故障树分析
Fault Tree Analysis

北京航空航天大学工程系统工程系

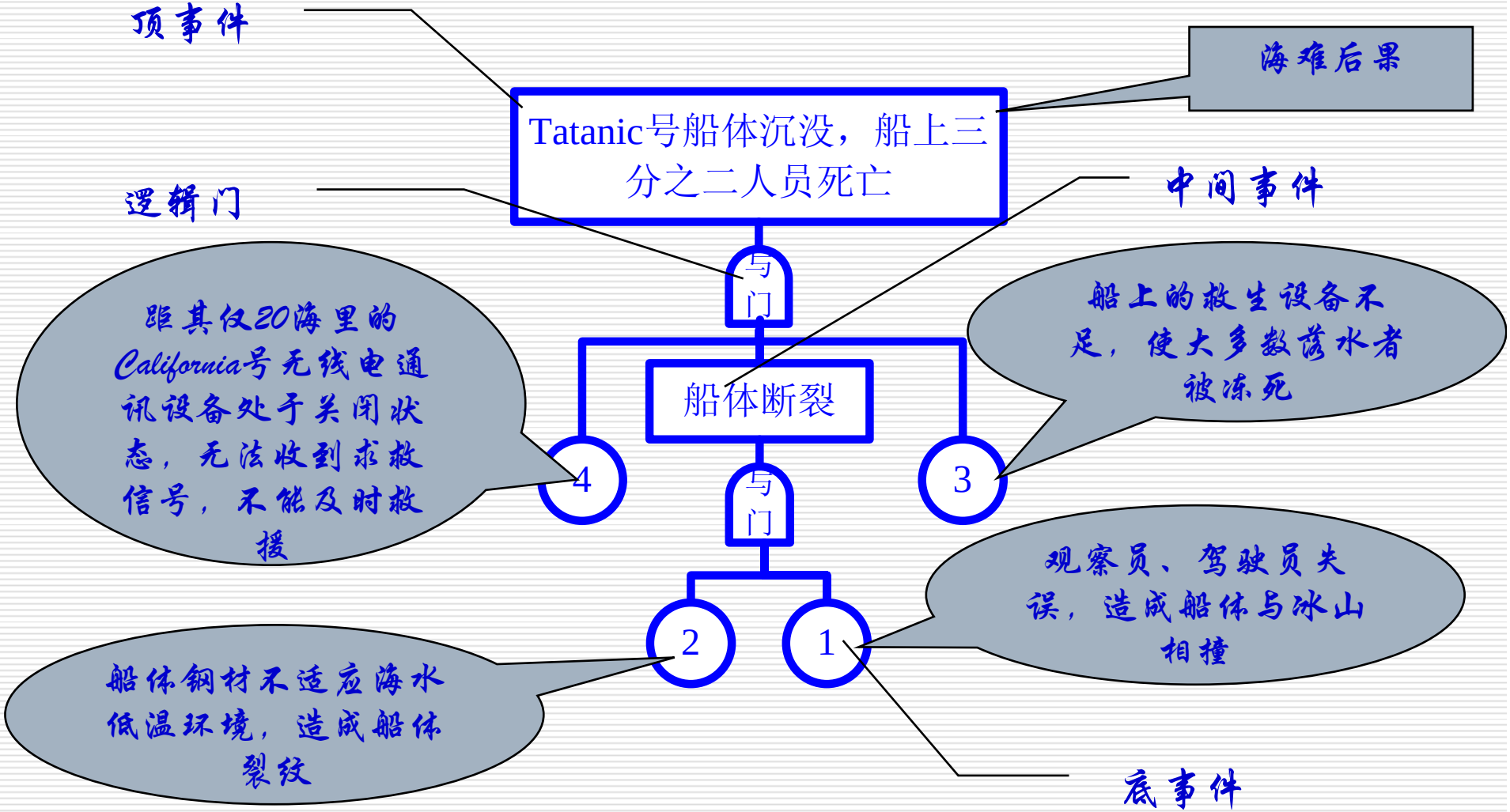
内容提要

- 概述
- 故障树的基本概念
  - 定义
  - 目的、特点
  - FTA工作要求
  - 常用事件、逻辑门符号
- 故障树分析
  - 定性分析
  - 定量分析
  - 重要度分析
- 故障树的简化

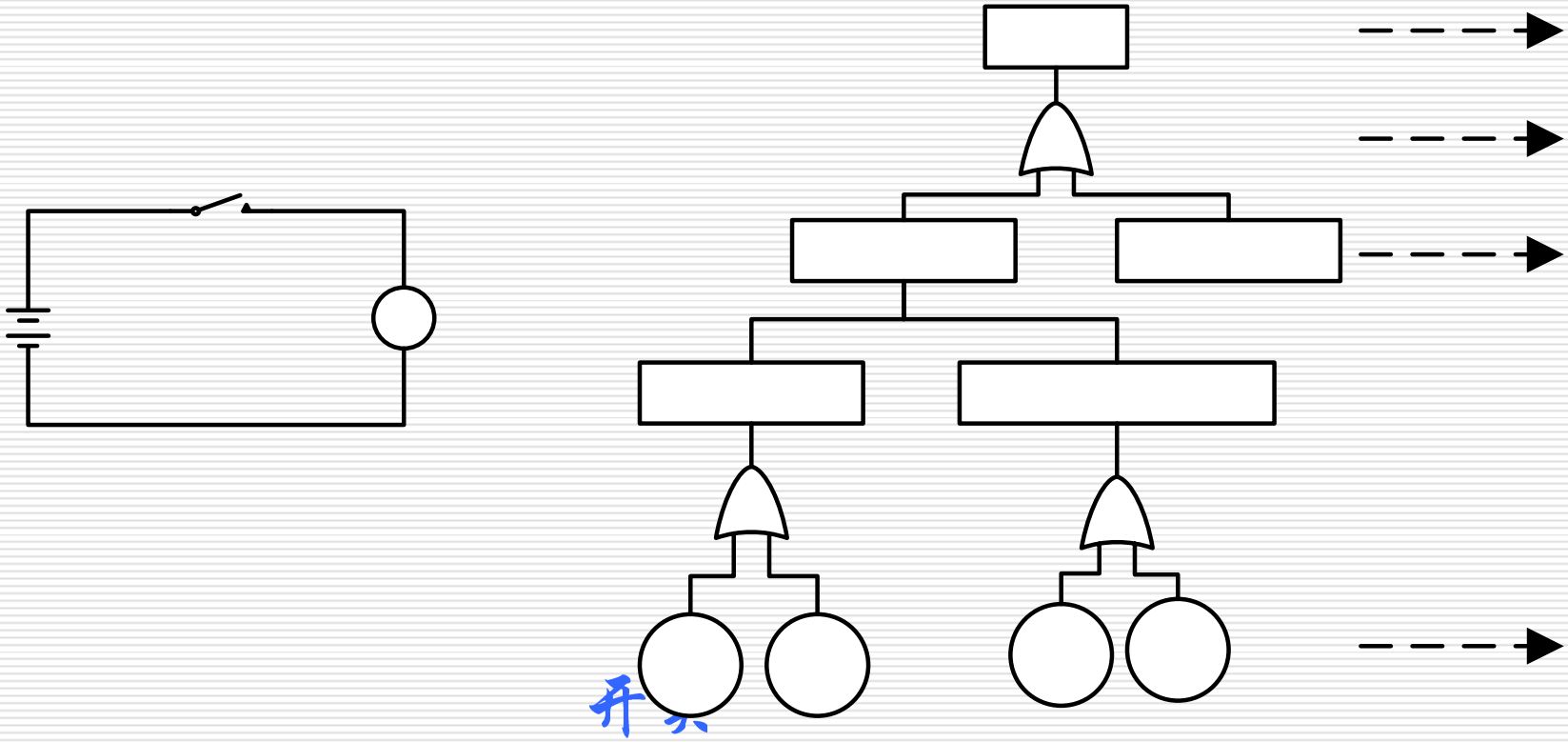
概述

- 切尔诺贝利核泄露事故、美国的挑战者号升空后爆炸和印度的博帕尔化学物质泄露。
- FMECA：单因素分析法，只能分析单个故障模式对系统的影响。
- FTA可分析多种故障因素（硬件、软件、环境、人为因素等）的组合对系统的影响。
- FMECA和FTA是工程中最有效的故障分析方法，FMECA是FTA的基础。
- 各工程领域广泛应用：核工业、航空、航天、机械、电子、兵器、船舶、化工等。

泰坦尼克海难



电机故障树



基本概念

故障树定义

故障树指用以表明产品哪些组成部分的故障或外界事件或它们的组合将导致产品发生一种给定故障的逻辑图。

- 故障树是一种逻辑因果关系图，构图的元素是事件和逻辑门
  - 事件用来描述系统和元、部件故障的状态
  - 逻辑门把事件联系起来，表示事件之间的逻辑关系

基本概念

故障树分析（FTA）

通过对可能造成产品故障的硬件、软件、环境、人为因素进行分析，画出故障树，从而确定产品故障原因的各种可能组合方式和(或)其发生概率。

- 定性分析
- 定量分析

FTA目的

- 目的
  - 帮助判明可能发生的故障模式和原因；
  - 发现可靠性和安全性薄弱环节，采取改进措施，以提高产品可靠性和安全性；
  - 计算故障发生概率；
  - 发生重大故障或事故后，FTA是故障调查的一种有效手段，可以系统而全面地分析事故原因，为故障“归零”提供支持；
  - 指导故障诊断、改进使用和维修方案等。

FTA特点

- 特点
  - 是一种自上而下的图形演绎方法；
  - 有很大的灵活性；
  - 综合性：硬件、软件、环境、人素等；
  - 主要用于安全性分析；

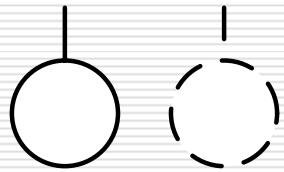
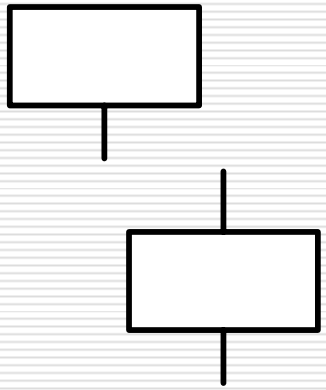
FTA工作要求

- 在产品研制早期就应进行FTA，以便早发现问题并进行改进。随设计工作进展，FTA应不断补充、修改、完善。
- “谁设计，谁分析。”
  - 故障树应由设计人员在FMEA基础上建立。可靠性专业人员协助、指导，并由有关人员审查，以保证故障树逻辑关系的正确性。
- 应与FMEA工作相结合
  - 应通过FMEA找出影响安全及任务成功的关键故障模式（即I、II类严酷度的故障模式）作为顶事件，建立故障树进行多因素分析，找出各种故障模式组合，为改进设计提供依据。

FTA工作要求

- FTA输出的设计改进措施，必须落实到图纸和有关技术文件中
- 应采用计算机辅助进行FTA
  - 由于故障树定性、定量分析工作量十分庞大，因此建立故障树后，应采用计算机辅助进行分析，以提高其精度和效率。

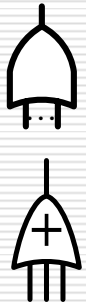
故障树常用事件符号

符号		说明
底事件		元、部件在设计的运行条件下发生的随机故障事件。 ■实线圆——硬件故障 ■虚线圆——人为故障
		未探明事件 表示该事件可能发生，但是概率较小，勿需再进一步分析的故障事件，在故障树定性、定量分析中一般可以忽略不计。
	顶事件	人们不希望发生的显著影响系统技术性能、经济性、可靠性和安全性的故障事件。顶事件可由FMECA分析确定。
	中间事件	故障树中除底事件及顶事件之外的所有事件。

故障树常用事件符号

符号	说明
	开关事件：已经发生或必将要发生的特殊事件。
	条件事件：描述逻辑门起作用的具体限制的特殊事件。
	入三角形：位于故障树的底部，表示树的A部分分支在另外地方。 出三角形：位于故障树的顶部，表示树A是在另外部分绘制的一棵故障树的子树。

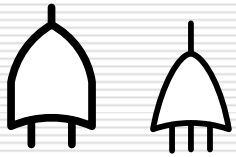
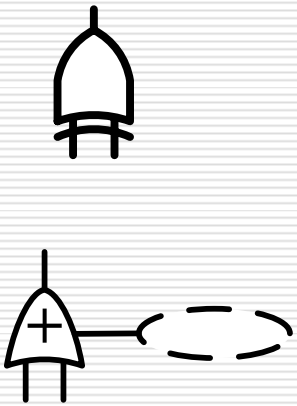
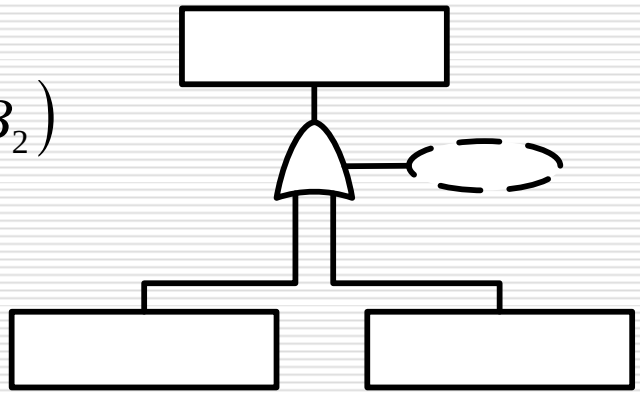
故障树常用逻辑门符号

符号	说明
与门 	□B_i(i=1,2,…,n)为门的输入事件，A为门的输出事件 □B_i同时发生时，A必然发生，这种逻辑关系称为事件交 □用逻辑“与门”描述，逻辑表达式为 $A = B_1 \cap B_2 \cap B_3 \cap \cdots \cap B_n$
或门 	□当输入事件中至少有一个发生时，输出事件A发生，称为事件并 □用逻辑“或门”描述，逻辑表达式为 $A = B_1 \cup B_2 \cup B_3 \cup \cdots \cup B_n$

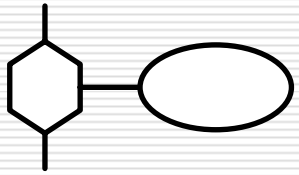
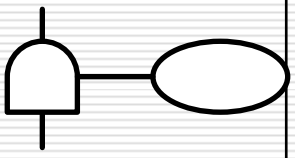
A

B1 Bn

故障树常用逻辑门符号

符号	说明
	表决门：n个输入中至少有r个发生，则输出事件发生；否则输出事件不发生。
	异或门：输入事件B1，B2中任何一个发生都可引起输出事件A发生，但B1，B2不能同时发生。相应的逻辑代数表达式为 $A=(B_1\cap\overline{B}_2)\cup(\overline{B}_1\cap B_2)$ 

故障树常用逻辑门符号

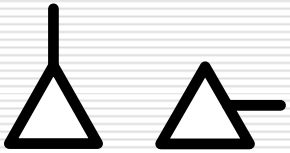
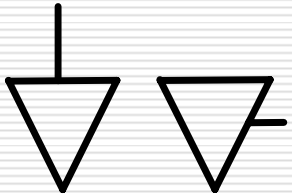
符号	说明
	禁门： ■仅当“禁门打开条件”发生时，输入事件B发生才导致输出事件A发生； ■打开条件写入椭圆框内。
	顺序与门：仅当输入事件B按规定的“顺序条件”发生时，输出事件A才发生。
	非门：输出事件A是输入事件B的逆事件。

A

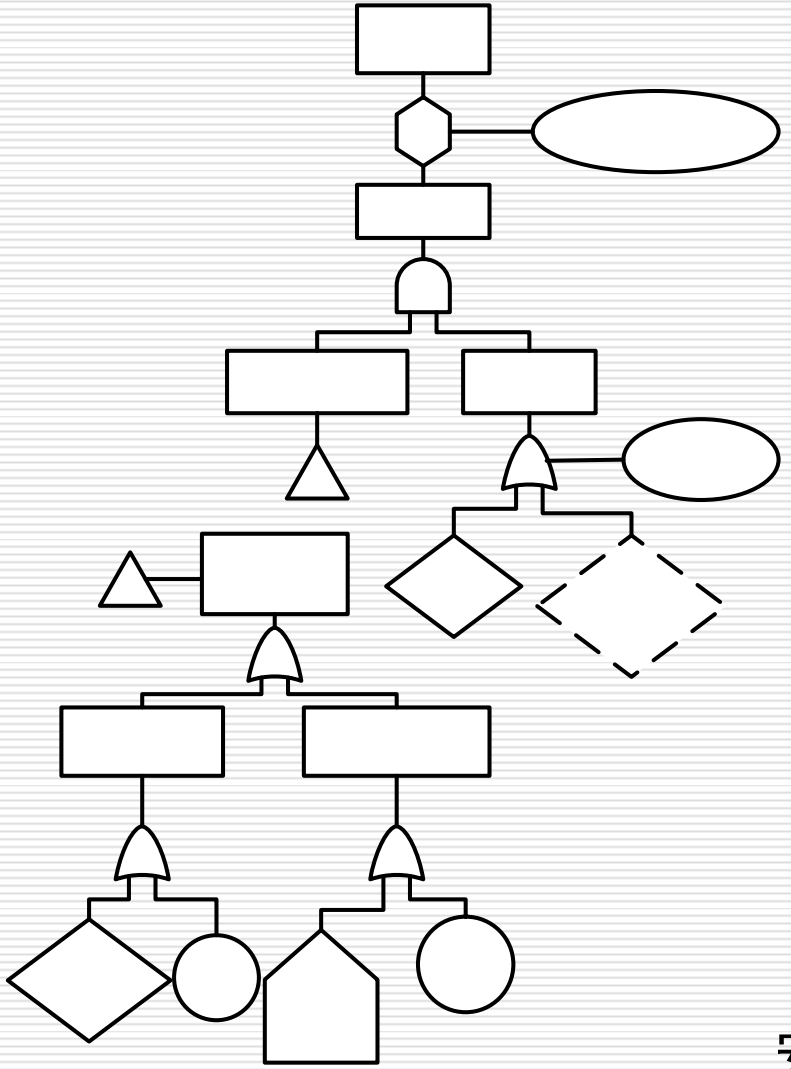
禁门打开条件

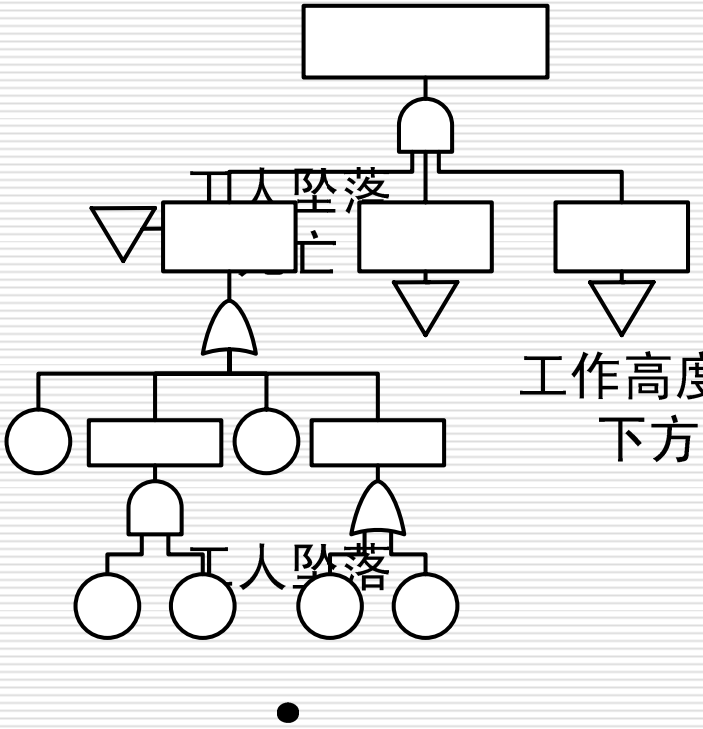
B

故障树常用逻辑门符号

符号	说明
	相同转移符号（A是子树代号，用字母数字表示）： ■左图表示“下面转到以字母数字为代号所指的地方去” ■右图表示“由具有相同字母数字的符号处转移到这里来”
	相似转移符号（A同上）： ■左图表示“下面转到以字母数字为代号所指结构相似而事件标号不同的子树去”，不同事件标号在三角形旁注明 ■右图表示“相似转移符号所指子树与此处子树相似但事件标号不同” A　　A

故障树示例





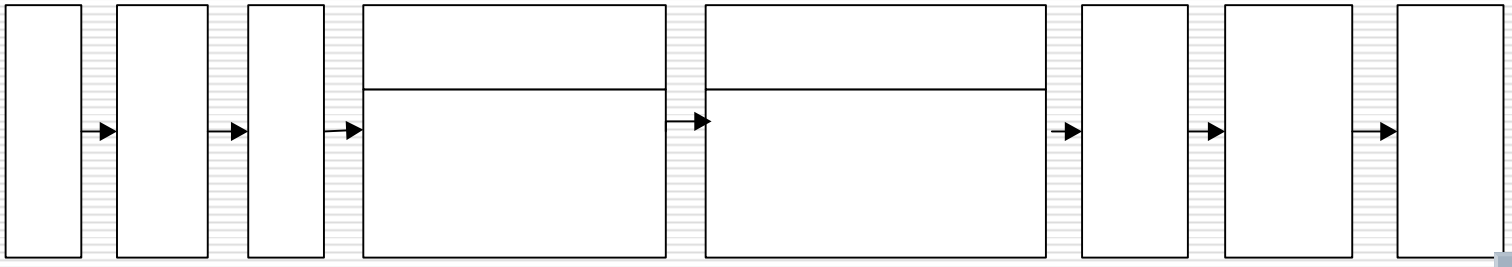
安全带设施
不起作用

工人失足
坠落

身体重心在

故障树分析

- 建树步骤
  - 广泛收集并分析系统及其故障的有关资料；
  - 选择顶事件；
  - 建造故障树；
  - 简化故障树。
- 分析步骤
  - 建立故障树；
  - 故障树定性分析
  - 故障树定量分析
  - 重要度分析
  - 分析结论：薄弱环节
  - 确定改进措施



故障树定性分析

- 目的
  - 寻找顶事件的原因事件及原因事件的组合（最小割集）
  - 发现潜在的故障
  - 发现设计的薄弱环节，以便改进设计
  - 指导故障诊断，改进使用和维修方案
- 割集、最小割集概念
  - 割集：故障树中一些底事件的集合，当这些底事件同时发生时，顶事件必然发生；
  - 最小割集：若将割集中所含的底事件任意去掉一个就不再成为割集了，这样的割集就是最小割集。

最小割集的意义

- 最小割集对降低复杂系统潜在事故风险具有重大意义
  - 如果能使每个最小割集中至少有一个底事件恒不发生(发生概率极低)，则顶事件就恒不发生(发生概率极低)，系统潜在事故的发生概率降至最低
- 消除可靠性关键系统中的一阶最小割集，可消除单点故障
  - 可靠性关键系统不允许有单点故障，方法之一就是设计时进行故障树分析，找出一阶最小割集，在其所在的层次或更高的层次增加“与门”，并使“与门”尽可能接近顶事件。

最小割集的意义

- 最小割集可以指导系统的故障诊断和维修
  - 如果系统某一故障模式发生了，则一定是该系统中与其对应的某一个最小割集中的全部底事件全部发生了。进行维修时，如果只修复某个故障部件，虽然能够使系统恢复功能，但其可靠性水平还远未恢复。根据最小割集的概念，只有修复同一最小割集中的所有部件故障，才能恢复系统可靠性、安全性设计水平。

故障树定性分析

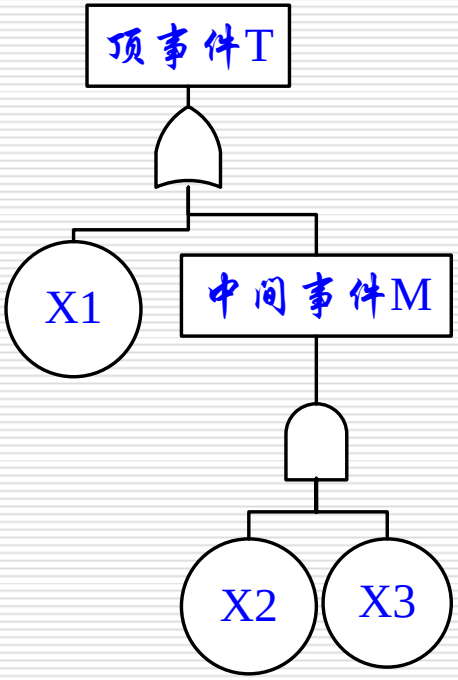
- 示例
  - 根据与、或门的性质和割集的定义，可方便找出该故障树的割集是：

{X1},{X2,X3},{X1,X2,X3},{X2,X1},{X1,X3}

  - 根据与、或门的性质和割集的定义，可方便找出该故障树的最小割集是：

{X1},{X2,X3}

- 最小割集求解方法
  - 常用的有下行法与上行法两种



下行法求解最小割集

步骤	1	2	3	4	5	6
过程	x_1	x_1	x_1	x_1	x_1	x_1
	M_1	M_2	M_4, M_5	M_4, M_5	x_4, M_5	x_4, x_6
	x_2	M_3	M_3	x_3	x_5, M_5	x_4, x_7
		x_2	x_2	M_6	X_3	x_5, x_6
				x_2	M_6	x_5, x_7
					x_2	x_3
						x_6
						x_8
						x_2

最小割集比较

- 根据最小割集含底事件数目(阶数)排序，在各个底事件发生概率比较小，且相互差别不大的条件下，可按以下原则对最小割集进行比较：
  - 阶数越小的最小割集越重要
  - 在低阶最小割集中出现的底事件比高阶最小割集中的底事件重要
  - 在最小割集阶数相同的条件下，在不同最小割集中重复出现的次数越多的底事件越重要

故障树定量分析

- 假设
  - 独立性：底事件之间相互独立；
  - 两态性：元、部件和系统只有正常和故障两种状态
  - 指数分布：元、部件和系统寿命
- 故障树的数学描述
  - 结构函数
  - 典型逻辑门的结构函数
  - 结构函数示例
  - 单调关联系统
- 典型逻辑门的概率计算
- 顶事件发生概率计算

故障树结构函数

故障树的数学描述

$$x_i=\begin{cases}1 & \text{底事件}x_i\text{发生（即元、部件故障）}\\ 0 & \text{底事件}x_i\text{不发生（即元、部件正常）}\end{cases}$$

$$\Phi=\begin{cases}1 & \text{顶事件发生（即系统故障）}\\ 0 & \text{顶事件不发生（即系统正常）}\end{cases}$$

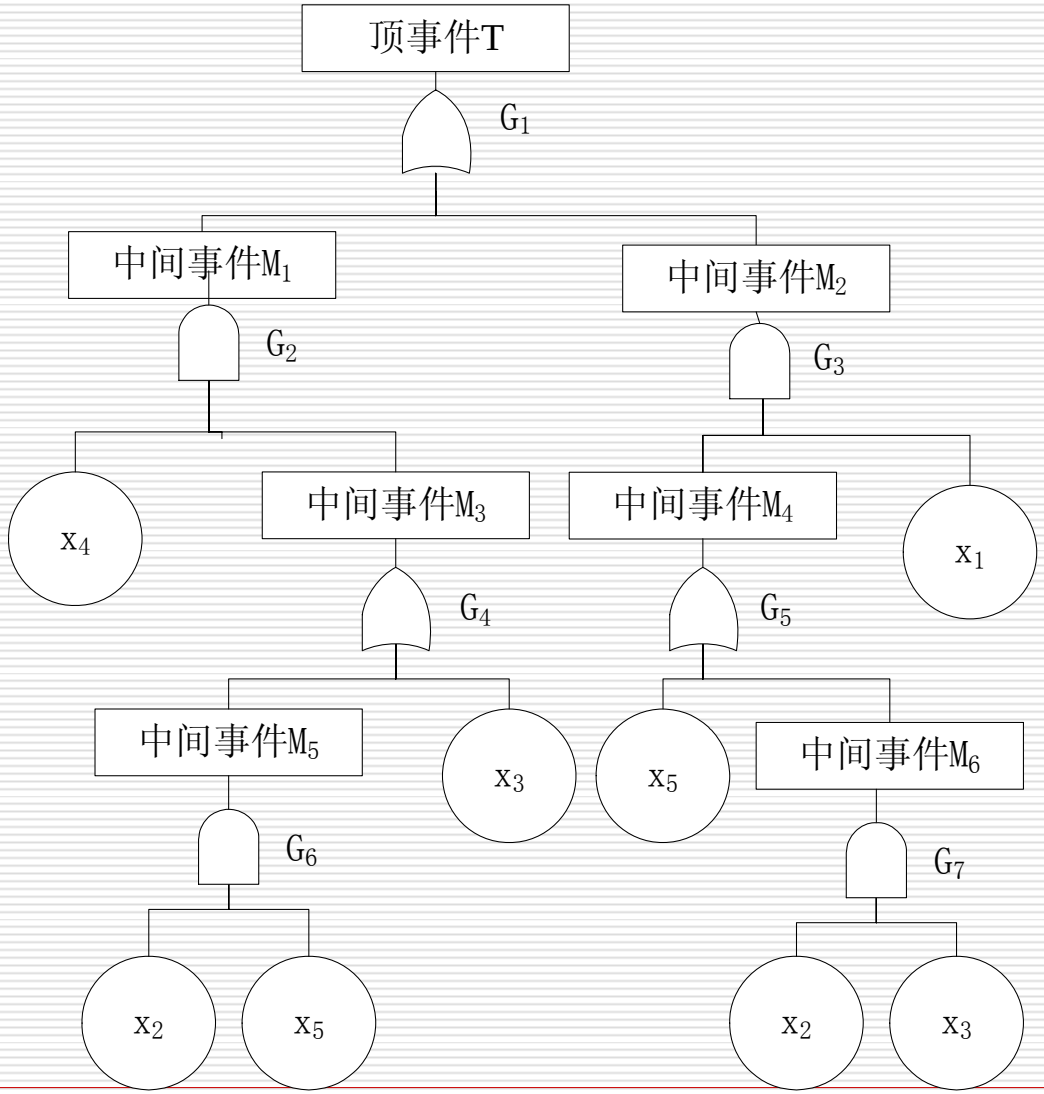
故障树结构函数——表示系统状态布尔函数：

$$\Phi=\Phi\left(\vec{X}\right),\qquad \vec{X}=\left(x_1,x_2,\cdots x_n\right)$$

典型逻辑门的结构函数

序号	名称	描述
1	与门	$\Phi(\vec{X})=\prod_{i=1}^{n} x_i$
2	或门	$\Phi(\vec{X})=1-\prod_{i=1}^{n}(1-x_i)$
3	n中取r	$\Phi(\vec{X})=\begin{cases}1 & \text{当}\sum x_i \geq r\text{时}\\ 0 & \text{其它情况}\end{cases}$
4	异或门	$\Phi(\vec{X})=1-[1-x_1\cap(1-x_2)]\cap[1-(1-x_1)\cap x_2]$ $=(\bar{x}_1\cap x_2)\cup(\bar{x}_2\cap x_1)$

结构函数示例



结构函数示例

$$\Phi(\vec{X})=\{x_4\cap[x_3\cup(x_2\cap x_5)]\}\cup\{x_1\cap[x_5\cup(x_3\cap x_2)]\}$$

对于复杂系统来说，其结构函数是相当冗长繁杂的，可根据逻辑运算规则或最小割集的概念，对结构函数进行改写，以利于故障树的定性分析和定量计算。用逻辑运算分配律：

$$x_4\cap[x_3\cup(x_2\cap x_5)]=(x_3\cap x_4)\cup(x_2\cap x_5\cap x_4)$$

$$x_1\cap[x_5\cup(x_3\cap x_2)]=(x_1\cap x_5)\cup(x_1\cap x_3\cap x_2)$$

所以 $\Phi(\vec{X})=(x_1\cap x_5)\cup(x_3\cap x_4)\cup(x_2\cap x_4\cap x_5)\cup(x_1\cap x_2\cap x_3)$

同样用下行法可求得最小割集为

$\{X_1,X_5\},\{X_3,X_4\},\{X_2,X_4,X_5\},\{X_1,X_2,X_3\}$

根据以上最小割集，其结构函数可写成

$$\Phi(\vec{X})=(x_1\cap x_5)\cup(x_3\cap x_4)\cup(x_2\cap x_4\cap x_5)\cup(x_1\cap x_2\cap x_3)$$

单调关联系统

定义

- 指系统中任一组成单元的状态由正常（故障）转为故障（正常），不会使系统的状态由故障（正常）转为正常（故障）的系统。

性质

- 系统中的每一个元、部件对系统可靠性都有一定影响，只是影响程度不同。
- 系统中所有元、部件故障（正常），系统一定故障（正常）。
- 系统中故障元、部件的修复不会使系统由正常转为故障；正常元、部件故障不会使系统由故障转为正常。
- 单调关联系统的可靠性不会比由相同元、部件构成的串联系统坏，也不会比由相同元、部件构成的并联系统好。

典型逻辑门的概率计算

序号	名称	描述
1	与门	$F_s(t)=E\left[\Phi\left(\vec{X}\right)\right]=E\left[\prod_{i=1}^{n}x_i(t)\right]=F_1(t)\cdot F_2(t)\cdots F_n(t)$
2	或门	$F_s(t)=E\left[\Phi\left(\vec{X}\right)\right]=E\left[1-\prod_{i=1}^{n}(1-x_i)\right]=1-[1-F_1(t)][1-F_2(t)]\cdots[1-F_n(t)]$
3	n中取r	当 n 个 输 入 事 件 为 同 类 事 件 时 ： $F_s(t)=E\left[\Phi\left(\vec{X}\right)\right]=\sum_{m=r}^{n}C_n^m F_i^m(t)[1-F_i(t)]^{n-m}$
4	异或门	$F_s(t)=E\left[\Phi\left(\vec{X}\right)\right]=\{1-[1-E(x_1)E(1-x_2)]\}\{1-[1-E(x_2)E(1-x_1)]\}$ $=[1-(1-(1-R_1)R_2)][1-(1-(1-R_2)R_1)]$

顶事件概率计算

- 最小割集之间不相交
- 最小割集之间相交
  - 全概率法
  - 直接化法
  - 递推化法
  - 近似算法
- 示例1
- 示例2

最小割集之间不相交

已知故障树的全部最小割集为 K_1，K_2，…，K_{Nk}，并且假定各最小割集中没有重复出现的底事件，也就是假定最小割集之间是不相交的。则有

$$T=\Phi(\vec{X})=\bigcup_{j=1}^{N_k}K_j(t)$$

$$P[K_j(t)]=\prod_{i\in K_j}F_i(t)$$

式中：$P[K_j(t)]$ —— 在时刻 t 第 j 个最小割集发生的概率；

$F_i(t)$ —— 在时刻 t 第 j 个最小割集中第 i 个部件的故障概率；

N_k —— 最小割集数。

则

$$P(T)=F_s(T)=P[\Phi(\vec{X})]=1-\prod_{j=1}^{N_k}\left(1-\prod_{i\in K_j}F_i(t)\right)$$

全概率法

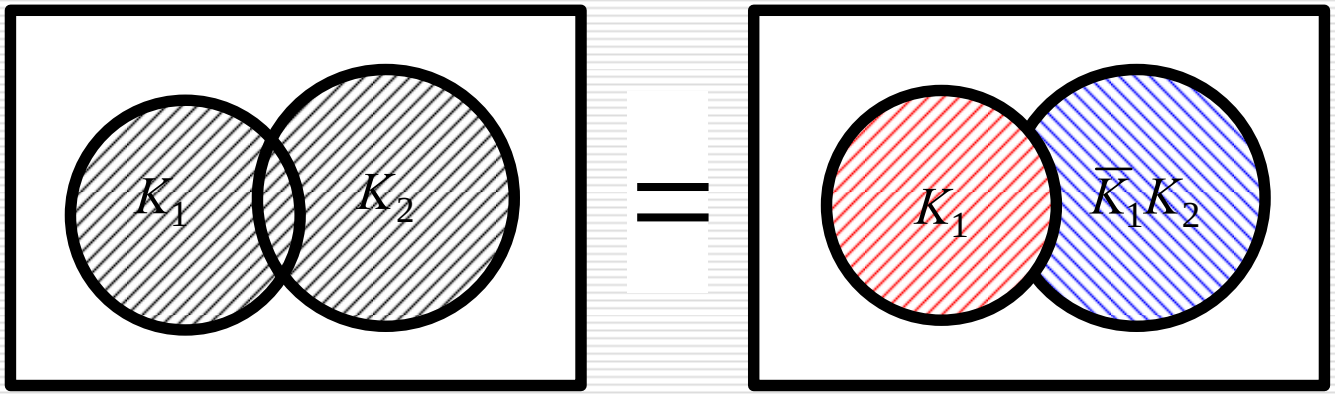
$$P(T)=P\left(K_1\cup K_2\cup\cdots\cup K_{N_i}\right)$$

$$=\sum_{i=1}^{N_i}P(K_i)-\sum_{i<j=2}^{N_k}P(K_iK_j)+\sum_{i<j<k=3}^{N_k}P(K_iK_jK_k)+\cdots$$

$$+(-1)^{N_k-1}P\left(K_1,K_2,.\cdots K_{N_k}\right)$$

直接化法

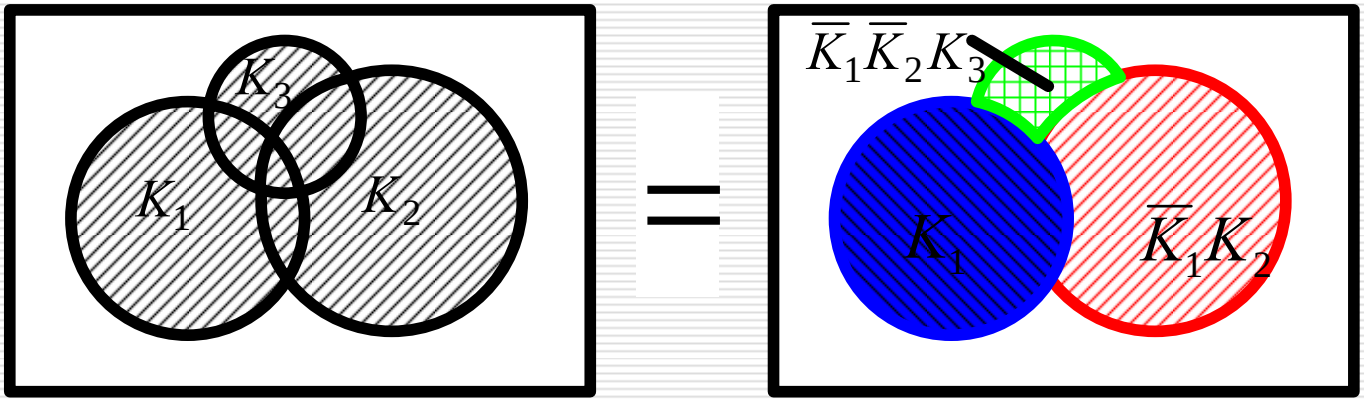
$$K_1 \cup K_2 = K_1 + \overline{K}_1 K_2$$



$K_1 \cup K_2$ $\qquad$ $K_1 + \overline{K}_1 K_2$

递推化法

$$K_1 \cup K_2 \cup K_3 = K_1 + \overline{K}_1 K_2 + \overline{K}_1 \overline{K}_2 K_3$$



$K_1 \cup K_2 \cup K_3$ $\qquad$ $K_1 + \overline{K}_1 K_2 + \overline{K}_1 \overline{K}_2 K_3$

近似算法

一阶近似：
$$P(T)\approx S_1=\sum_{i=1}^{N_k}P(K_i)$$

二阶近似：
$$P(T)\approx S_1-S_2=\sum_{i=1}^{N_k}P(K_i)-\sum_{i<j=2}^{N_k}P(K_iK_j)$$

$$P(K_i)=\prod P(x_k),\qquad x_k\in K_i$$

近似算法计算示例

最小割集：$\{x_1\}$，$\{x_4, x_7\}$，$\{x_5, x_7\}$，$\{x_3\}$，$\{x_6\}$，$\{x_8\}$

故障树

一阶近似算法：

$$P(T) \approx \sum_{i=1}^{N_k} P(K_i)$$

$$= P(x_1) + P(x_4) \bullet P(x_7) + P(x_5) \bullet P(x_7) + P(x_6) + P(x_3) + P(x_8)$$

顶事件概率计算示例

二阶近似算法：

$$P(T)\approx\sum_{i=1}^{N_k}P(K_i)-\sum_{i<j=2}^{N_k}P(K_iK_j)=P(x_1)+P(x_4)\bullet P(x_7)+P(x_5)\bullet P(x_7)$$

$$+P(x_6)+P(x_3)+P(x_8)-P(x_1)\bullet P(x_4)\bullet P(x_7)$$

$$-P(x_1)\bullet P(x_5)\bullet P(x_7)-P(x_1)\bullet P(x_6)-P(x_1)\bullet P(x_3)$$

$$-P(x_1)\bullet P(x_8)-P(x_4)\bullet P(x_5)\bullet P(x_7)-P(x_4)\bullet P(x_6)\bullet P(x_7)$$

$$-P(x_3)\bullet P(x_4)\bullet P(x_7)-P(x_4)\bullet P(x_7)\bullet P(x_8)$$

$$-P(x_5)\bullet P(x_6)\bullet P(x_7)-P(x_3)\bullet P(x_5)\bullet P(x_7)$$

$$-P(x_5)\bullet P(x_7)\bullet P(x_8)-P(x_3)\bullet P(x_6)-P(x_6)\bullet P(x_8)$$

$$-P(x_3)\bullet P(x_8)$$

示例2

[示例 2]故障树如图所示，其中 $F_A=F_B=0.2,F_C=F_D=0.3,F_E=0.36$。

该故障树的最小割集为：

$K_1=\{A,C\},K_2=\{B,D\},K_3=\{A,D,E\},K_4=\{B,C,E\}$，求顶事件发生概率。

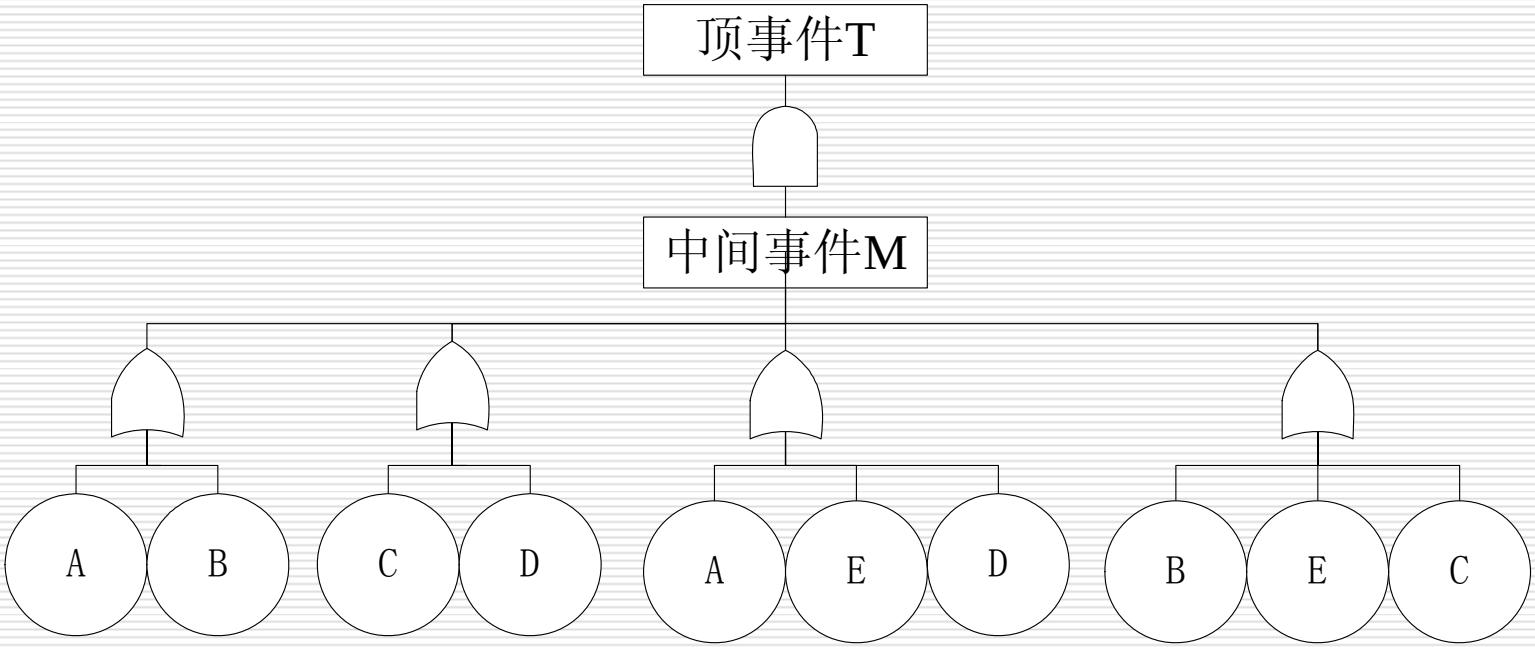


图7-15

示例2

□ 直接化法

$$
\begin{aligned}
T &= K_1 \cup K_2 \cup K_3 \cup K_4 \\
&= K_1 + \overline{K}_1 (K_2 \cup K_3 \cup K_4) \\
&= AC + \overline{AC}(BD \cup ADE \cup BCE) \\
&= AC + (\overline{A} \cup \overline{C})(BD \cup ADE \cup BCE) \\
&= AC + \overline{A}BD + \overline{\overline{A}BD}(\overline{A}BCE \cup \overline{C}BD \cup \overline{C}ADE) \\
&= \cdots\cdots \\
&= AC + \overline{A}BD + A\overline{C}BD + \overline{DA}BCE + \overline{B}\,\overline{C}ADE
\end{aligned}
$$

$$
\begin{aligned}
P(T) &= P(A)P(C) + P(\overline{A})P(B)P(D) + P(A)P(\overline{C})P(B)P(D) + \\
&\quad + P(\overline{D})P(\overline{A})P(B)P(C)P(E) + P(\overline{B})P(\overline{C})P(A)P(D)P(E) \\
&= 0.2 \times 0.3 + 0.8 \times 0.2 \times 0.3 + 0.2 \times 0.7 \times 0.2 \times 0.3 + \\
&\quad + 0.7 \times 0.8 \times 0.2 \times 0.3 \times 0.36 + 0.8 \times 0.7 \times 0.2 \times 0.3 \times 0.36 \\
&= 0.140592
\end{aligned}
$$

示例2

递推化法

$$
\begin{aligned}
T &= K_1 \cup K_2 \cup K_3 \cup K_4 \\
&= K_1 + \overline{K}_1 K_2 + \overline{K}_1 \overline{K}_2 K_3 + \overline{K}_1 \overline{K}_2 \overline{K}_3 K_4 \\
&= \cdots\cdots \\
&= AC + \overline{A}BD + AB\overline{C}D + \overline{C}\,\overline{B}AED + \overline{A}\,\overline{D}BCE
\end{aligned}
$$

重要度分析

- 重要度的概念
  - 定义
    - 底事件或最小割集对顶事件发生的贡献
  - 目的
    - 确定薄弱环节和改进设计方案
  - 重要度分类
    - 概率重要度
    - 结构重要度

概率重要度

概率重要度概念

- 第i个部件不可靠度的变化引起系统不可靠度变化的程度。用数学公式表达为

$$\Delta g_i(t)=\frac{\partial g\left[\vec{F}(t)\right]}{\partial F_i(t)}=\frac{\partial F_s(t)}{\partial F_i(t)}$$

$\Delta g_i(t)$ —— 概率重要度；

$F_i(t)$ —— 元、部件不可靠度；

$g\left[\vec{F}(t)\right]$ —— 顶事件发生概率，

$F_s(t)$ —— 系统不可靠度；

$\vec{F}(t)=\left[F_1(t),F_2(t),\cdots,F_n(t)\right]$；

$F_s(t)=P(T)=g\left[\vec{F}(t)\right]$

概率重要度

概率重要度示例

已知：λ1=0.001/h, λ2=0.002/h, λ3=0.003/h,。试求当t=100h时各部件的概率重要度、结构重要度和关键重要度。

解

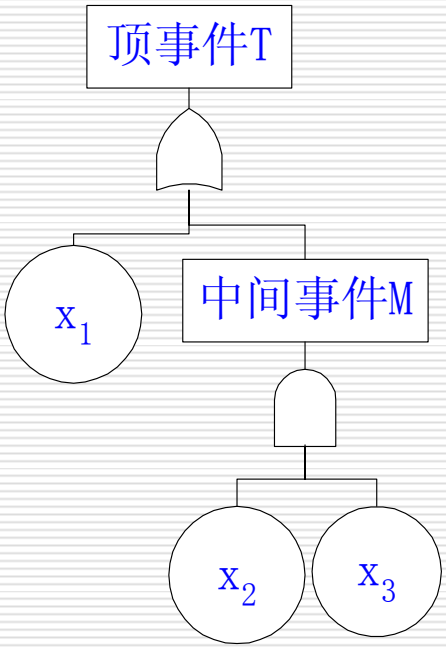
$$\because F_S(t)=1-[1-F_1(t)][1-F_2(t)\bullet F_3(t)]$$

$$\therefore \Delta g_1(t)=\partial F_S(t)\Big/\partial F_1(t)=1-F_2(t)\bullet F_3(t)$$

$$\Delta g_1(100)=1-(1-e^{-0.002\times 100})(1-e^{-0.003\times 100})=0.953$$

$$\Delta g_2(100)=[1-F_1(100)]\bullet F_3(100)=0.2345$$

$$\Delta g_3(100)=[1-F_1(100)]\bullet F_2(100)=0.164$$



结构重要度

结构重要度概念

- 元、部件在系统中所处位置的重要程度，与元、部件本身故障概率毫无关系。其数学表达式为

$$I_i^{\phi} = \frac{1}{2^{n-1}} n_i^{\phi}$$

$$n_i^{\phi} = \sum_{2^{n-1}} \left[\Phi\left(1_i, \vec{X}\right) - \Phi\left(0_i, \vec{X}\right) \right]$$

- I_i^{ϕ} ——第i个元、部件的结构重要度；
- n ——系统所含元、部件的数量；

- 两种状态

$$\Phi\left(0_i, \vec{X}\right) = 0 \rightarrow \Phi\left(1_i, \vec{X}\right) = 1, \Phi\left(1_i, \vec{X}\right) - \Phi\left(0_i, \vec{X}\right) = 1$$

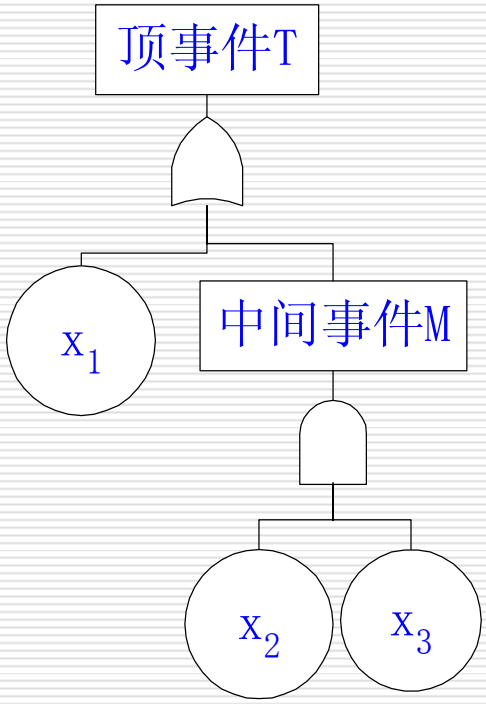
$$\Phi\left(0_i, \vec{X}\right) = 0 \rightarrow \Phi\left(1_i, \vec{X}\right) = 0, \Phi\left(1_i, \vec{X}\right) - \Phi\left(0_i, \vec{X}\right) = 0$$

结构重要度

结构重要度示例

求解如图所示故障树中的底事件结构重要度

解：二个部件，共有 $2^{3-1}=4$ 种状态：



$$\Phi(0,0,0)=0,\Phi(0,0,1)=0,\Phi(0,1,1)=1,\Phi(0,1,0)=0$$

$$\Phi(1,0,0)=1,\Phi(1,0,1)=1,\Phi(1,1,0)=1,\Phi(1,1,1)=1$$

$$n_1^{\phi}=[\Phi(1,0,0)-\Phi(0,0,0)]+[\Phi(1,0,1)-\Phi(0,0,1)]$$

$$+[\Phi(1,1,0)-\Phi(0,1,0)]+[\Phi(1,1,1)-\Phi(0,1,1)]=3$$

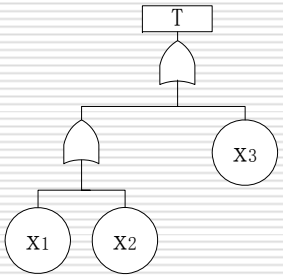
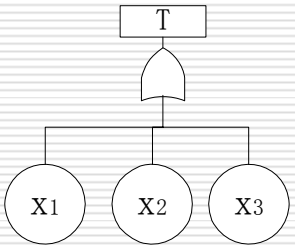
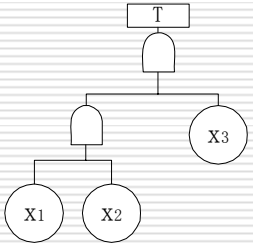
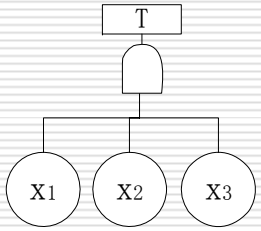
$$n_2^{\phi}=[\Phi(0,1,1)-\Phi(0,0,1)]=1$$

$$n_3^{\phi}=[\Phi(0,1,1)-\Phi(0,1,0)]=1$$

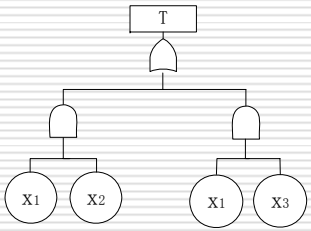
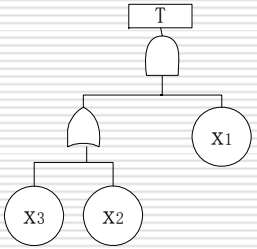
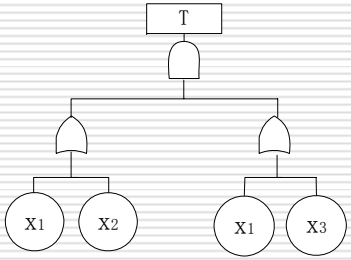
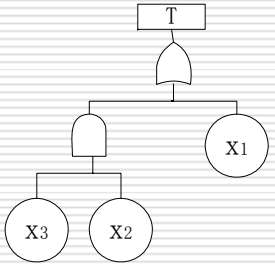
$$I_1^{\phi}=\frac{3}{4},I_2^{\phi}=\frac{1}{4},I_3^{\phi}=\frac{1}{4}$$

故障树的简化

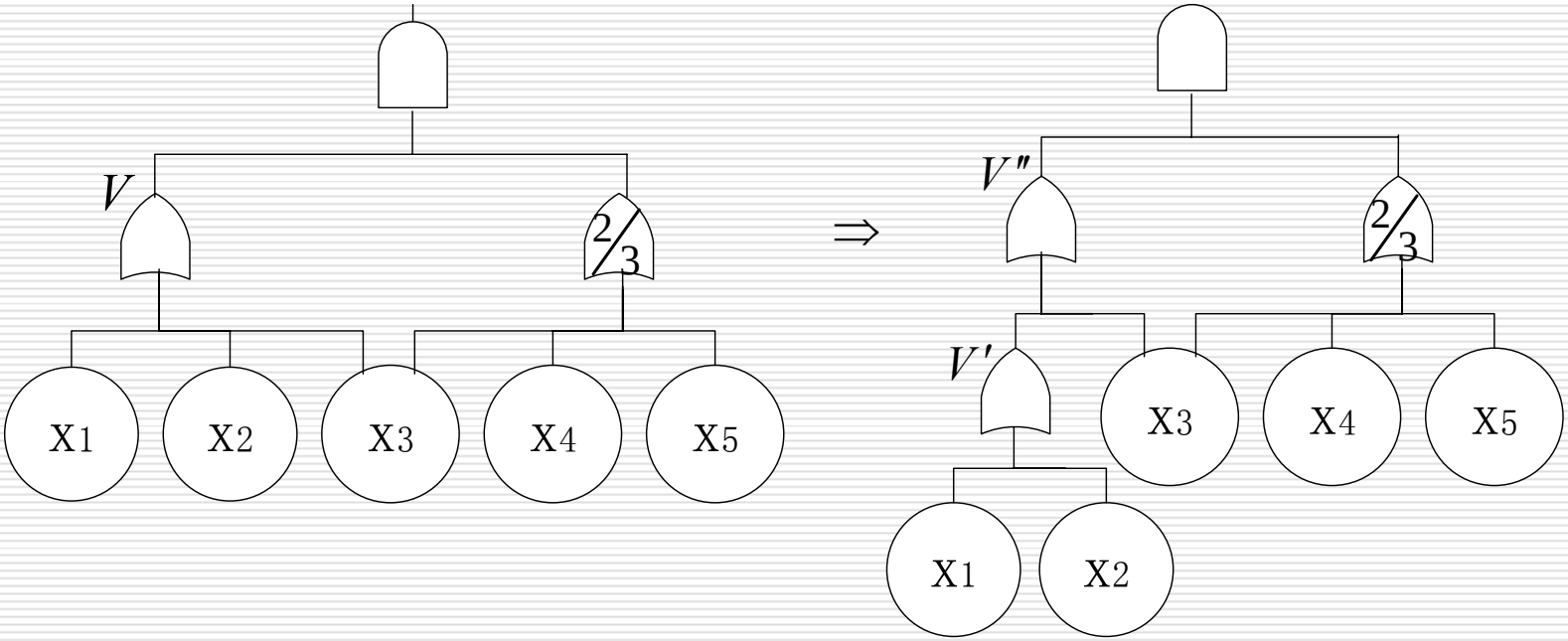
故障树的逻辑简化

简化原理	原故障树	简化故障树
结合律Ⅰ $(x_1 \cup x_2) \cup x_3$ $= x_1 \cup x_2 \cup x_3$		
结合律Ⅱ $(x_1 \cap x_2) \cap x_3$ $= x_1 \cap x_2 \cap x_3$		

故障树的简化

简化原理	原故障树	简化故障树
分配律 I $(x_1 \cap x_2) \cup (x_1 \cap x_3)$ $= x_1 \cap (x_2 \cup x_3)$		
分配律 II $(x_1 \cup x_2) \cap (x_1 \cup x_3)$ $= x_1 \cup (x_2 \cap x_3)$		

故障树的模块分解



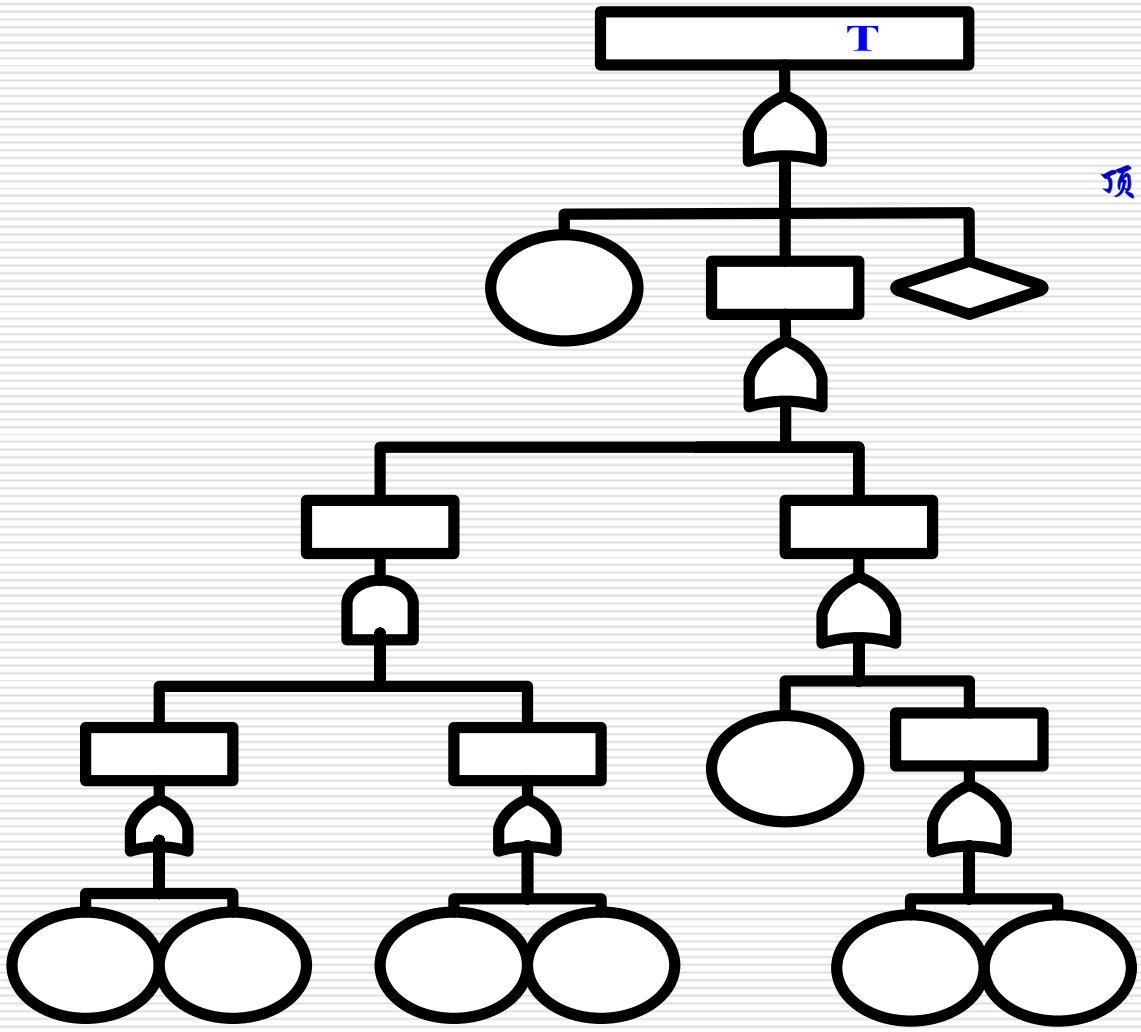
割顶点法示例

故障树的早期不变化

- 当重复事件多时，无法应用模块分解法。早期不变化可有效地消除重复事件。规则是遇到与门，其输入、输出均不变；遇到或门，对输入不变化。

$$
\begin{aligned}
G_i &= X_{i_1} \cup X_{i_2} \cup X_{i_k} \cup \cdots\cdots \cup G_{i_{k+1}} \cup \cdots\cdots \cup G_{i_n} \\
&= X_{i_1} + \overline{X}_{i_1} X_{i_2} + \overline{X}_{i_1} \overline{X}_{i_2} \cdots\cdots \overline{X}_{i_{k-1}} X_{i_k} + \overline{X}_{i_1} \cdots\cdots \overline{X}_{i_k} G_{i_{k+1}} \\
&\quad + \cdots\cdots + \overline{X}_{i_1} \cdots\cdots \overline{X}_{i_k} \overline{G}_{i_{k+1}} \cdots\cdots \overline{G}_{i_{n+1}} G_{i_n}
\end{aligned}
$$

故障树示例



求解最小割集

顶事件概率计算算例

顶事

x_1

M_1

故障树示例