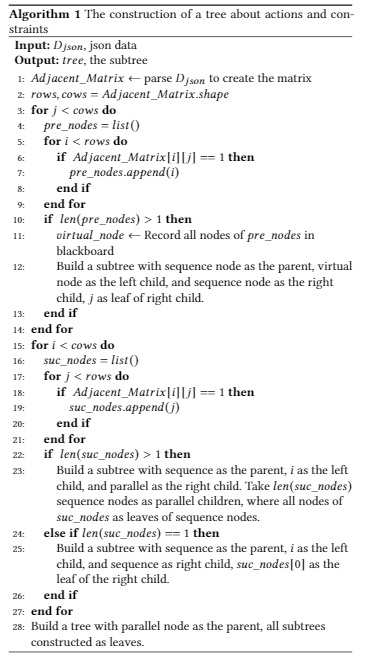
行为树构建算法

行为树构建算法包括两个部分，一个是execute中行为的子树构建一个是oracle中cretiria子树的构建

## 1.behavior子树构建



首先根据execution中的动作，获得所有的行为，约束条件创建矩阵，以行为数作为邻接矩阵的行列大小，按照execution中的时序关系创建有向图的邻接矩阵，有向图的节点是execution中的行为，边从execution中先发生的行为指向后发生的行为。

根据一个节点的前驱数目将邻接矩阵拆分，按列遍历邻接矩阵，判断节点的前驱数目，如果一个节点没有前驱将其暂存到列表中；如果一个节点有多个前驱，创建触发行为，它和此列代表的行为共同作为一个Sequence节点的子行为，最后将这个Sequence节点作为根节点的子节点。

按行遍历邻接矩阵，如果该行只有一个关系，则此行代表的行为只有一个后继，将其加入该行为的父节点即可，因为它的父节点是Sequence行为。如果该行有多个关系，则此行代表的行为有多个后继，将这些后继作为一个Parallel行为的子节点，再将Parallel节点作为此行为的父节点的子节点。

最后将没有前驱的节点作为一个Parallel节点的子节点，此Parallel节点是begin的Sequence父节点的子节点。

一个图对应的行为树的样例如图。C1和B2是B1的后继。B2和B3有一个动作触发后，B4开始执行。B表示行为，C表示约束条件。

图1测试样例图

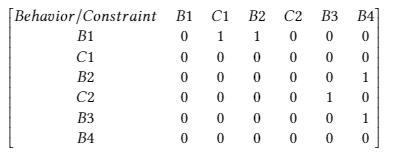
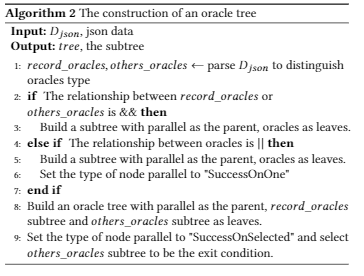


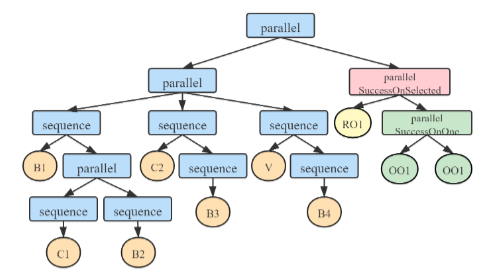
图2有向图对应的邻接矩阵

## 2.oracle子树构建



我们解析了json数据，并将oracle类型分为record、period和final。Period和final这两种类型会影响到模拟数据的生成，所以我们将它们存储在𝑜𝑡ℎ𝑒𝑟𝑠\_ora𝑐𝑙𝑒𝑠。我们首先为𝑟𝑒𝑐𝑜𝑟𝑑\_ora𝑐𝑙𝑒𝑠创建两个子树。如果oracle之间的关系是&&，我们将parallel作为父节点，表示这些oracles必须同时得到满足；如果关系是||，我们将parallel的类型设置为SuccessOnOne，表示至少有一个oracle得到了满足。第8行表示程序将两个子树合并为一个完整的oracle树。根节点parallel被设置为SuccessOnSelected，指定𝑜𝑡ℎ𝑒𝑟𝑠\_ora𝑐𝑙𝑒𝑠为退出条件。

## 3.构建完整的行为树



完整的Oracle子树

我们将算法1和算法2创建的两个子树合并为一个完整的行为树。我们把parallel作为父节点，把行为树和oracle树作为叶子。我们指定paralle节点的类型为SuccessOnSelected，oracle子树为退出条件。