## 行为树的基本概念

行为树的执行从根结点开始，根结点以给定频率生成 Tick 信 号。这些信号允许执行一个结点，并传递给选中结点的一个或多个子结点。当且 仅当结点接收到 Tick 信号时才会执行该结点。如果子结点的执行还没有结束，或 者它已经实现了目标，则分别向父结点返回运行（Running）和成功（Success）， 否则返回失败（Failure）。

存在顺序、回退、并行、装饰（Sequence 、Fallback 、Parallel 、Decorator）四 类控制流结点和动作、条件（Action 、Condition）两类执行结点，表 1.1 进行了说 明和总结。行为树利用分层的树状结构来组织任务转换逻辑，不同的转换逻辑由 具有不同语义的控制流结点或者它们的组合给出。

表 1.1 行为树结点类型

|  |  |  |  |
| --- | --- | --- | --- |
| **返回状态** **结点类型** | **成功（Success）** | **失败（Failure）** | **运行（Running）** |
| 顺序  （Sequence） | 如果所有子结点成功 | 如果 1 个子结点失败 | 如果 1 个子结点 返回运行 |
| 回退  （Fallback） | 如果 1 个子结点成功 | 如果所有子结点失败 | 如果 1 个子结点 返回运行 |
| 并行  （Parallel） | 如果≥ M个  子结点成功 | 如果> (N 一 M)个 子结点失败 | 其余情况 |
| 装饰  （Decorator） | 自定义 | 自定义 | 自定义 |
| 动作  （Action） | 完成后 | 当不可能完成时 | 完成期间 |
| 条件  （Condition） | 如果满足条件 | 如果不满足条件 |  |

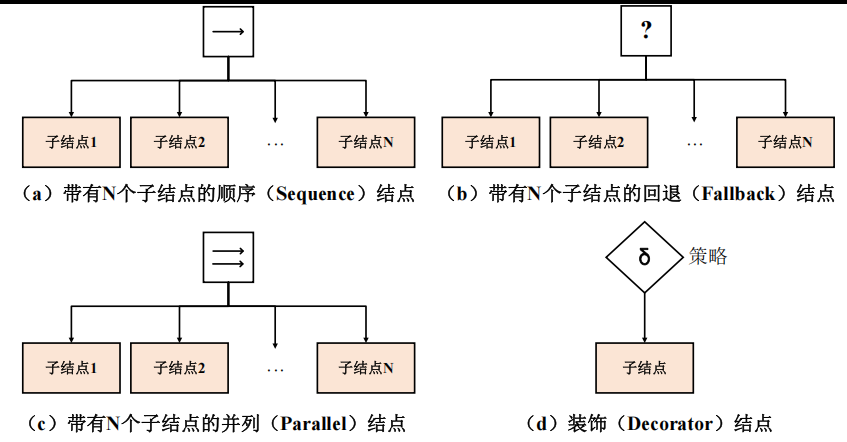


图 2.2 行为树控制流结点图形表示

* 顺序（Sequence）结点：顺序结点将 tick 信号从左到右路由到其子结点，当 且仅当所有子结点都返回 Success 时，该结点才返回 Success；否则，当找到返回 Failure 或 Running 的子结点时，它将接收到的状态返回给其父结点。当有一个子 结点返回 Running 或 Failure 时，顺序结点将不会把 tick 信号路由到下一个子结点 （如果存在下一个子结点）。顺序结点的符号如图2.2（a）所示。
* 回退（Fallback）结点：回退结点将 tick 信号从左到右路由到其子结点，当且 仅当所有子结点都返回 Failure 时，该结点才返回 Failure；否则， 当找到返回 Success 或 Running 的子结点，它将接收到的状态返回给其父结点。当有一个子结 点返回 Running 或 Success 时，回退结点将不会把时钟信号路由到下一个子结点 （如果存在下一个子结点）。 回退结点的符号如图2.2（b）所示。
* 并行（Parallel）结点：并行结点将 tick 信号同时发送到所有子结点，如果其 N个子结点中超过阈值M个子结点返回 Success，则返回 Success；如果超过(N - M)个子结点返回 Failure，则返回 Failure；上述条件都不满足时返回 Running。并 行结点的符号如图 2.2（c）所示。
* 装饰（Decorator）结点：装饰结点是带有一个子结点的控制流结点，该结点 根据用户定义的规则操纵其子结点的返回状态，并根据预定义的规则有选择性地 触发子结点。装饰结点的符号是菱形，如图2.2（d）所示。



**动作**



**条件**

**（a）动作（Action）结点** **（b）条件（Condition）结点**

图 2.3 行为树执行结点图形表示

* 动作（Action）结点：当动作结点接收到 tick 信号时将执行命令。如果动作 成功完成，则返回 Success；如果动作失败，则返回 Failure。当动作正在运行时， 它返回 Running 。动作结点如图2.3（a）所示。
* 条件（Condition）结点：当条件结点接收到 tick 信号时，检查自定义条件。 基于条件是否被满足返回 Success 或 Failure。条件结点不会返回 Running 状态，条 件结点如图2.3（b）所示。