

## 3.2 与或树的宽度搜索与深度搜索



西安电子科技大学  
XIDIAN UNIVERSITY



### 人工智能导论

#### 与或图的盲目式搜索（上）

主讲人：刘若辰

西安电子科技大学 人工智能学院



➤ 与或图搜索：在与或图上执行搜索的过程，其目的在于标明起始节点是有解的，即搜索不是去寻找到目标节点的一条路径，而是寻找一个解树。

- 执行可解节点标志和不可解点标志过程。
- 解树：由能够证明初始节点是可解的可解节点构成的连通的子图。

➤ 与或图/与或树

- 与或树：除初始节点，其余节点只有一个父节点
- 与或图：除初始节点，其余节点允许有多个父节点

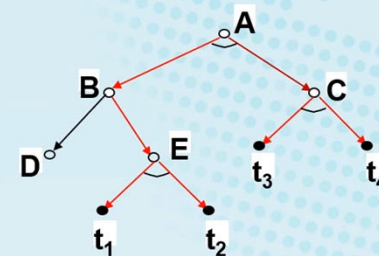


图3.5：与或树

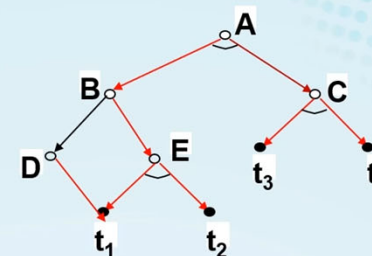


图3.6：与或图



## 与或树搜索

### ➤ 与或树的一般搜索过程：

- (1) 把原始问题作为初始节点 $S_0$ ，并把它作为当前节点；
- (2) 应用分解或等价变换对当前节点进行扩展；
- (3) 为每个节点设置指向父节点的指针；
- (4) 选择适合的节点作为当前节点，反复执行第(2)步和第(3)步，在此其间要多次调用可解标志过程和不可解标志过程，直到初始节点被标为可解节点或不可解节点为止。

### ➤ 与或树的搜索的目的是寻找解树，从而求得原始问题的解。



## 与或树搜索

### ➤ 与或树搜索与状态空间图搜索的区别：

- 搜索目的不同：是证明起始节点是否可解，而可解节点是递归定义的，取决于后继节点是否可解，即搜索过程是能否找到可解的叶节点。
- 结果不同：若初始节点被标志为可解，则搜索成功结束；若初始节点被标志为不可解，则搜索失败。
- 节点处理不同：一旦发现不可解节点，应把该节点从图中删去。



## 与或树搜索

与/或树的搜索策略

盲目式搜索

- 宽度优先搜索
- 深度优先搜索
- 等代价搜索

启发式搜索

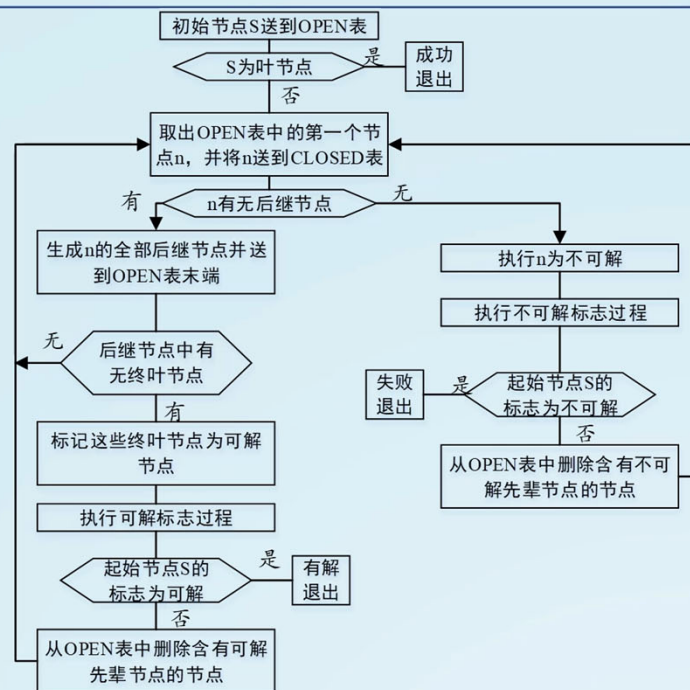


## 与或树的宽度优先搜索

- 与或树的**宽度**优先搜索的基本思想：
  - 按照“**先产生的节点先扩展**”的原则进行搜索
  - 搜索过程中要多次调用可解标志过程和不可解标志 过程
  - OPEN表、CLOSED表类似状态空间搜索
    - OPEN表：存放待扩展的节点
    - CLOSED表：存放已扩展的节点



## 流程图

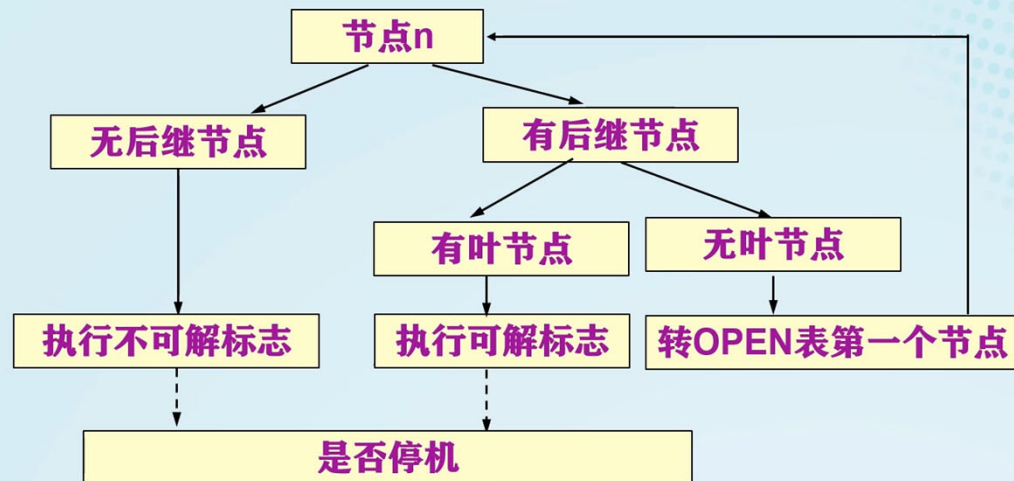






## 与或树的宽度优先搜索

➤ **关键步骤：**扩展节点n，生成其全部后继节点，送到OPEN表末端，并设置指向n的指针。

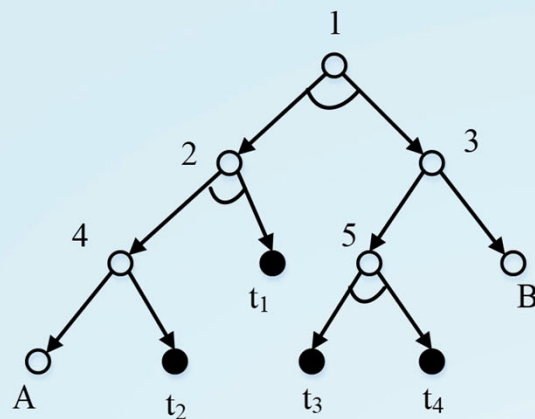






## 与或树的宽度优先搜索

**例：**设有如图所示的与/或树，其中  $t_1, t_2, t_3, t_4$  均为终叶节点，A和B是不可解的端节点。用**宽度优先搜索法**求出解树。





## 与或树的宽度优先搜索

### ➤ 初始化:

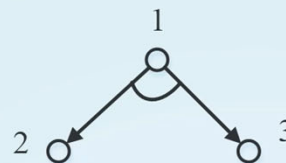
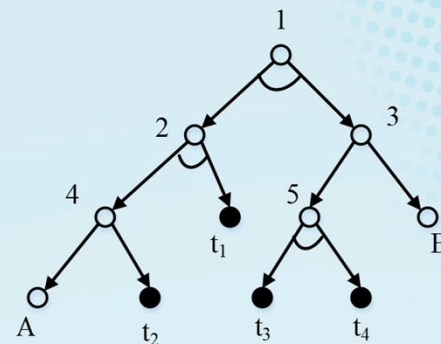
节点1送到OPEN表, 且不为终叶节点

OPEN表	CLOSED 表
1	

### ➤ STEP1:

- 把节点1放入CLOSED表, 扩展, 得到节点2、3;
- 节点2、3都不是终叶节点, 接着扩展节点2, 此时OPEN表只剩节点3。

OPEN表	CLOSED 表
1	
2,3	1
3	1,2



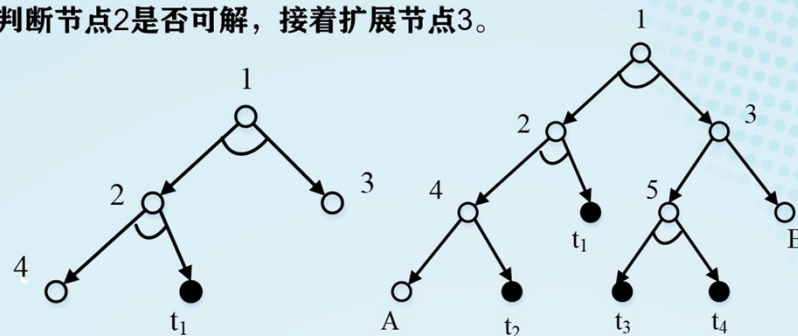


## 与或树的宽度优先搜索

### ➤ STEP2:

- 扩展节点2后，得到节点4、 $t_1$ ，此时OPEN表中的节点有3，4， $t_1$ ；
- 节点 $t_1$ 是终叶节点且为可解节点，对其先辈节点进行标志；
- $t_1$ 的父节点是“与”节点，无法判断节点2是否可解，接着扩展节点3。

OPEN表	CLOSED表
1	
2,3	1
3,4, $t_1$	1,2
4, $t_1$	1,2,3



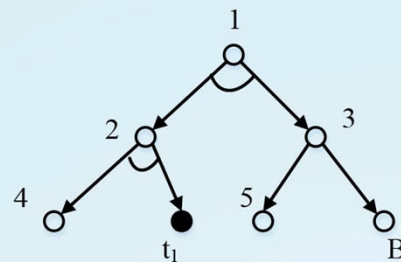
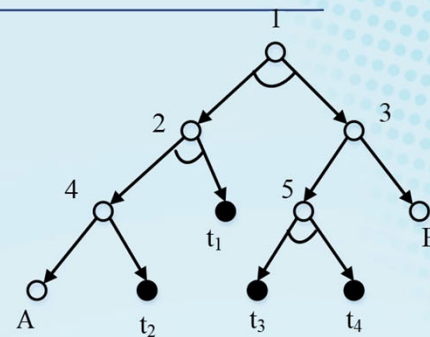


## 与或树的宽度优先搜索

### ➤ STEP3:

- 扩展节点3后，得到节点5、B；
- 节点5，B都不是终叶节点，接着扩展节点4。

OPEN表	CLOSED 表
1	
2,3	1
3,4,t <sub>1</sub>	1,2
4,t <sub>1</sub> ,5,B	1,2,3
t <sub>1</sub> ,5,B	1,2,3,4





## 与或树的宽度优先搜索

### STEP4:

- 扩展节点4后，得到节点A、 $t_2$ ；
- 节点 $t_2$ 是终叶节点且为可解节点，对其先辈节点进行标志；
- 节点4、2可解，但不确定1是否可解；从OPEN表中删掉A；
- 此时节点5是OPEN表中第一个待考察的节点，下一步将扩展节点5。

OPEN表	CLOSED表
1	
2,3	1
3,4, $t_1$	1,2
4, $t_1$ ,5,B	1,2,3
$t_1$ ,5,B,A, $t_2$	1,2,3,4
5,B, $t_2$	1,2,3,4, $t_1$
B, $t_2$	1,2,3,4, $t_1$ ,5





## 与或树的宽度优先搜索

### ➤ STEP5:

- 扩展节点5后，得到节点 $t_3$ 、 $t_4$ ；
- 节点 $t_3$ 、 $t_4$ 都是终叶节点，且为可解节点，对其先辈节点进行标志；
- 节点5可解，接着推出节点3可解，节点1可解，从OPEN表中删掉B，成功退出。

OPEN表	CLOSED表
1	
2,3	1
3,4, $t_1$	1,2
4, $t_1$ ,5,B	1,2,3
$t_1$ ,5,B,A, $t_2$	1,2,3,4
5,B, $t_2$	1,2,3,4, $t_1$
B, $t_2$ , $t_3$ , $t_4$	1,2,3,4, $t_1$ ,5
$t_2$ , $t_3$ , $t_4$	1,2,3,4, $t_1$ ,5

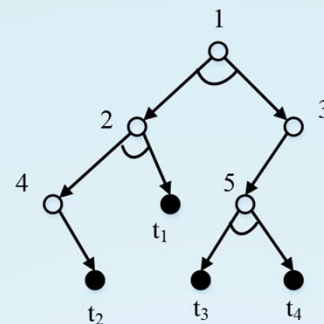
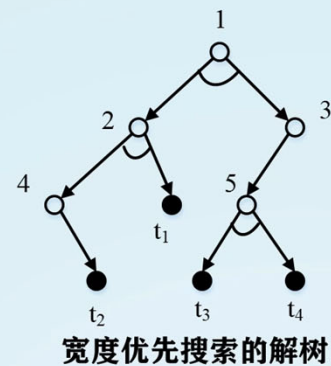
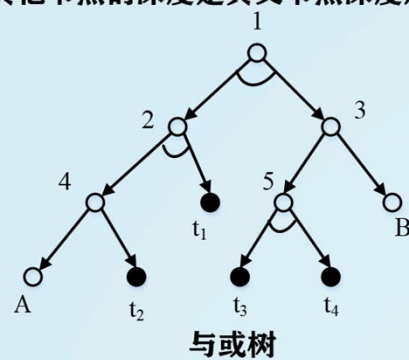


图3.7：解树



## 与或树的宽度优先搜索

- “先产生的节点先被扩展”
- 如果问题有解，宽度优先搜索一定会求得一个解树，而且其最深的叶节点具有最小的深度。
  - 初始节点深度为零
  - 其他节点的深度是其父节点深度加1







西安电子科技大学  
XIDIAN UNIVERSITY



# 人工智能导论

## 与或图的盲目式搜索（下）

主讲人：刘若辰

西安电子科技大学 人工智能学院



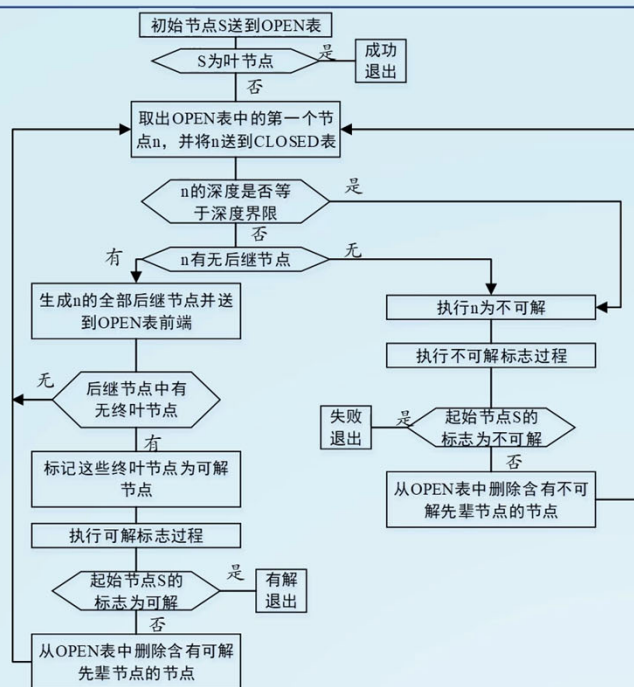
## 与或树的深度优先搜索

### ➤ 与或树的深度优先搜索

- 按照“**新产生的节点先扩展**”的原则进行搜索
- 设置深度界限
- 搜索过程中要多次调用可解标志过程和不可解标志过程



## 流程图





## 与或树的深度优先搜索

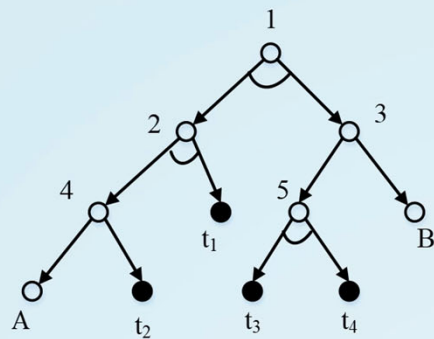
➤ 与宽度优先算法相比，深度优先算法的特殊之处：

- 第4步：要判断从OPEN表取出来的**节点的深度**。如果等于深度界限，认定它为不可解节点。
- 第6步：扩展节点n将其子节点放入OPEN表的**前端**，即**新产生的节点先扩展**。



## 与或树的深度优先搜索

**例：**设有如图所示的与/或树，其中 $t_1, t_2, t_3, t_4$ 均为终叶节点，A和B是不可解的端节点。采用**深度优先搜索法**求出解树。（规定深度界限为4）





## 与或树的深度优先搜索

### ➤ 初始化:

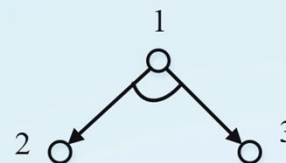
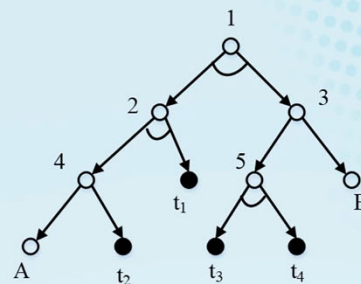
节点 1 送到OPEN表, 且不为终叶节点

OPEN表	CLOSED 表
1	

### ➤ STEP1:

- 扩展节点1, 得到节点2、3;
- 节点2、3都不是终叶节点, 接着扩展节点2, 此时OPEN表只剩节点3。

OPEN表	CLOSED 表
1	
2,3	1
3	1,2



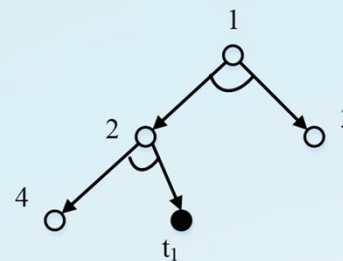
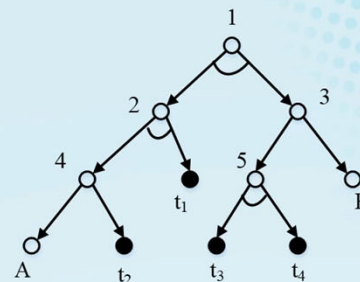


## 与或树的深度优先搜索

### ➤ STEP2:

- 扩展节点2后，得到节点4、 $t_1$ ；
- $t_1$ 是终叶节点，但无法标识节点2；
- 继续扩展节点4。

OPEN表	CLOSED 表
1	
2,3	1
4, $t_1$ ,3	1,2
$t_1$ ,3	1,2,4





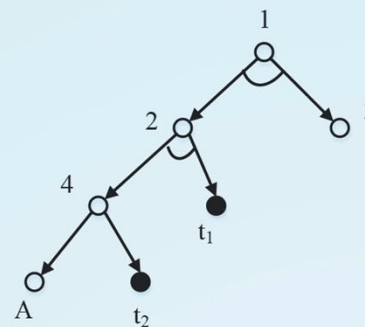
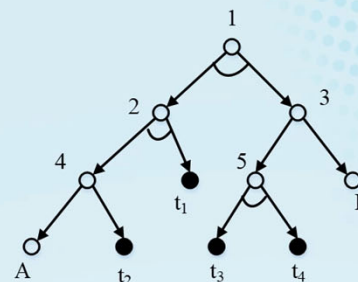


## 与或树的深度优先搜索

### ➤ STEP3:

- 扩展节点4后，得到节点A、 $t_2$ ;
- 标志4、2为可解节点，但不能确定1是否可解
- 删掉OPEN中的节点A，接着扩展节点3。

OPEN表	CLOSED 表
1	
2,3	1
4,t <sub>1</sub> ,3	1,2
A,t <sub>2</sub> ,t <sub>1</sub> ,3	1,2,4
3	1,2,4,t <sub>2</sub> ,t <sub>1</sub>



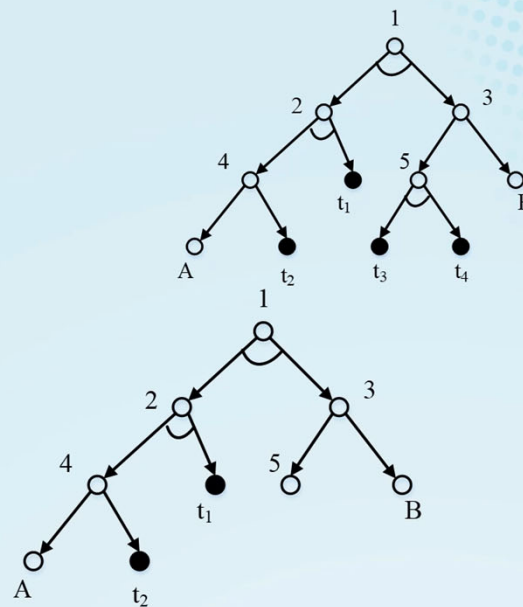


## 与或树的深度优先搜索

### ➤ STEP4:

- 扩展节点3后，得到节点5、B；
- 接着扩展节点5。

OPEN表	CLOSED 表
1	
2,3	1
4,t <sub>1</sub> ,3	1,2
A,t <sub>2</sub> ,t <sub>1</sub> ,3	1,2,4
3	1,2,4,t <sub>2</sub> ,t <sub>1</sub>
5,B	1,2,4,t <sub>2</sub> ,t <sub>1</sub> ,3
B	1,2,4,t <sub>2</sub> ,t <sub>1</sub> ,3,5



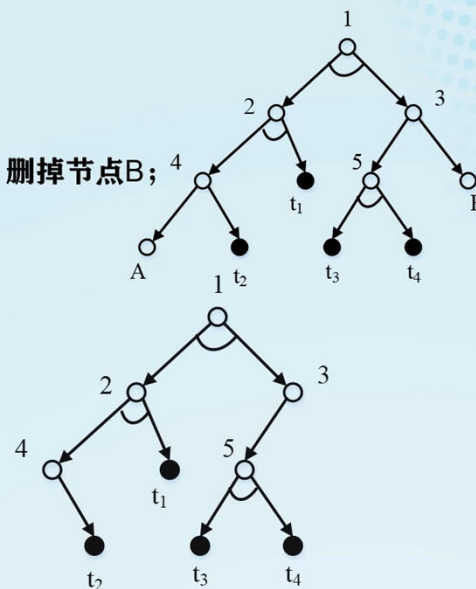


## 与或树的深度优先搜索

### ➤ STEP5:

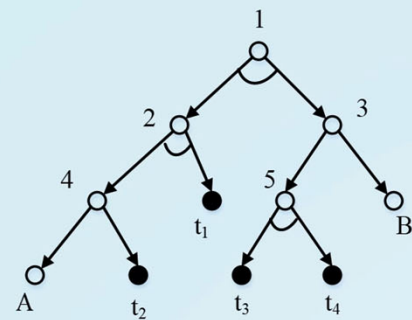
- 扩展节点5后，得到节点 $t_3, t_4$ ;
- 标志可解节点5, 3, 推出初始节点1可解，删掉节点B;
- 成功退出。

OPEN表	CLOSED表
1	
2,3	1
4, $t_1$ , 3	1, 2
A, $t_2$ , $t_1$ , 3	1, 2, 4
3	1, 2, 4, $t_2$ , $t_1$
5, B	1, 2, 4, $t_2$ , $t_1$ , 3
$t_3, t_4, B$	1, 2, 4, $t_2$ , $t_1$ , 3, 5
$t_3, t_4$	1, 2, 4, $t_2$ , $t_1$ , 3

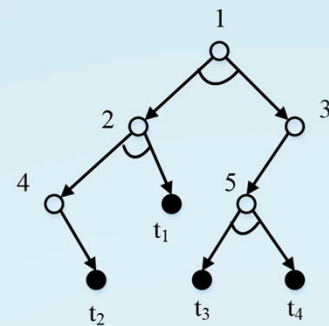




## 与或树的深度优先搜索



与或树



深度优先搜索的解树