对MakeValid的调研结果显示：MakeValid类是geos3.11相对于geos3.6新增的类，MakeValid的唯一功能是检测几何图形的拓扑正确性；MakeValid检测图形正确性过程中，所调用的大部分函数功能在geos3.6版本的库中就已存在；这些检测功能函数包见表1(下页)；分析表1可知；相对于3.6版本，3.11版本将一些冗余功能进行合并，同时增加了判断所有边自交的功能；基于上述比较，同时也考虑到升级geos库从3.6到3.11工作量较大，所以将当前“升级geos库”的想法改为“暂时在3.6版本的基础上构建FMakeValid类”以满足使用需要；构建的FMakeValid除了具有表1所示的geos3.6功能以外，还添加了3.11的任意边的自交检测功能。Geos库中检测边自交的算法是“单调链 + STRtree”，该算法的流程图为图1，同时在图1后也给出了算法实现的详细解释。

|  |  |  |
| --- | --- | --- |
| 3.11 | 3.6 | 功能 |
| checkCoordianteValid | checkInvalidCoordinates | 判断点有效 |
| checkRingsClosed | checkClosedRings | 判断闭环 |
| checkRingsPoinSize | checkTooFewPoints | 判断点数量 |
| checkAreaIntersections | 无 | 判断边自交 |
| checkHolesInShell | checkHolesInShell | 判断洞在shell内 |
| checkHolesNotNested | checkHolesNotNested | 判断洞是否嵌套 |
| checkShellsNotNested | checkShellsNotNested | 判断shell是否嵌套 |
| checkInteriorConnected | checkConnectedInteriors | 判断内部接触 |
| 无 | checkConsistentArea | 检查几何一致性区域 |
| 无 | checkNoSelfIntersectingRings | 判断环自交 |

表1



图1

图1为geos3.11.6版本中makevalid中求解Lines自交的主要函数的流程图；

初始化:对于数据的初始化包括从输入的Geometry类型数据内得到若干边界顶点的顺序排列；

构建单调链(monochains):设初始化得到的n个顶点按P(1)-P(n)排列，根据P(i)相对于P(i-1)在哪个象限，判断其方向为NE(NorthEast),NW(NorthWest),SE(SouthEast),SW(SouthWest)四个方向；当“P(i)相对于P(i-1)的方向”与“P(i-1)相对于P(i-2)的方向”相同时,将P(i-2)-->P(i-1)-->P(i)列入同一个单调链(monochain)，单调链长度不限，以此类推如P(0)-->P(1)-->P(2)-->.......->P(X)。最终得到若干独立的单调链monochains；

构建STRtree:从单调链构建STRtree，将原始的monochains作为STRtree的叶子节点，为若干叶子节点添加一个共同的父节点，一层一层地添加下去，直到某一层得到唯一的父节点即为STRtree的根节点。为每层添加父节点的过程中，首先根据单调链的envelope(bound box)对所有该层节点按照x坐标排序；然后，根据计算的N个子节点对应一个父节点，那么在先前整层排序的基础上，仅对这个N个连续子节点按envelop的y坐标进行排序；排序后并不能直接将这N个子节点连接一个父节点，考虑到影响到查找速度的分支因子，将这N个节点分为B份，每N/B个子节点连接一个父节点，X默认值为10(geos3.11.6库内的设定)；N/B个子节点对应一个父节点，该父节点的envelop为这N/B个子节点envelop的merge结果；

查找是否存在自交:对于monochains中的任意一个单调链L(0),在STRtree树上从根节点开始查找，设在STRtree树上查找到的某一节点为L(i)；只有当L(0)的envelop和L(i)的envelop存在重叠区域时才从L(i)节点继续往下查找，否则跳过L(i)的子节点。当查到STRtree的叶子节点时，设此时的叶子节点为L(j)，则直接比较L(0)与L(j)的自交情况；

判断两个实体单调链是否自交:很容易判断仅存在两个顶点的单调链的自交情况，同时该情况也被作为递归(后面有说明)的结束条件；对于长度大于2的单调链，比较其首尾两点分别相连后形成的线段是否相交；若判断结果不相交，则这两个单调链本身也确定为不相交；若判断结果为相交，则这两个单调了本身可能不相交，则通过二分法，将这两个单调链分别分成两个子链，将子链两两比较，两个子链的比较刚好跟开头的情况一模一样，故而递归执行该函数。

通过递归过程中的记录，得到Geometry内是否存自交情况。