行刷新

其中边的属性有以下这些



其中需要关注的仅有一个(回路ID)

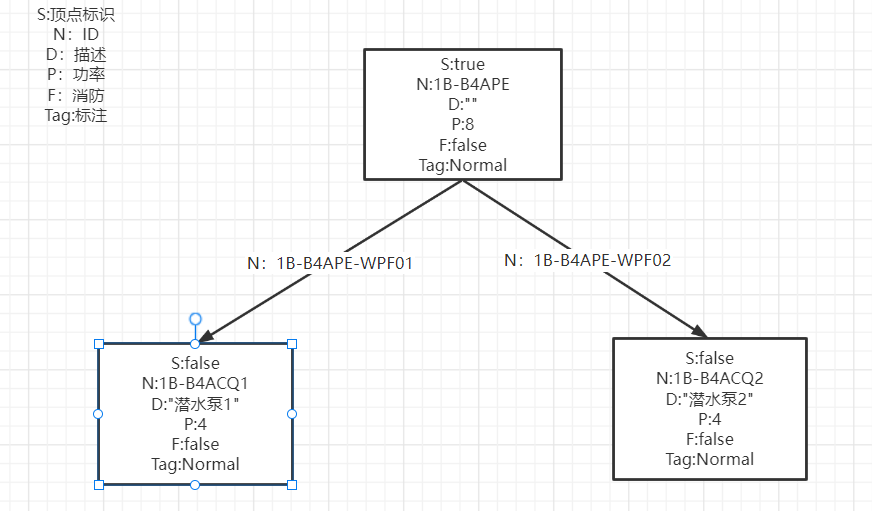
ID：非常重要的属性，是判断两条边是否相同的凭证，如果两个图比较同一

边，两个边的ID相同，可以认为他们是一个边，如果ID不同，那么要认为旧的边被**删除**了，新的边被**创建**了，如果ID相同的边在两个图的位置不同，那么认为这条边被**移位**了

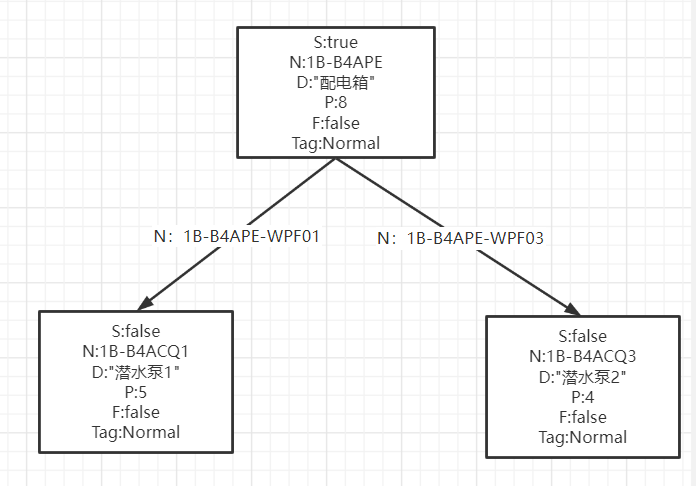
## 举例：

函数样例：public Graph DiffGraph(Graph graphA , Graph graphB);

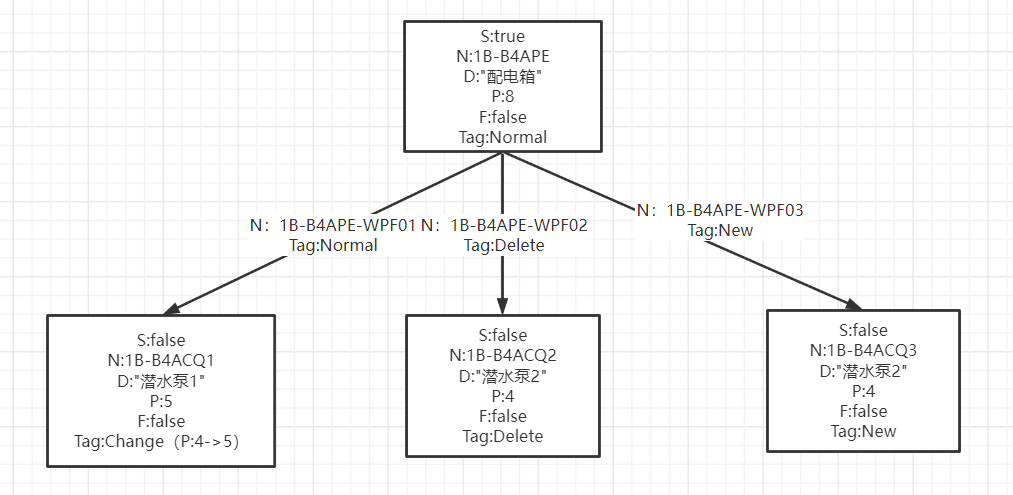
其中graphA ：



graphB：



Return:



其他：

Return里的数据，除了新增类型，其他的原有数据都是在A的基础上去更新，新增类型从B里面拿