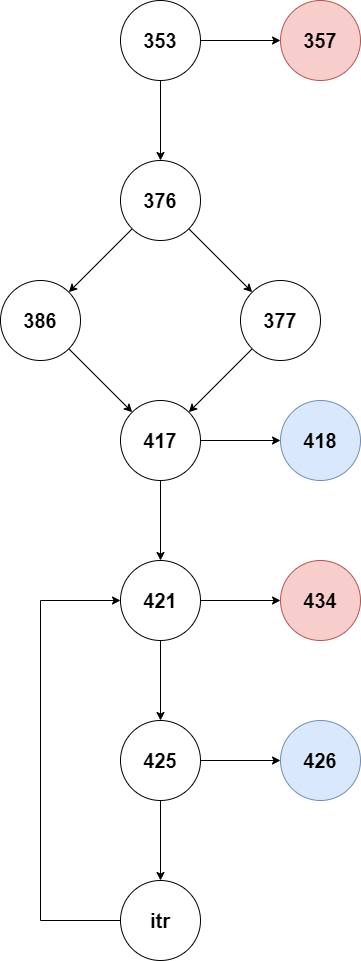
对于Message模块，我们采用基于计算mcCabe复杂度+DFS的方法进行测试。Message模块一共有三个函数，我们选取其中的Query函数作为代表，详细介绍基于mcCabe复杂度+DFS的测试用例的生成过程。而其他函数的测试用例生成过程略去，具体材料我们放在附件之中。

1. 绘制目标函数的程序流图

Query函数的程序流图如下



图**？？？** Query函数的程序流图

图中蓝色节点和红色节点均代表函数出口，但因部分出口为无法进行测试的抛出异常的接口，例如数据库网络出现故障，数据库服务压力较大返回错误等，我们在本次白盒测试中为了简化相应的内容，对于出口部分，我们仅考虑红色出口。

1. 根据路径计算mcCabe复杂度

由于我们的函数都是多出口的函数，为了使图成为强连通图，从图的出口点到入口点加一条用虚线表示的有向边，使图成为强连通图。之后使用：

进行计算，由于我们假设数据库能正常工作，数据库数据结构正确，因此相关decode和数据库连接的err检查均不在白盒测试的覆盖范围内。相关出口在图例中将以蓝色体现，逻辑错误退出在图例中以红色体现。在此种情况下我们再进行mcCabe复杂度的计算。

message 模块一共有三个函数：

(a \*srv) Create(ctx context.Context, req \*message.MessageCreateRequest, rsp \*message.MessageCreateResponse) error

其节点数量为16，边数量为17。红色出口数量为5，最终得出

(a \*srv) Find(ctx context.Context, req \*message.MessageFindRequest, rsp \*message.MessageFindResponse) error

其节点数量为19，边数量为22，红色出口数量为4，最终得出

(a \*srv) Query(ctx context.Context, req \*message.MessageQueryRequest, rsp \*message.MessageQueryResponse) error

其节点数量为9，边数量为10，红色出口数量为2，最终得出

1. 统计变量的定义/使用节点

根据程序流图，对其进行DFS，找出mcCabe条基路径，并根据对应的路径生成测试用例。

|  |  |  |  |
| --- | --- | --- | --- |
| 数据库 | | | |
| fromUser | toUser | badge | infos |
| 1 | 2 | 1 | [{unread:false,forward:false},{unread:false,forward:false}] |
| 2 | 1 | 1 | [{unread:false,forward:false},{unread:false,forward:false}] |
| 3 | 1 | 1 | [{unread:false,forward:true},{unread:false,forward:true}] |
| 1 | 3 | 1 | [{unread:true,forward:true},{unread:true,forward:true}] |

表**？？？** Query函数测试的数据库预插入数据

|  |  |
| --- | --- |
| 路径 | 请求数据 |
| 353-376-421-425-itr-421-434 | {userID:1, oldMsg:1} |
| 353-376-421-425-itr-421-434 | {userID:4, oldMsg:0} |
| 353-357 | {userID:null} |
| 353-376-386-421-425-itr-421-434 | {userID:3, oldMsg:0} |

表**？？？** Query函数路径与测试用例

完整的测试用例详见源代码和excel表格。

1. 测试反馈与感想

从理论而言，通过计算mcCabe复杂度并结合深度优先搜索所得出的测试用例一定能够完全覆盖所有路径，从而覆盖所有代码，message模块净利润率为93.8%，剩余的6%的代码为数据库错误返回检查，只能在数据库访问压力较大或服务与数据库之间网络出现问题时才能体现，故认为不应当纳入白盒测试范围。