

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 criteria 子树的构建

1.behavior 子树构建

Algorithm 1 The construction of a tree about actions and constraints

Input: D_{json} , json data
Output: $tree$, the subtree

```
1:  $Adjacent\_Matrix \leftarrow \text{parse } D_{json}$  to create the matrix
2:  $rows, cows = Adjacent\_Matrix.shape$ 
3: for  $j < cows$  do
4:    $pre\_nodes = list()$ 
5:   for  $i < rows$  do
6:     if  $Adjacent\_Matrix[i][j] == 1$  then
7:        $pre\_nodes.append(i)$ 
8:     end if
9:   end for
10:  if  $len(pre\_nodes) > 1$  then
11:     $virtual\_node \leftarrow$  Record all nodes of  $pre\_nodes$  in blackboard
12:    Build a subtree with sequence node as the parent, virtual node as the left child, and sequence node as the right child,  $j$  as leaf of right child.
13:  end if
14: end for
15: for  $i < cows$  do
16:    $suc\_nodes = list()$ 
17:   for  $j < rows$  do
18:     if  $Adjacent\_Matrix[i][j] == 1$  then
19:        $suc\_nodes.append(j)$ 
20:     end if
21:   end for
22:  if  $len(suc\_nodes) > 1$  then
23:    Build a subtree with sequence as the parent,  $i$  as the left child, and parallel as the right child. Take  $len(suc\_nodes)$  sequence nodes as parallel children, where all nodes of  $suc\_nodes$  as leaves of sequence nodes.
24:  else if  $len(suc\_nodes) == 1$  then
25:    Build a subtree with sequence as the parent,  $i$  as the left child, and sequence as right child,  $suc\_nodes[0]$  as the leaf of the right child.
26:  end if
27: end for
28: Build a tree with parallel node as the parent, all subtrees constructed as leaves.
```

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

<i>Behavior/Constraint</i>	<i>B1</i>	<i>C1</i>	<i>B2</i>	<i>C2</i>	<i>B3</i>	<i>B4</i>
<i>B1</i>	0	1	1	0	0	0
<i>C1</i>	0	0	0	0	0	0
<i>B2</i>	0	0	0	0	0	1
<i>C2</i>	0	0	0	0	1	0
<i>B3</i>	0	0	0	0	0	1
<i>B4</i>	0	0	0	0	0	0

图 2 有向图对应的邻接矩阵

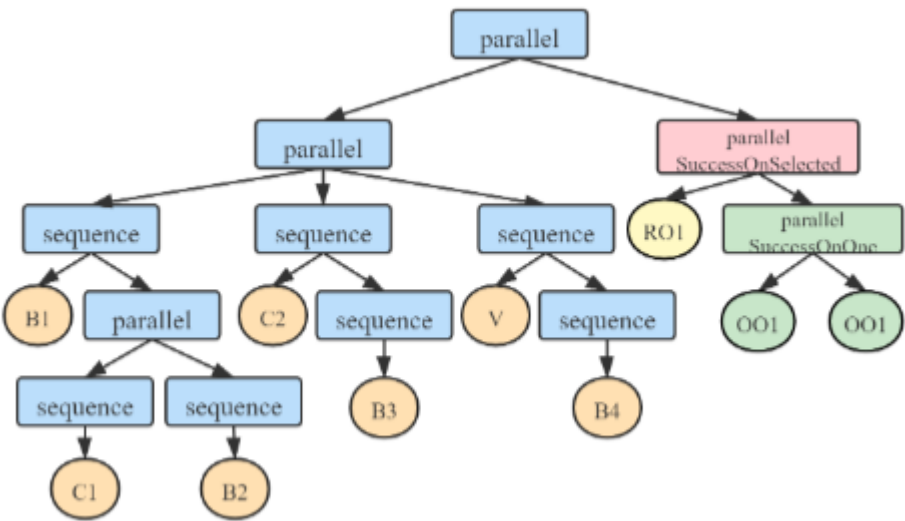
2.oracle 子树构建

Algorithm 2 The construction of an oracle tree
Input: D_{json} , json data
Output: <i>tree</i> , the subtree
1: <i>record_oracles</i> , <i>others_oracles</i> \leftarrow parse D_{json} to distinguish oracles type
2: if The relationship between <i>record_oracles</i> or <i>others_oracles</i> is && then
3: Build a subtree with parallel as the parent, oracles as leaves.
4: else if The relationship between oracles is then
5: Build a subtree with parallel as the parent, oracles as leaves.
6: Set the type of node parallel to "SuccessOnOne"
7: end if
8: Build an oracle tree with parallel as the parent, <i>record_oracles</i> subtree and <i>others_oracles</i> subtree as leaves.
9: Set the type of node parallel to "SuccessOnSelected" and select <i>others_oracles</i> subtree to be the exit condition.

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 parallel 节点的类型为 SuccessOnSelected，oracle 子树为退出条件。