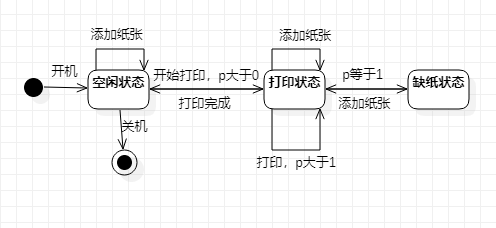
使用transition tree的方法。首先根据上课内容绘制打印系统的状态图：



覆盖策略1（节点覆盖）：

1. 开机
2. 开始打印
3. 打印直到缺纸状态
4. 添加纸张
5. 打印
6. 打印完成
7. 关机

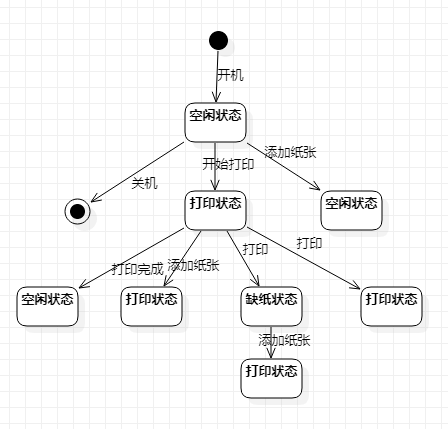
覆盖策略2（方法覆盖）：

1. 开机
2. 开始打印
3. 打印到缺纸状态
4. 添加纸张
5. 关机

覆盖策略3（边覆盖）

1. 开机
2. 添加纸张
3. 开始打印
4. 添加纸张
5. 打印直到缺纸状态
6. 添加纸张
7. 打印直到完成
8. 关机

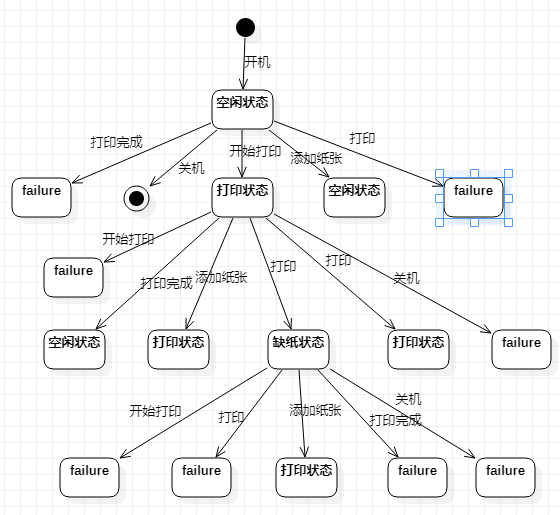
使用转化算法将状态图转化为树状图如下：



可以看出，该数有6个叶子节点，为了覆盖所有的叶子节点我们需要设计6个测试用例，设计的测试用例如下：

|  |  |
| --- | --- |
| 测试用例编号 | 活动 |
| 1 | 1. 开机 2. 关机 |
| 2 | 1. 开机 2. 添加纸张 |
| 3 | 1. 开机 2. 开始打印 3. 打印完成 |
| 4 | 1. 开机 2. 开始打印 3. 添加纸张 |
| 5 | 1. 开机 2. 开始打印 3. 打印 4. 添加纸张 |
| 6 | 1. 开机 2. 开始打印 3. 打印 |

接下来进行健壮性的考虑，对每个节点添加缺少的非法功能来达到failure状态



可以看出，该数有14个叶子节点，为了覆盖所有的叶子节点我们需要设计14个测试用例，设计的额外测试用例如下：

|  |  |
| --- | --- |
| 测试用例编号 | 活动 |
| 7 | 1. 开机 2. 打印完成 |
| 8 | 1. 开机 2. 开始打印 3. 开始打印 |
| 9 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 开始打印 |
| 10 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 打印 |
| 11 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 打印完成 |
| 12 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 关机 |
| 13 | 1. 开机 2. 开始打印 3. 关机 |
| 14 | 1. 开机 2. 打印 |