

## BDD

### BDT

**BDT**,即二叉决策树, 设 $T$ 是一棵有限二叉判定树。按如下方式,  $T$ 确定以非终止结点为变量的唯一布尔函数。给定 $T$ 中出现的布尔变量赋值0或1, 从 $T$ 的根开始, 只要当前结点的变量值为0, 那么沿虚线走; 否则沿实线走。函数值就是所到达的终止结点的值。

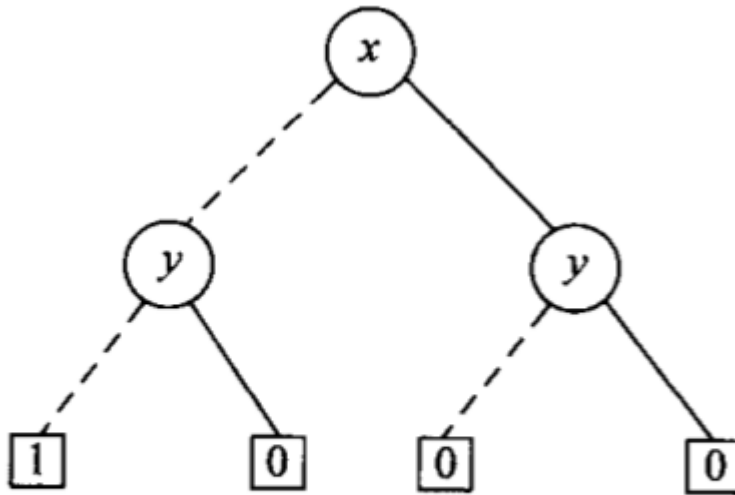


图 6-2 二叉判定树的例子

## BDD

**BDD**(Binary Decision Diagram), 即二叉决策图, 是一种用于表示布尔函数的数据结构。它通过DAG的形式, 将布尔函数的复杂逻辑关系以图形化的方式展现出来。

**定义:** 二叉判定图是一个具有唯一初始节点的有限有向无环图, 其所有终止结点标记为0或1, 所有非终止结点用布尔变量标记。每个非终止结点恰好有两条边指向其他结点: 一个标记为0, 一个标记为1 (分别用实线和虚线表示)。

简约BDD是通过BDD使用 $C_1 \sim C_3$ 的方式化简到不能再化简得来的

## Note

- C1:去掉重复的终止节点。
- C2:去掉冗余的测试。
- C3: 去掉重复的非终止节点

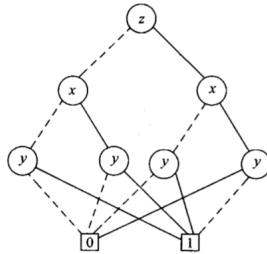


图 6-4 具有重复子 BDD 的 BDD

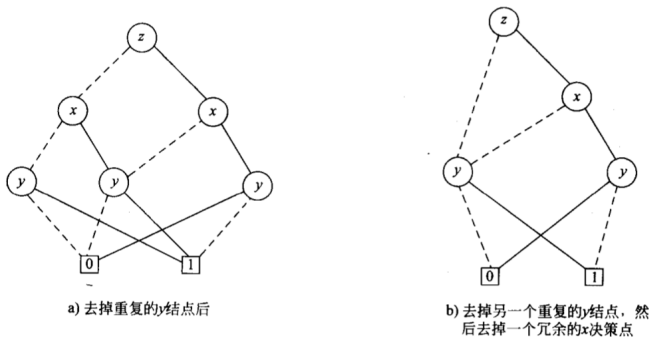


图 6-5 图 6-4 的 BDD

## OBDD

OBDD是BDD的一种特殊形式，它是带有有序变量表的BDD，不仅使得BDD的结构变得简单统一，并且使得很多操作更加高效。

- 由OBDD的定义可知，沿一条路径不能的任意变量都不能出现多次

$[x_1, \dots, x_n]$  是无重复的有序变量表，B是一个 BDD，其所有变量都出现在该表的某处。我们说B有次序  $[x_1, \dots, x_n]$ ，如果B的所有变量标记都出现在该表中，且沿B中任何路径，对 $x_j$ 跟随 $x_i$ 后边的每次出现，都有 $i < j$ 。

## 简约OBDD的算法

对两个OBDD操作时，它们要有相容的变量序(compatible variable order)

在《面向计算机科学的数理系统建模与推理》这本书中，OBDD的操作主要有以下几

种：

1.reduce() 对OBDD进行化简

算法	输入 OBDD	输出 OBDD	时间复杂度
reduce	$B$	简约的 $B$	$O( B  \cdot \log  B )$
apply	$B_f, B_g$ (简约的)	$B_{f \circ g}$ (简约的)	$O( B_f  \cdot  B_g )$
restrict	$B_f$ (简约的)	$B_{f[0/x]}$ 或 $B_{f[1/x]}$ (简约的)	$O( B_f  \cdot \log  B_f )$
$\exists$	$B_f$ (简约的)	$B_{\exists x_1. \exists x_2. \dots \exists x_n. f}$ (简约的)	NP 完备的

图 6-23 我们实现的布尔公式算法在最坏情形下关于输入 OBDD 的运行时间上界