GJB

中华人民共和国国家军用标准

FL 0111

GJB/Z 768A－98

故障树分析指南

Guide to fault tree analysis

1998－03－16 发布　　1998－09－01 实施

国防科学技术工业委员会　批准

前　言

故障树分析(FTA)是产品(系统) 可靠性和安全性分析的工具之一,用来寻找导致不希望的系统故障或灾难性危险事件(顶事件)发生的所有原因和原因组合,在具有基础数据时求出顶事件发生的概率及其他定量指标。FTA 也是分析已经发生的事故的一种基本方法。在 GJB 450《装备研制与生产的可靠性通用大纲》的“可靠性定性要求”一节提出了应用 FTA 方法的任务。

1989 年发布了 GJB 768.1《建造故障树的基本规则和方法》、GJB 768.2《故障树表述》、GJB 768.3《正规故障树定性分析》。本指导性技术文件是对上述标准的修订,补充了单调故障树定量分析的内容和考虑产品部件及系统多状态故障的参考分析方法,并对故障树术语和符号加以统一整理和补充,使之完整配套。

1990 年国际电工委员会发布的标准 IEC 1025(第一版)《故障树分析》的特点是比较概括,但为本指导性技术文件提供了参考。在此基础上本指导性技术文件的技术内容力求更为具体、严格和统一,以适应各类军事装备应用的需要。

本指导性技术文件对故障树术语和符号加以统一整理,基本上仍按习用的故障树符号,略加校订。而将 IEC 1025 推荐的故障树符号列为本指导性技术文件附录 C(参考件)。

目　　次

中华人民共和国国家军用标准

故障树分析指南

Guide to fault tree analysis

GJB/Z 768A－98
代替GJB 768.1－89
GJB 768.2－89
GJB 768.3－89

1 范围

1.1 主题内容

本指导性技术文件规定了产品(系统)故障树分析的一般程序和方法。

1.2 适用范围

本指导性技术文件适用于在产品的研制、生产、使用阶段进行故障树建造和对单调故障树进行定性、定量分析。故障树的各种故障事件可包括硬件故障、软件故障、人的失误、环境影响等各种故障因素,以及能导致人员伤亡、职业病、设备损坏或财产损失、环境严重污染等事故的各种危险因素。

1.3 应用指南

故障树分析是系统安全性和可靠性分析的工具之一。在产品设计阶段,故障树分析可帮助判明潜在的系统故障模式和灾难性危险因素,发现可靠性和安全性薄弱环节,以便改进设计。在生产、使用阶段,故障树分析可帮助故障诊断,改进使用维修方案。故障树分析也是事故调查的一种有效手段。

根据需要和所掌握的数据条件,可对本指导性技术文件进行剪裁,只进行故障树建造和定性分析。

2 引用文件

GJB 368A－94	装备维修性通用大纲
GJB 450－88	装备研制与生产的可靠性通用大纲
GJB 451－90	可靠性维修性术语
GJB 900－90	系统安全性通用大纲

3 定义

3.1 事件及其符号

在故障树分析中各种故障状态或不正常情况皆称故障事件,各种完好状态或正常情况皆称成功事件。两者均可简称为事件。

3.1.1 底事件 bottom event

底事件是故障树中仅导致其它事件的原因事件。它位于所讨论的故障树底端,总是某个逻辑门的输入事件而不是输出事件。

国防科学技术工业委员会 1998－03－16 发布　　1998－09－01 实施

底事件分为基本事件与未探明事件。

3.1.1.1 基本事件 basic event

基本事件是在特定的故障树分析中无须探明其发生原因的底事件。其图形符号见图 3.1。

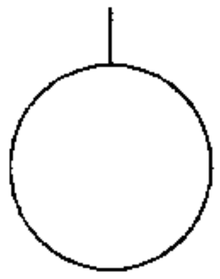


图 3.1 基本事件符号

3.1.1.2 未探明事件 undeveloped event

未探明事件是原则上应进一步探明其原因但暂时不必或者不能探明其原因的底事件，其图形符号见图 3.2。

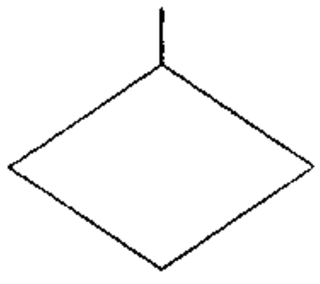


图 3.2 未探明事件符号

3.1.2 结果事件 resultant event

结果事件是故障树分析中由其他事件或事件组合所导致的事件。它位于某个逻辑门的输出端。结果事件分为顶事件与中间事件。其图形符号见图 3.3。

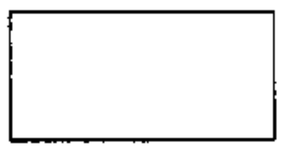


图 3.3 结果事件符号

3.1.2.1 顶事件 top event

顶事件是故障树分析中所关心的最后结果事件。它位于故障树的顶端，总是所讨论故障树中逻辑门的输出事件而不是输入事件。

3.1.2.2 中间事件 intermediate event

中间事件是位于底事件和顶事件之间的结果事件。它既是某个逻辑门的输出事件，同时又是别的逻辑门的输入事件。

3.1.3 特殊事件 special event

特殊事件指在故障树分析中需用特殊符号表明其特殊性或引起注意的事件。

3.1.3.1 开关事件 switch event

已经发生或者必将要发生的特殊事件，其图形符号见图 3.4，示例见图 3.16。

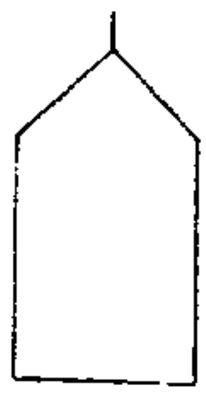


图 3.4　开关事件符号

3.1.3.2　条件事件　conditional event

条件事件是描述逻辑门起作用的具体限制的特殊事件，其图形符号见图 3.5。

图 3.5　条件事件符号

为区分人的失误事件和其它故障事件，可采用虚线表示的失误事件。示例见图 3.16。

3.2　逻辑门及其符号

在故障树分析中逻辑门只描述事件间的因果关系。与门、或门和非门是三个基本门，其他的逻辑门为特殊门。

3.2.1　与门　AND gate

与门表示仅当所有输入事件发生时，输出事件才发生。其图形符号见图 3.6。

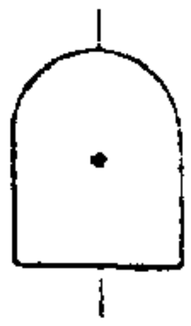


图 3.6　与门符号

3.2.2　或门　OR gate

或门表示至少一个输入事件发生时，输出事件就发生。其图形符号见图 3.7。

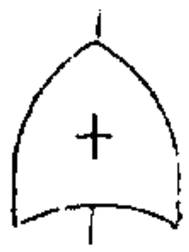


图 3.7　或门符号

3.2.3　非门　NOT gate

非门表示输出事件是输入事件的逆事件。其图形符号见图 3.8。

图 3.8　非门符号

3.2.4　顺序与门　sequential AND gate

顺序与门表示仅当输入事件按规定的顺序发生时，输出事件才发生。其图形符号如图 3.9，示例见图 3.10。

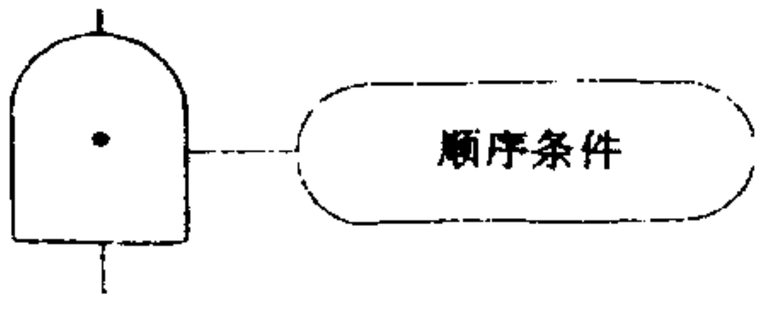


图 3.9　顺序与门符号

顺序与门示例：有主发电机和备份发电机(带开关控制器)的系统停电故障分析。

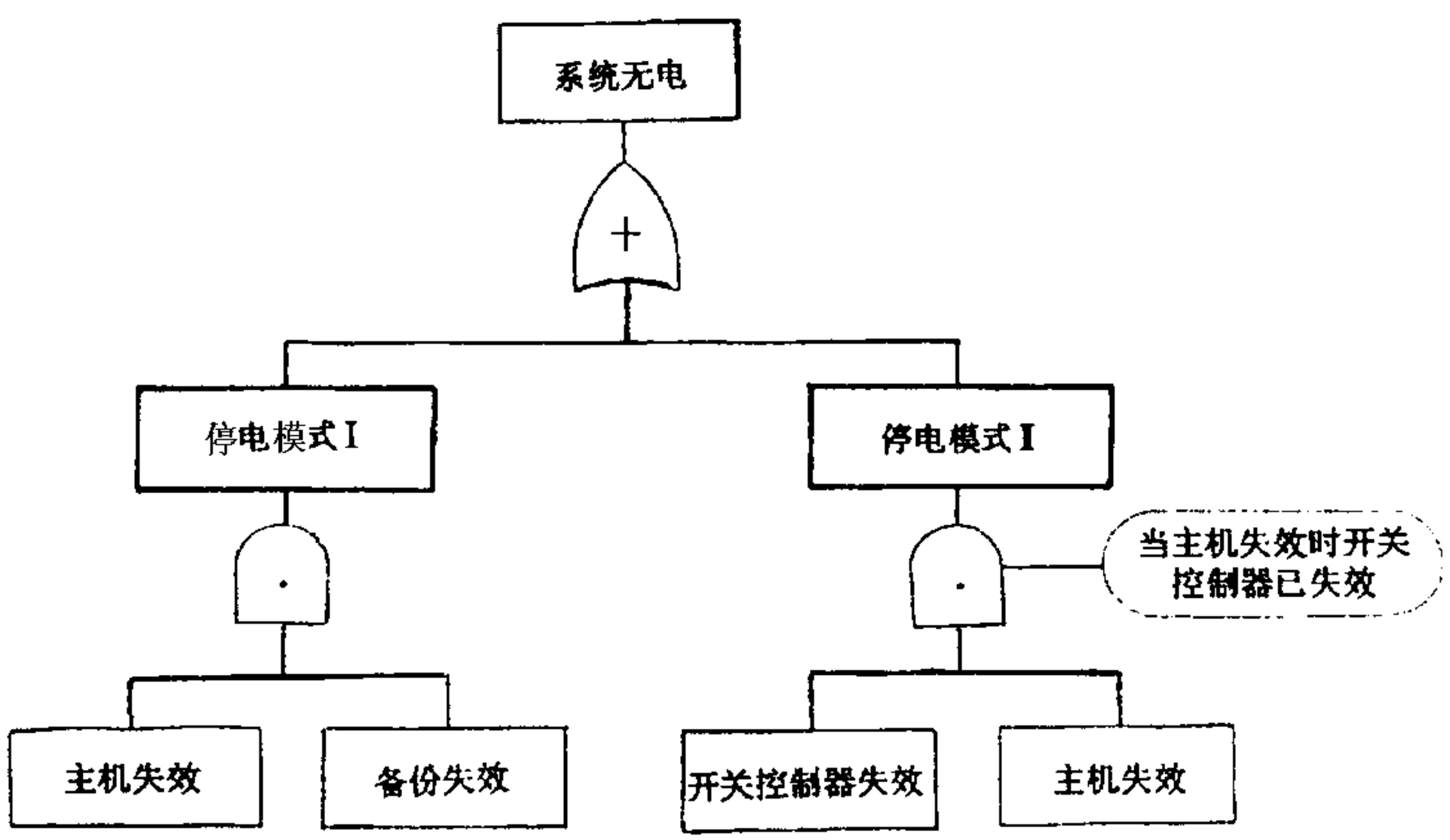


图 3.10　系统停电故障分析

3.2.5　表决门　voting gete

表决门表示仅当 n 个输入事件中有 r 个或 r 个以上的事件发生时，输出事件才发生 ($1 \leqslant r \leqslant n$)。其图形符号见图 3.11，示例见图 3.18。

或门和与门都是表决门的特例,或门是 $r = 1$ 的表决门,与门是 $r = n$ 的表决门。

图 3.11　表决门符号

3.2.6　异或门　exclusive－OR gate

异或门表示仅当单个输入事件发生时,输出事件才发生。其图形符号见图 3.12,示例见图 3.13。

图 3.12　异或门符号

异或门示例:双发电机电站丧失部分电力故障分析(图 3.13)。

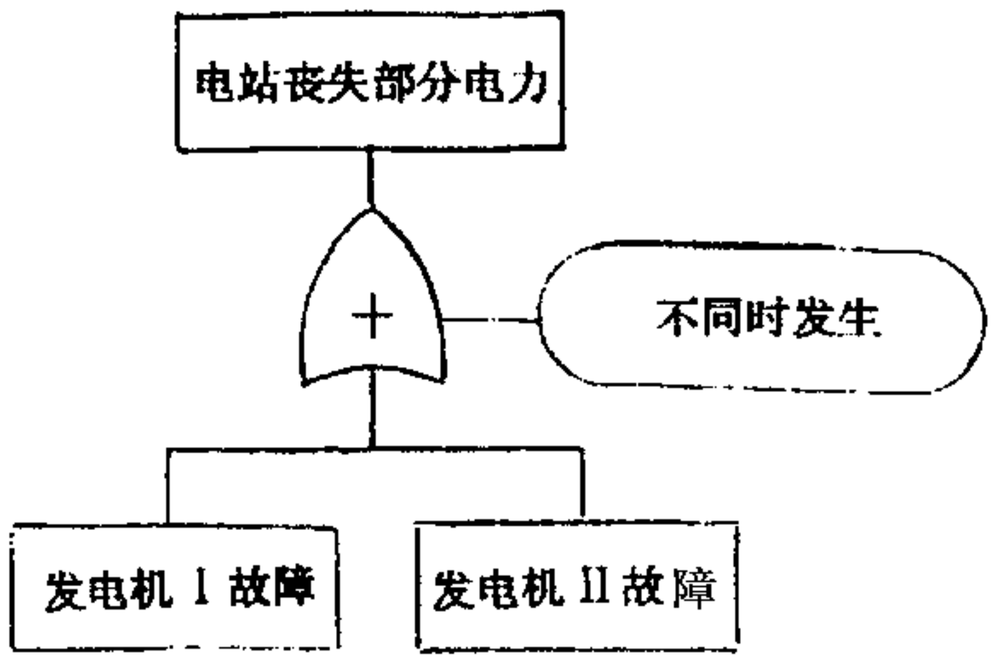


图 3.13　丧失部分电力故障分析

3.2.7　禁门　inhibit gate

禁门表示仅当禁门打开条件事件发生时,输入事件的发生方导致输出事件的发生。其图形符号见图 3.14。

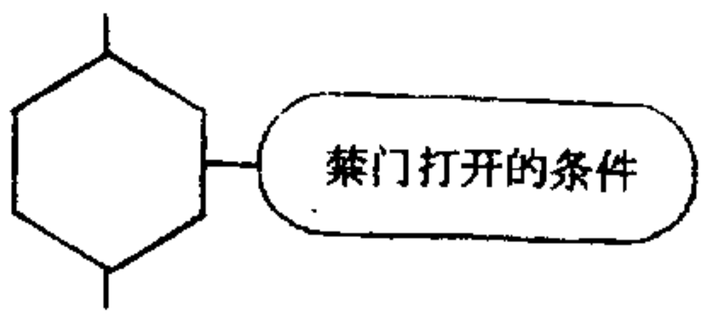


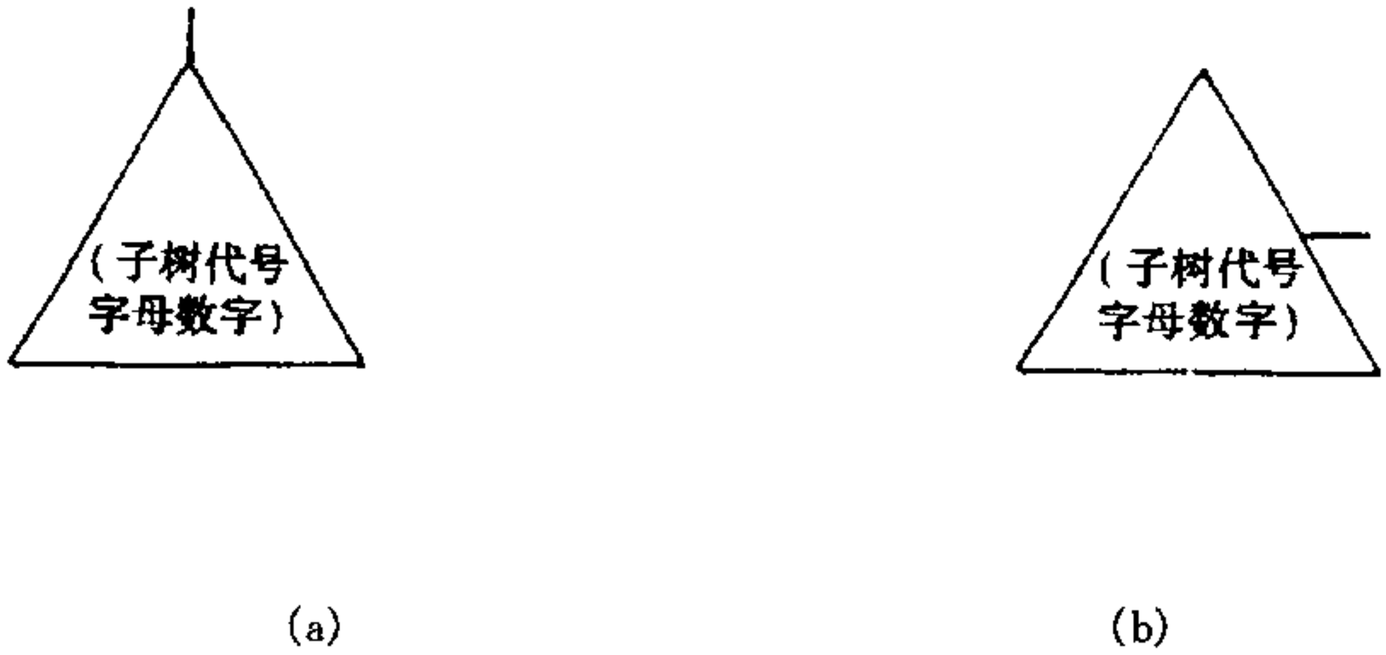
图 3.14　禁门符号

3.3 转移符号

转移符号是为了避免画图时重复和使图形简明而设置的符号。

3.3.1 相同转移符号 identical transfer symbol

图 3.15 是一对相同转移符号，用以指明子树的位置。图 3.15(a)符号表示“下面转到以字母数字为代号所指的子树去”。图 3.15(b)符号表示“由具有相同字母数字的符号处转到这里来”。



(a) (b)

图 3.15 相同转移符号

开关事件符号及相同转移符号示例：造船工人高空作业坠落死亡事故分析(图 3.16)。

3.3.2 相似转移符号 similar transfer symbol

图 3.17 是一对相似转移符号，用以指明相似子树的位置。图 3.17(a)符号表示“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”，不同的事件标号在三角形旁注明。图 3.17(b)符号表示“相似转移符号所指子树与此处子树相似但事件标号不同”。

表决门及相似转移符号示例：对某型号飞机不能正常飞行的分析。已知该机三台发动机中若有二台发生故障时便不能正常飞行(图 3.18)。

IEC 1025 推荐故障树符号表见附录 C(参考件)。

3.4 故障树 fault tree

故障树是一种特殊的倒立树状逻辑因果关系图，它用前述的事件符号、逻辑门符号和转移符号描述系统中各种事件之间的因果关系。逻辑门的输入事件是输出事件的“因”，逻辑门的输出事件是输入事件的“果”。

3.4.1 两状态故障树 2－state fault tree

如果故障树的底事件描述一种状态，而其逆事件也只描述一种状态，则称为两状态故障树。

3.4.2 多状态故障树 multistate fault tree

如果故障树的底事件描述一种状态，而其逆事件包含两种或两种以上互不相容的状态，并且在故障树中出现上述的两种或两种以上状态的底事件，则称为多状态故障树。

3.4.3 规范化故障树 normalized fault tree

将画好的故障树中各种特殊事件与特殊门进行转换或删减，变成仅含有底事件、结果事件以及“与”、“或”、“非”三种逻辑门的故障树，这种故障树称为规范化故障树。

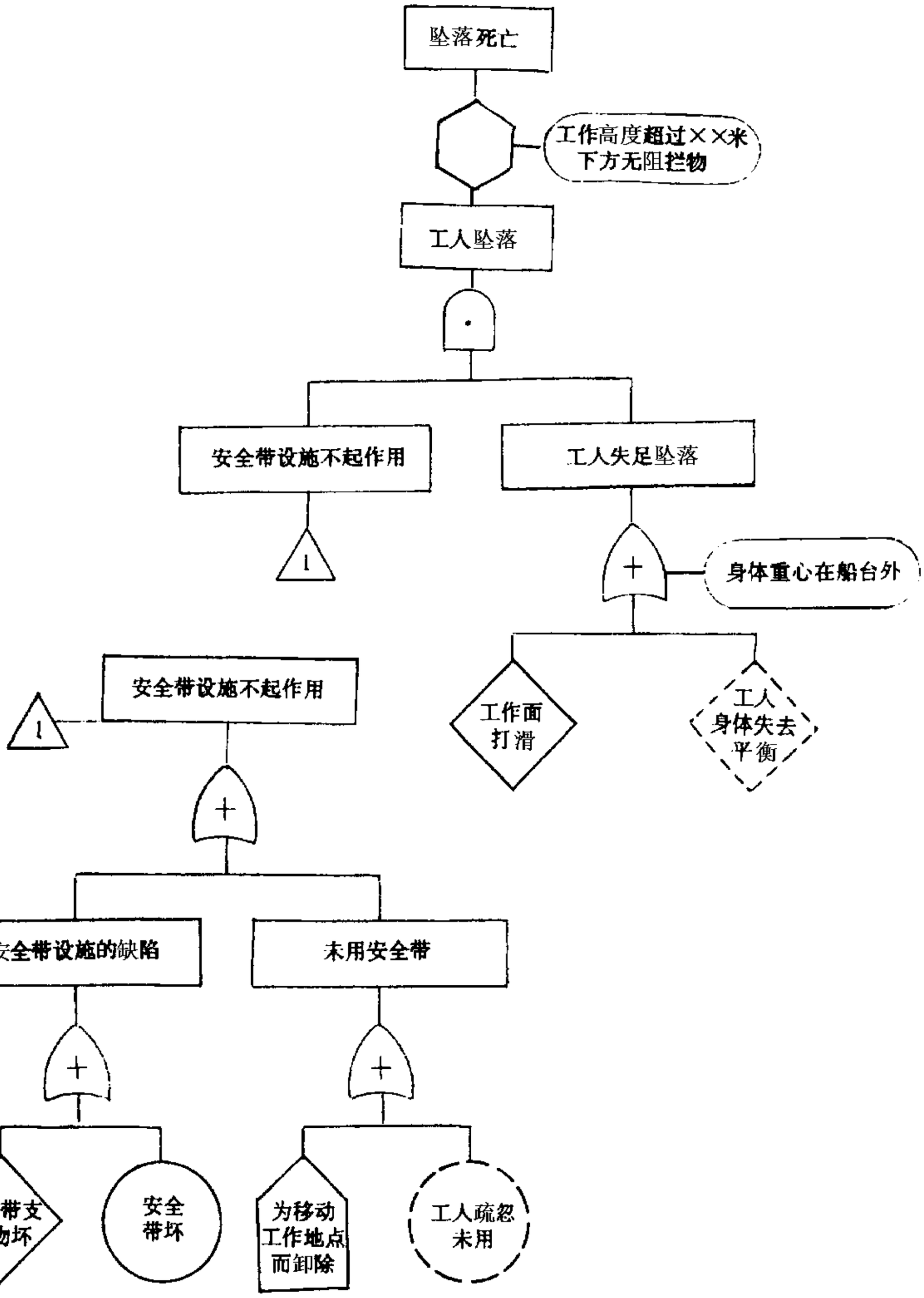
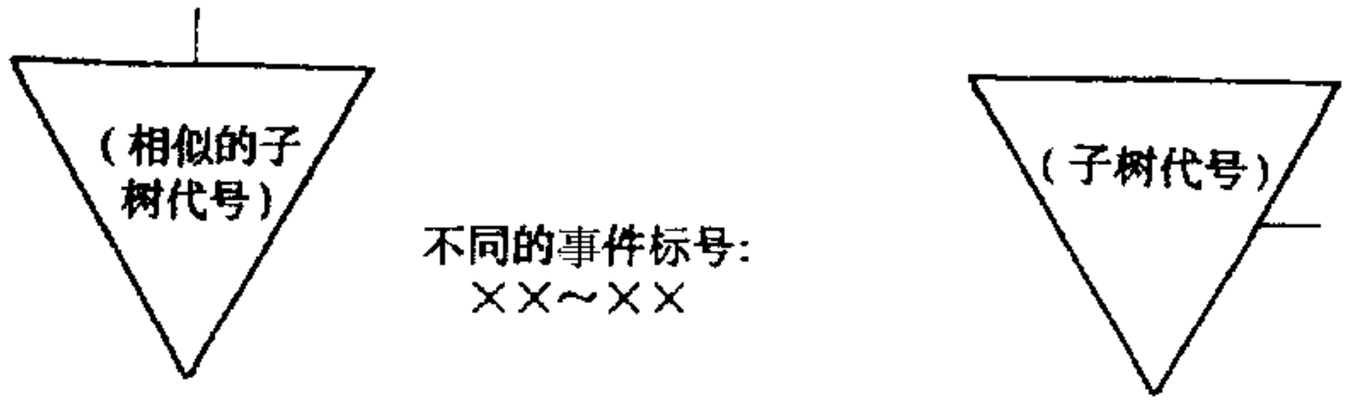


图 3.16　造船工人坠落死亡事故分析



(a)　　(b)

图 3.17　相似转移符号

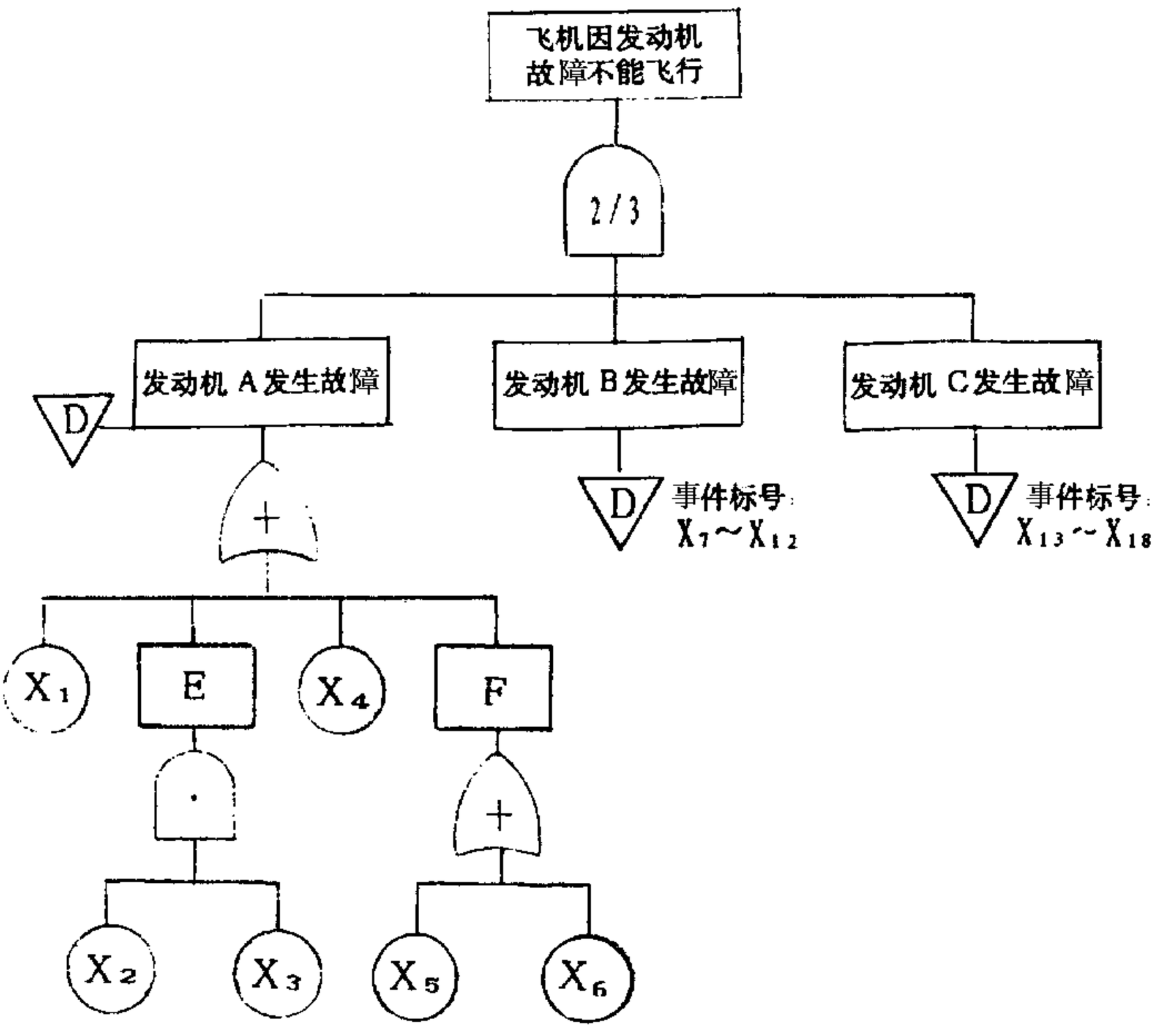


图 3.18　飞机因发动机故障不能飞行的故障树

3.5　结构函数　structure function

y 为描述顶事件状态的布尔变量，故障树的结构函数定义为：

$$y = \Phi(x_1, x_2, \cdots, x_n) = \begin{cases} 1, \text{若顶事件发生}, \\ 0, \text{若顶事件不发生}, \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

其中 n 为故障树底事件的数目，$x_1, x_2, \cdots x_n$ 为描述底事件状态的布尔变量，即

$$x_i = \begin{cases} 1, \text{若第 } i \text{ 个底事件发生}, \\ 0, \text{若第 } i \text{ 个底事件不发生}, \end{cases} \quad i = 1,2,\cdots,n \cdots\cdots (2)$$

3.6 单调性和关联性

3.6.1 关联性 relevance

对状态变量 x_i，若存在$(x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$，使得

$$\Phi(x_1, x_2, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n) \neq \Phi(x_1, x_2, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n) \cdots\cdots (3)$$

则称底事件状态变量 x_i 与结构函数 $\Phi(x_1, x_2, \cdots, x_n)$ 是关联的，或称第 i 个底事件与顶事件是关联的。

与顶事件不关联的底事件对顶事件是否发生不起任何作用，这样的底事件在分析中可以删去。

3.6.2 单调性 monotonicity

若由 $x_i \leqslant y_i, i = 1,2,\cdots,n$，可推得

$$\Phi(x_1, x_2, \cdots, x_n) \leqslant \Phi(y_1, y_2, \cdots, y_n) \cdots\cdots (4)$$

则称结构函数 $\Phi(x_1, x_2, \cdots, x_n)$ 是单调的。

3.7 单调故障树 monotone fault tree

结构函数具有单调性的故障树称为单调故障树。

仅含故障事件，以及与门、或门的故障树是单调故障树。

单调故障树的示例：某工厂金属屑伤眼事故分析(图 3.19)。

3.8 非单调故障树 non-monotone fault tree

结构函数不具有单调性的故障树称为非单调故障树。

非单调故障树的示例：化工厂传输液溢出贮液箱事故分析(图 3.20 和图 3.21)。

若假定：

a. 输入只有正常、过量两状态；

b. 阀门不会失效；

c. 探测器的故障只有一种：报低；

d. 控制器的故障只有一种：无反应，

则传输液溢出事故有图 3.21 的两状态非单调故障树。在图 3.21 的故障树中，“液位探测器报低”这一故障事件(X_2)出现时，如果“阀门控制器故障”这一故障事件 X_3 不出现(即$\overline{X_3}$出现)，则“液体累积过量Ⅲ”这一中间事件出现，导致液体溢出，顶事件发生。而如果 X_2 出现时 X_3 也出现($\overline{X_3}$不出现)，即液位探测器报低时阀门控制器也发生故障，则顶事件反而不发生。此时故障树结构函数是非单调的，所以这是一棵非单调故障树。

3.9 单调关联故障树 coherent fault tree

若故障树的结构函数是单调的，且所有底事件都与故障树的顶事件关联，则称该故障树为单调关联故障树。

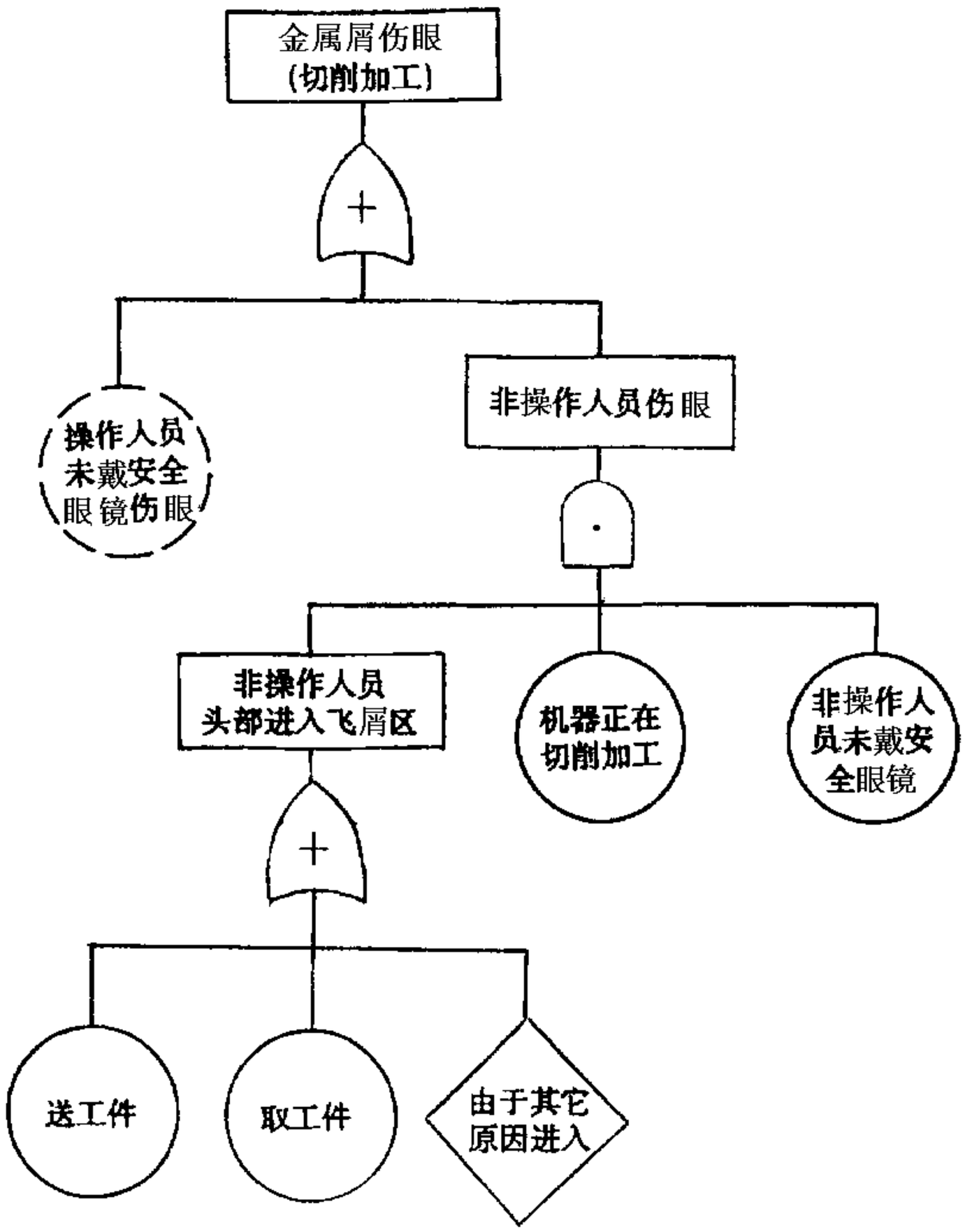


图 3.19　金属屑伤眼故障树

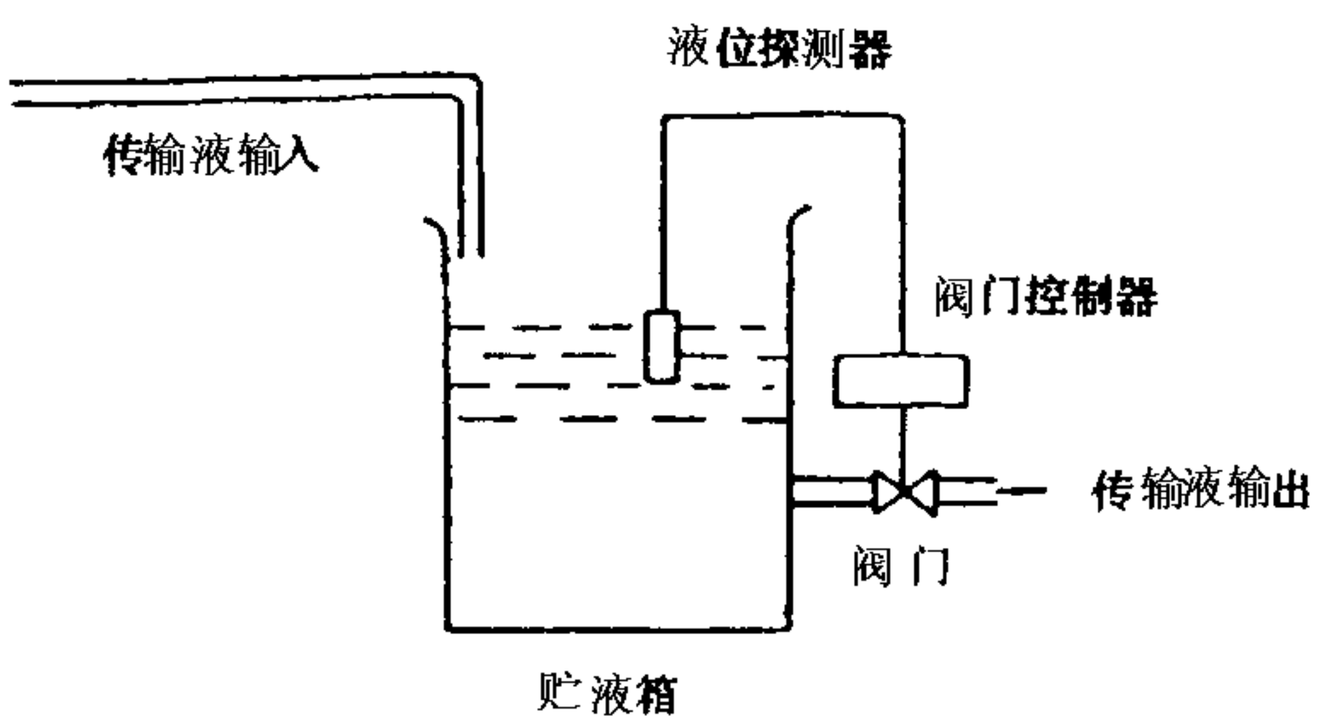


图 3.20　传输液流程示意图

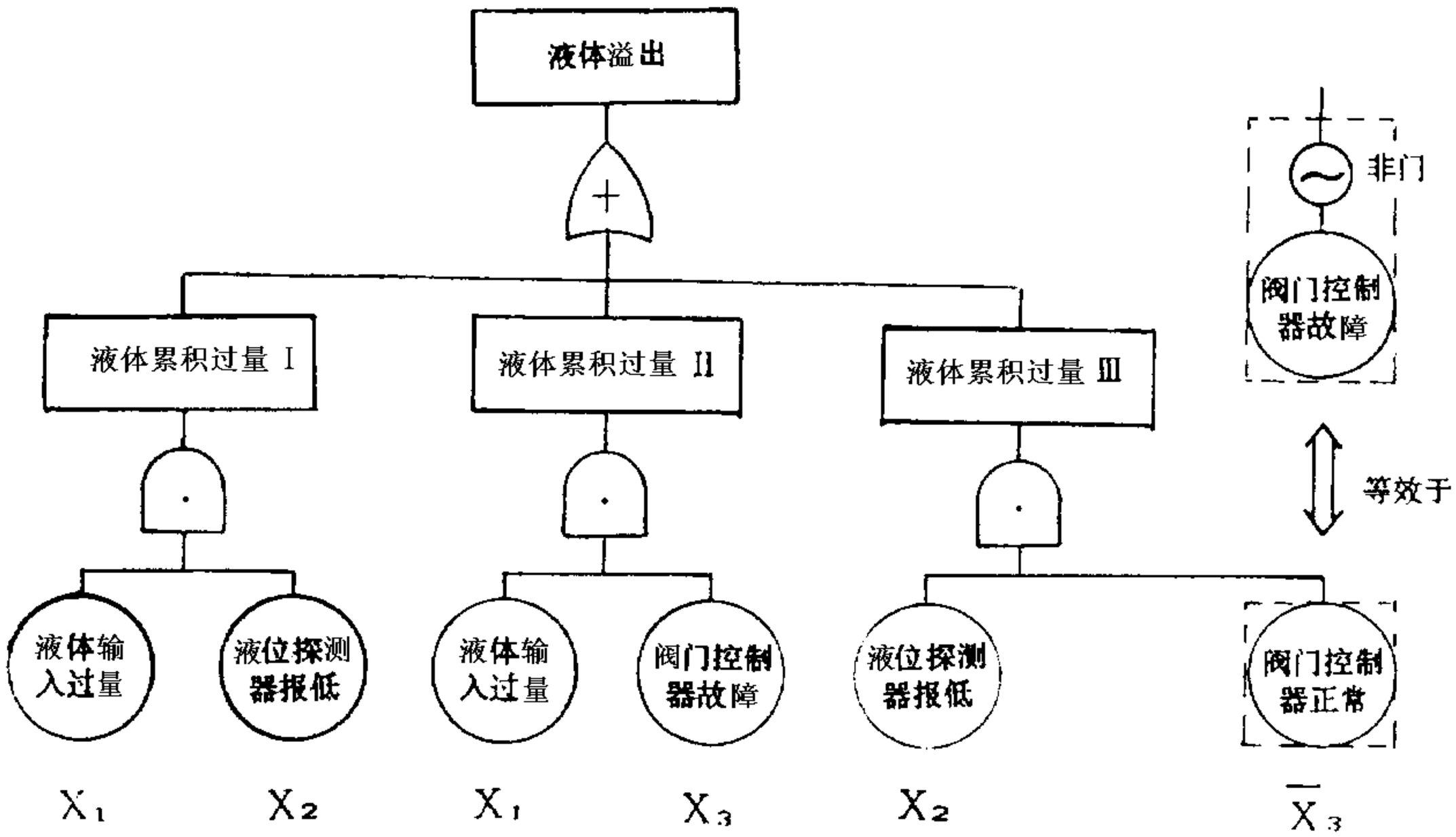


图 3.21　液体溢出贮液箱故障树

3.10　非单调关联故障树　non-coherent fault tree

不是单调关联的故障树称为非单调关联故障树。

与顶事件不关联的底事件在分析中可以删去，因此非单调关联故障树通常是指非单调故障树，即故障树的结构函数不单调。

3.11　故障树的模块和最大模块

3.11.1　模块　module

对于已经规范化和简化(见 5.2 条)的两状态故障树，模块是至少有两个底事件，但不是所有底事件的集合。集合中的这些底事件向上可到达同一个逻辑门，并且必须通过此门才能到达顶事件。故障树的所有其它底事件向上均不能到达该逻辑门。

3.11.2　最大模块　maximal module

经规范化和简化的故障树的最大模块是该故障树的一个模块，且没有其它模块包含它。

3.11.3　模块子树　modular sub-tree

故障树的模块连同向上可到的同一逻辑门和全部中间逻辑门和事件构成一株较小的故障树，称为原故障树的一个模块子树。

3.12　故障树的割集和最小割集

3.12.1　割集　cutset

割集是单调故障树的若干底事件的集合，如果这些底事件都发生将导致顶事件发生。

3.12.2　最小割集　minimal cutset

最小割集是底事件的数目不能再减少的割集，即在该最小割集中任意去掉一个底事件之后，剩下的底事件集合就不是割集。一个最小割集代表引起故障树顶事件发生的一种故障模式。

3.13 故障概率函数 failure probabilistic function

在故障树所有底事件互相独立的条件下，顶事件发生的概率 Q 是底事件发生概率 $q_1, q_2, \cdots, q_n$ 的一个函数，记为

$$Q = \mathrm{Q}(q_1, q_2, \cdots, q_n) \quad \cdots\cdots (5)$$

称其为故障树的故障概率函数。

3.14 重要度

故障树中的底事件并非同等重要的。若能对故障树中每个底事件的重要性程度给予定量的描述，对系统设计和故障分析都是很有价值的。几个常用的底事件的重要度定义如下。

3.14.1 底事件结构重要度 structure importance of bottom event

第 i 个底事件的结构重要度为：

$$I_\Phi(i) = \frac{1}{2^{n-1}} \sum_{(x_1, \cdots, x_{i-1}, \cdot, x_{i+1}, \cdots, x_n)} [\Phi(x_1, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n) - \Phi(x_1, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n)], \quad i = 1, 2, \cdots, n \quad \cdots\cdots (6)$$

其中 $\Phi(\cdot)$ 是故障树的结构函数，$\sum_{(x_1, \cdots, x_{i-1}, \cdot, x_{i+1}, \cdots, x_n)}$ 是对 $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$ 分别取 0 或 1 的所有可能求和 。

底事件结构重要度从故障树结构的角度反映了各底事件在故障树中的重要程度。

3.14.2 底事件概率重要度 probabilistic importance of bottom event

在故障树所有底事件互相独立的条件下，第 i 个底事件的概率重要度为

$$I_P(i) = \frac{\partial}{\partial q_i} \mathrm{Q}(q_1, q_2, \cdots, q_n), \quad i = 1, 2, \cdots, n \quad \cdots\cdots (7)$$

其中 $\mathrm{Q}(q_1, q_2, \cdots, q_n)$ 为故障树的故障概率函数。

第 i 个底事件的概率重要度表示，第 i 个底事件发生概率的微小变化而导致顶事件发生概率的变化率。

3.14.3 底事件的相对概率重要度 relative probabilistic importance of bottom event

在故障树所有底事件互相独立的条件下，第 i 个底事件的相对概率重要度为

$$I_C(i) = \frac{q_i}{Q(q_1, q_2, \cdots, q_n)} \cdot \frac{\partial}{\partial q_i} \quad Q(q_1, q_2, \cdots, q_n), \quad i = 1, 2, \cdots, n \quad \cdots\cdots (8)$$

其中 $Q(q_1, q_2, \cdots, q_n)$ 为故障树的故障概率函数。

第 i 个底事件的相对概率重要度表示，第 i 个底事件发生概率微小的相对变化而导致顶事件发生概率的相对变化率。

3.14.4 底事件的相关割集重要度 correlated cutset importance of bottom event

若 $\mathrm{X}_1, \mathrm{X}_2, \cdots, \mathrm{X}_n$ 是故障树的所有底事件，$C_1, C_2, \cdots, C_r$ 是由底事件组成的故障树的所有最小割集，其中包含第 i 个底事件的最小割集为 $C_1^{(i)}, C_2^{(i)}, \cdots, C_{r_i}^{(i)}$，记

$$Q_i = P(\sum_{k=1}^{r_i} \prod_{X_j \in C_k^{(i)}} \mathrm{X}_j) \quad \cdots\cdots (9)$$

以上∑和∏分别表示集合(事件)运算的并和交。当故障树所有底事件相互独立的条件下，Q_i 是底事件发生概率 $q_1, q_2, \cdots, q_n$ 的函数

$$Q_i = Q_i(q_1, q_2, \cdots, q_n) \quad \cdots\cdots (10)$$

第 i 个底事件的相关割集重要度定义为

$$I_{RC}(i) = \frac{Q_i(q_1, q_2, \cdots, q_n)}{Q(q_1, q_2, \cdots, q_n)} \quad \cdots\cdots (11)$$

其中 $Q(q_1, q_2, \cdots, q_n)$ 为故障树的故障概率函数。

第 i 个底事件的相关割集重要度表示:包含第 i 个底事件的所有故障模式中至少有一个发生的概率与顶事件发生的概率之比。

4 一般要求

4.1 目的

故障树分析以一个不希望的系统故障事件(或灾难性的系统危险)即顶事件作为分析的目标,通过由上向下的严格按层次的故障因果逻辑分析,逐层找出故障事件的必要而充分的直接原因,最终找出导致顶事件发生的所有原因和原因组合。在具有基础数据时计算出顶事件发生概率和底事件重要度等定量指标。

故障树分析应和 GJB 368A、GJB 450、GJB 900 的工作项目相协调,特别是和故障模式及影响分析(FMEA)、故障模式影响及危害性分析(FMECA)、初步危险分析、分系统危险分析、系统危险分析等工作项目协调配合,相辅相成,更全面地查明系统薄弱环节,更有效地改善系统安全性、可靠性、维修性。

4.2 故障树分析的准备工作

4.2.1 熟悉资料

必须熟悉设计说明书、原理图(流程图、结构图)、运行规程、维修规程和有关资料。实际上,开始建树时,资料往往不全,必须补充收集某些资料或作必要假设来弥补这种欠缺。随着资料的逐步完善,故障树也会修改得更加符合实际情况和更加完善。

4.2.2 熟悉系统

a. 应透彻掌握系统设计意图、结构、功能、边界(包括人机接口)和环境情况;

b. 辨明人的因素和软件对系统的影响;

c. 辨识系统可能采取的各种状态模式及它们和各单元状态的对应关系,辨识这些模式之间的相互转换,必要时应绘制系统状态模式及转换图以帮助弄清系统成功或故障与单元成功或故障之间的关系,有利于正确地建造故障树;

d. 根据系统复杂程度和要求,必要时应进行系统 FME(C)A 以帮助辨识各种故障事件以及人的失误和共因故障;

e. 根据系统复杂程度,必要时应绘制系统可靠性框图以帮助正确形成故障树的顶部结构和实现故障树的早期模块化以缩小树的规模;

f. 为透彻地熟悉系统,建树者除完成上述工作外还应随时征求有经验的设计人员和使用、维修人员的意见,最好有上述人员参与建树工作,方能保证建树工作顺利开展和建成的故

障树的正确性以达到预期的分析目的。

4.2.3 确定分析目的

应根据任务要求和对系统的了解确定分析目的。同一个系统,因分析目的不同,系统模型化结果会大不相同,反映在故障树上也大不相同。如果本次分析关注的对象是硬件故障,系统模型化时可以略去人的因素;如果关注对象是内部事件,则模型化将不考虑外部事件。有时(但不是所有场合)需要考虑硬件、软件故障、人的失误和外部事件等所有因素。

4.2.4 确定故障判据

根据系统成功判据来确定系统故障判据,只有故障判据确切,才能辨明什么是故障,从而才能正确确定导致故障的全部直接的必要而又充分的原因。

4.2.5 确定顶事件

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可能不止一个,在充分熟悉资料和系统的基础上,做到既不遗漏又分清主次地将全部重大故障事件一一列举,必要时可应用 FME(C)A,然后再根据分析目的和故障判据确定出本次分析的顶事件。

4.3 一般分析程序

4.3.1 建造故障树

本指导性技术文件推荐演绎法人工建树。

4.3.2 故障树规范化、简化和模块分解

必须将建好的故障树规范化以便于分析,同时尽可能对故障树进行简化和模块分解以节省分析工作量。

4.3.3 定性分析

用下行法或上行法求出单调故障树所有最小割集,即所有导致顶事件发生的系统故障模式。在没有基础数据因而无法进一步定量分析的情形下,可以仅作定性比较。

4.3.4 定量分析

在各个底事件相互独立和已知其发生概率的条件下,求出单调故障树顶事件发生概率和一些重要度指标。

4.3.5 多状态故障的处理

对于工程上必须考虑的系统及其组成单元多状态故障,可参考附录 B(补充件)的办法处理。

4.4 故障树分析报告的编写

报告可包括以下内容:

a. 前言(指明本次分析的任务,所涉及的范围);

b. 系统描述(系统的功能原理、边界定义、运行状态描述);

c. 基本假设;

d. 系统故障的定义和判据;

e. 系统顶事件的定义和描述;

f. 故障树建造;

g. 故障树的定性分析;

h. 故障树的定量分析；

i. 故障树分析的结果和建议；

j. 附件。

附件可包括前 9 部分未给出的必要的图表和说明资料，例如：

a. 可靠性数据表及数据来源说明；

b. 其他希望补充说明的系统资料，如系统原理图，功能框图，可靠性框图等；

c. 故障树图；

d. 最小割集清单；

e. 重要度排序表。

5 详细要求

5.1 故障树的建造

5.1.1 目的

通过建树透彻了解系统的故障逻辑关系，找出导致顶事件的所有基本故障原因事件或基本故障原因事件组合，从而辨识出系统在安全性或可靠性设计上的薄弱环节以便改善设计。故障树的建造是实施故障树定性、定量分析的最基本前提条件。

5.1.2 建树指南

本指南仅给出人工演绎建造故障树的基本规则和方法。人工建树也往往借助计算机画故障树，这和用计算机自动建树不同。计算机自动建树是将所分析的系统按一定规则输入计算机后，计算机自动生成故障树。

建树的依据是系统图、本次分析目的及顶事件定义。如果需要考虑的顶事件包括几种不同的故障模式，例如“转速控制系统失控导致超速”、“转速控制系统故障造成低速”；或需要考虑部件的不同故障模式，例如“电阻器开路、短路、阻值漂移”，则按附录 B(补充件)的方法处理，而 5.1.3 条建树规则仍然适用。建成的故障树可能是单调的，也可能是非单调的；可能是两态的，也可能是多态的。

5.1.3 人工演绎法建树的基本规则

5.1.3.1 明确建树边界条件，确定简化系统图

故障树的边界应和系统的边界相一致，方能避免遗漏或出现不应有的重复；一个系统的部件以及部件之间的联接数目可能很多，但其中有些对于给定的顶事件是很不重要的，为了减小树的规模以突出重点，应在 FME(C)A 的基础上，将那些很不重要的部分舍去，从系统图的主要逻辑关系得到等效的简化系统图，然后从简化系统图出发进行建树。

划定边界、合理简化是完全必要的，同时，这又要非常慎重，避免主观地把看来“不重要”的底事件压缩掉，却把要寻找的隐患漏掉了。做到合理划定边界和简化的关键在于经过集思广益的推敲，作出正确的工程判断。

5.1.3.2 故障事件应严格定义

所有故障事件，尤其是顶事件必须严格定义，否则建出的故障树将不正确。

例如，原意希望分析“电路开关合上后电动机不转”，但由于省略，将事件表达为“电动机不

转”,如此得到不同的两棵故障树见图 5.1 和图 5.2,这导致了错误。

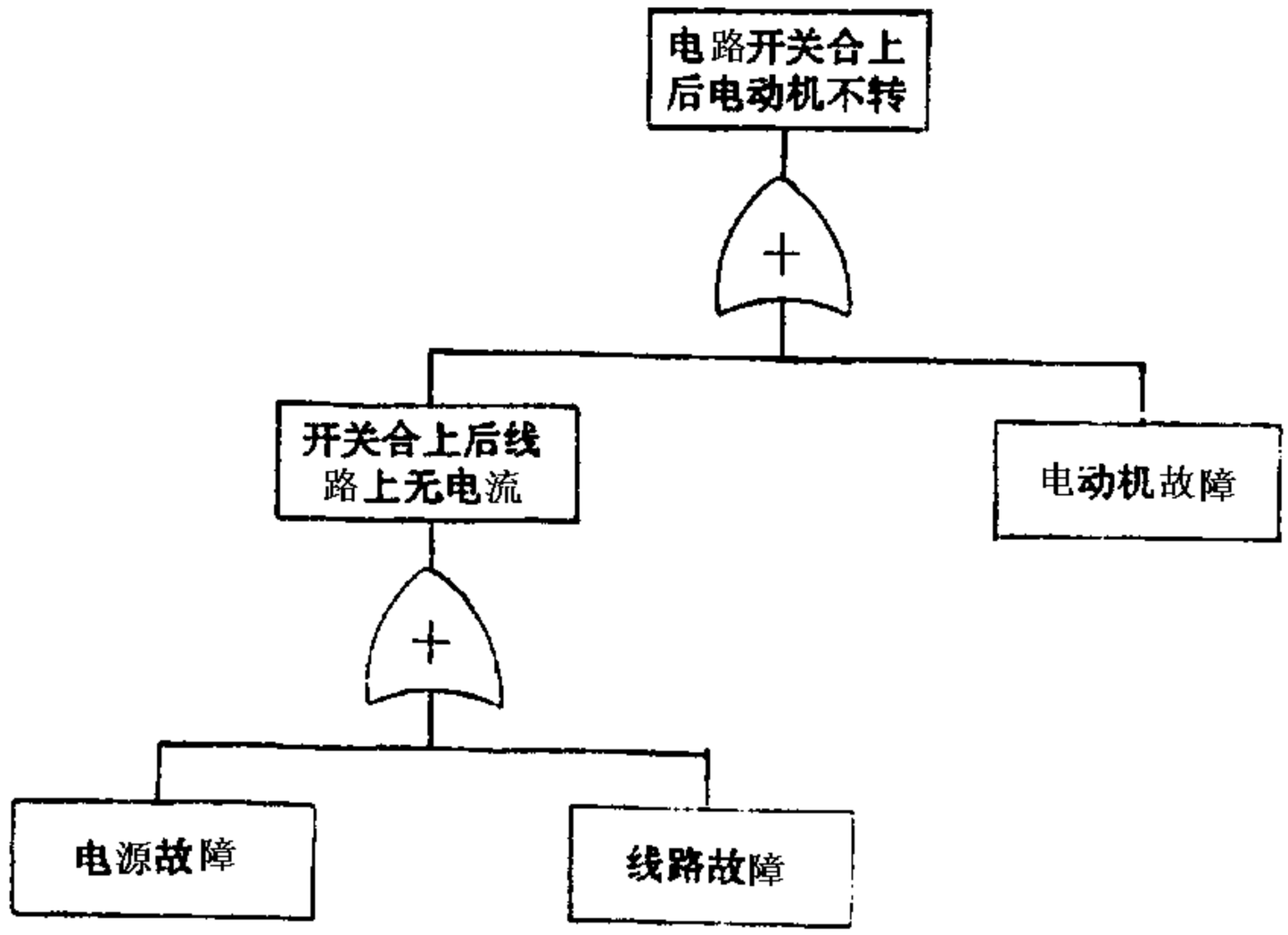


图 5.1　以“电路开关合上后电动机不转”为顶事件之故障树

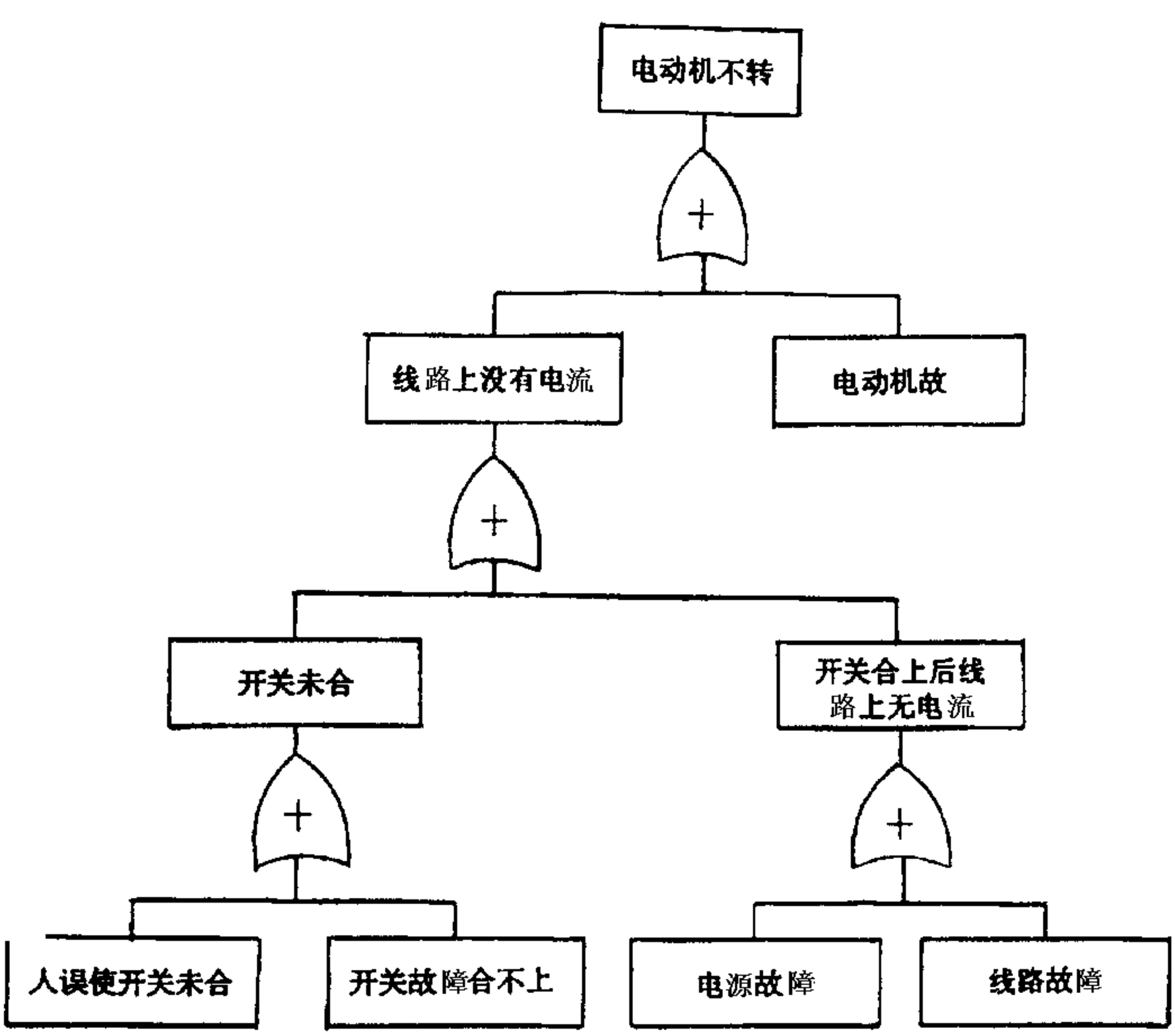


图 5.2　以“电动机不转”为顶事件之故障树

5.1.3.3　故障树演绎过程中首先寻找的是直接原因事件而不是基本原因事件

应不断利用“直接原因事件”作为过渡，逐步地无遗漏地将顶事件演绎为基本原因事件。

在故障树往下演绎过程中，还常常用等价的比较具体的或更为直接的事件取代比较抽象的或显得间接的事件，这时就会出现不经任何逻辑门的事件串，见图 5.4 所示。

5.1.3.4　应从上向下逐级建树

本条规则主要目的是避免遗漏。

例如一棵庞大的故障树，一级输入事件数可能超过 10，每一个输入都可能仍然是一棵庞大的子树，若未将 10 个输入的中间事件全都列出之前，比如列 9 个，就急于去发展其中某一个中间事件，这种发展工作甚至持续几天也难以完成，等全都发展完后，可能遗忘了还有第 10 个输入。

这条规则在采用图形编辑的故障树分析软件时尤其要注意遵守。

大的工程系统建造故障树时，应首先确定系统级顶事件，据以确定各分系统级顶事件；重视总体与分系统之间和分系统相互之间的接口，分层次、有计划、协调配合地进行故障树的建造，但从上到下进行故障演绎的逻辑相同。

5.1.3.5　建树时不允许逻辑门—逻辑门直接相连

本条规则防止建树者不从文字上对中间事件下定义即去发展该子树，5.1.3.2 条强调故障事件定义要严格，否则将会导致建树的错误。倘若建树时出现逻辑门—逻辑门相连而又根本不严格定义则更易出错，其次逻辑门—逻辑门相连的故障树使评审者无法判断对错，故不允许逻辑门—逻辑门直接相连。

在故障树已经建成，且确认无误情况下进行定性、定量分析时，这种逻辑门—逻辑门相连的故障树显得简明，可以使用，但需谨慎。

5.1.3.6　妥善处理共因事件

来自同一故障源的共同的故障原因会引起不同的部件故障甚至不同的系统故障。共同原因故障事件，简称共因事件。鉴于共因事件对系统故障发生概率影响很大，故建树时必须妥善处理共因事件。

若某个故障事件是共因事件，则对故障树的不同分支中出现的该事件必须使用同一事件标号。若该共因事件不是底事件，必须使用相同转移符号简化表示。一般说来，一个共因事件在同一系统故障树的不同子树中出现，这条规则往往可以得到遵守，但有时不同系统是相关的，比如公用同一电或水支持设施，甚至公用同一个阀门或管路，而这两个系统由不同人建树，这条规则往往得不到遵守，从而导致错误。因此对一些大项目实施故障树分析时，技术负责人一定要采取妥当的措施以保证规则能得到遵守，比如让同一个人负责有相同共因事件的不同系统故障树建造工作。

以上几条规则是故障树分析人员多年的经验教训总结，不遵守这些规则，将导致错、漏。一旦发生错、漏，不仅难以发现，即使发现了要全部改正过来，也很困难，因为这涉及不同的机构，不同的人，多张图纸和计算机的输入等等。所以要从建树开始，严格遵循规则，十分认真、仔细，一步步循序渐进，千万不要图省事。只有如此建成的树，方可作到错、漏最少，即使如此，建成的故障树也应请有实际工程经验又懂得可靠性知识的未参加建树人员审查，经审查并改正后的故障树才能成为进行故障树分析的基础，此时建树工作才告完成。

5.1.4　人工演绎建造故障树方法示例

人工建树是依靠建树人员对系统和故障树分析方法的理解，通过思考分析顶事件是怎么发生的，导致顶事件的直接原因事件是哪些？它们又是如何发生的？一直分析到底事件为止，并用有关的故障树符号将分析结果记录下来而形成故障树。

人工建造故障树方法，举例说明如下：

有一输变电系统见图 5.3 所示。A、B、C 为两级变电站，B、C 均由 A 供电。输电线 1、2 是 A 向 B 的输电线，输电线 3 是 A 向 C 的输电线，输电线 4、5 为站 B、站 C 之间的联络线（也是输电线）。输变电系统故障断电的判据为：

a. 站 B 停电；

b. 站 C 停电；

c. 站 B 和站 C 仅由同一条输电线供电，输电线将过载。

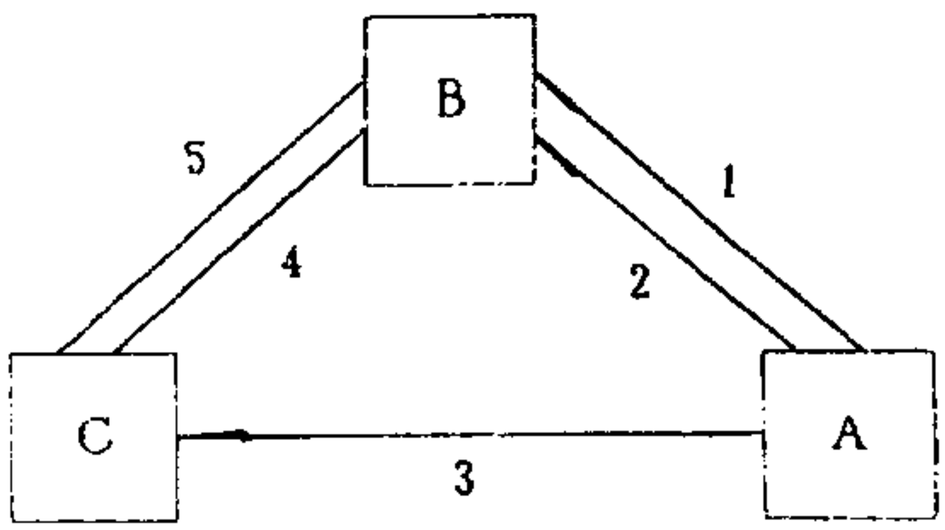


图 5.3　输变电系统图

本系统的不希望事件为系统故障停电，即顶事件。根据本次故障分析的目的是研究输电线路故障的影响。建树边界条件为：变电站本身的故障在故障树分析中可不予以考虑，这样故障树所涉及的基本事件数、逻辑门数均相应减少。其次，根据 5.1.3.2 条和 5.1.3.3 条，顶事件应严格定义或用一事件串对事件进一步解释，这形成本例建树的第一步。见图 5.4。

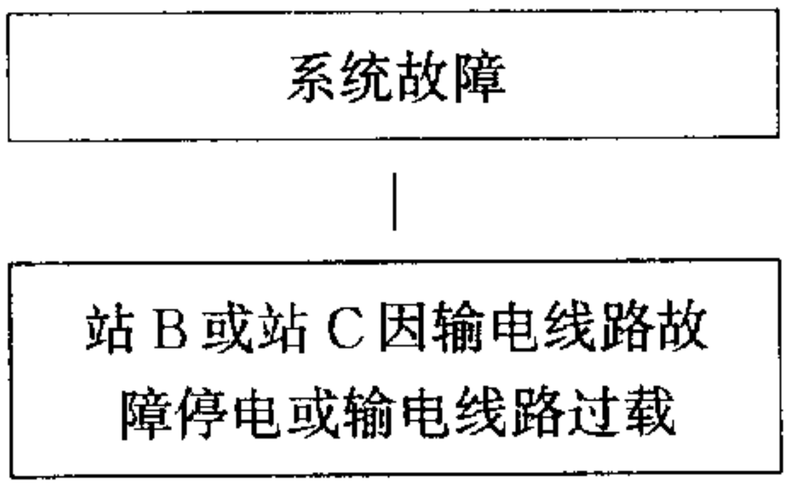


图 5.4　建树第一步

站 B 或站 C 停电或线路过载的直接原因事件显然就是上文已给出的三个故障判据事件，建树的第二步见图 5.5。

建树的第三步是发展图 5.5 左边的子树 D。从系统图可见“站 B 的输入线路上无电”的直接原因事件为来自站 A 及站 C 的输电线路上均无电，显然逻辑门应为“与门”，而该“与门”的输入事件为“由站 A 向站 B 的输电线路上无电”及“来自站 C 的输电线路上无电”。建树第三步见图 5.6。

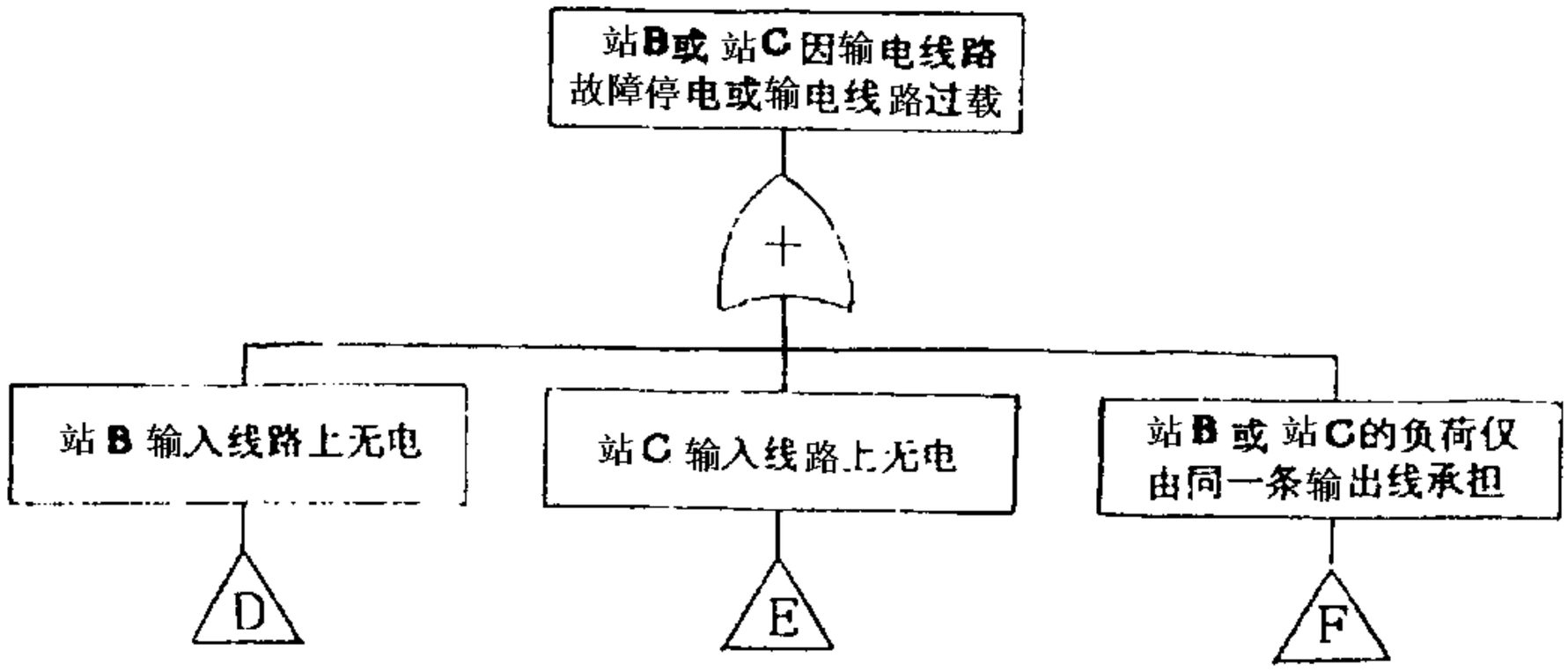


图 5.5　建树第二步。

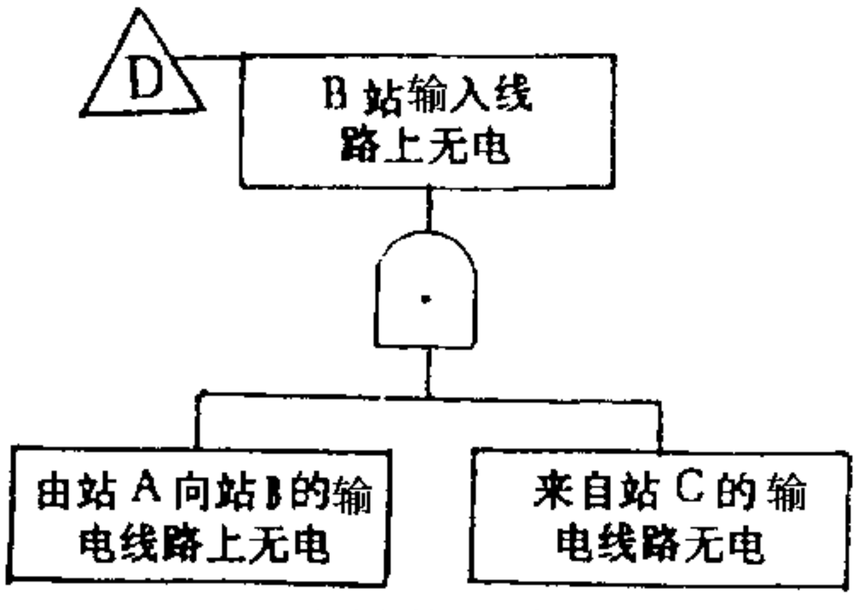


图 5.6　建树第三步

建树第四步是将中间结果事件“由站 A 向站 B 的输电线路无电”发展为底事件。由系统图可以看出，由站 A 向站 B 的输电线路有两条，即线路 1 及线路 2，故逻辑门应为“与门”，“与门”下的输入事件有 X_1“线路 1 故障断电”及 X_2“线路 2 故障断电”。此外，建树第四步还给中间结果事件“来自站 C 的输电线路无电”下的子树命名为 G。建树第四步见图 5.7。

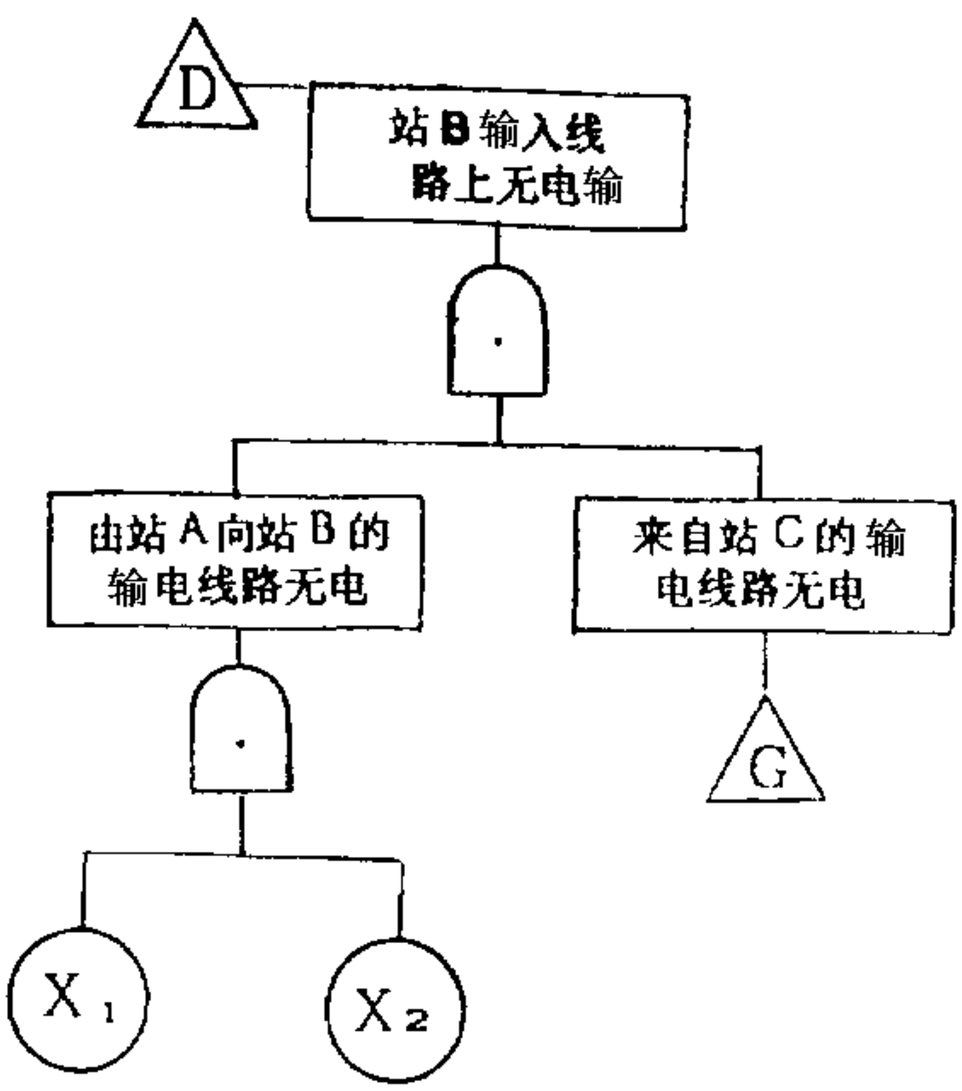


图 5.7　建树第四步

建树第五步是发展上图的右边子树 G。从系统图可以看出，导致中间结果事件“来自站 C 的输电线路无电”的直接原因事件为：或者“由站 C 向站 B 的输电线路故障”或者“由站 A 向站 C 的输电线路无电”，逻辑门为“或门”，该“或门”的输入事件为上述两个事件，建树第五步见图 5.8。

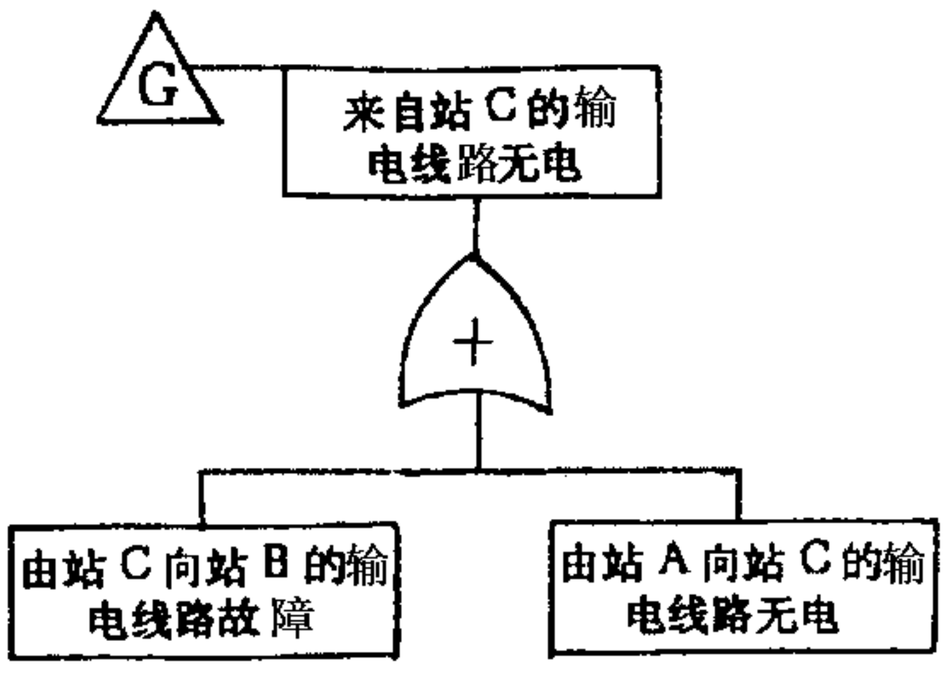


图 5.8　建树第五步

建树第六步是将上述子树 G 发展到底事件。由系统图可以看出，由站 A 向站 C 的输电线路只有一条(线路 3)，故事件“由站 A 向站 C 的输电线路无电”即为底事件 X_3“线路 3 无电”；由站 C 向站 B 的输电线路有两条，线路 4 和线路 5，故事件“由站 C 向站 B 的输电线路故障”下的逻辑门为“与门”，“与门”下的输入事件为 X_4“线路 4 故障断电”及 X_5“线路 5 故障断电”。建树第六步见图 5.9。

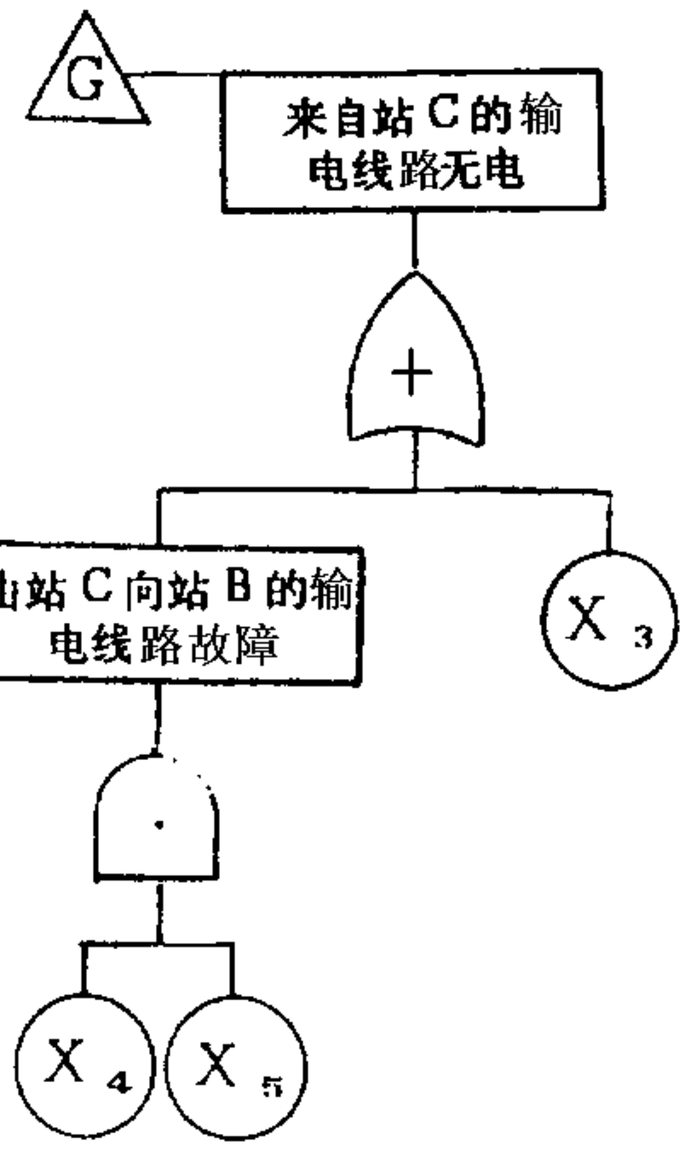


图 5.9　建树第六步

迄今子树 D 已发展完结，全部树叶均由底事件表示，现在就应回头发展子树 E。该子树的结果事件“站 C 的输入线路上无电”，从系统图可以看出，该事件的直接原因事件为“线路 3 故障断电”且“来自站 B 的输电线路无电”，得到对应子树即为建树的第七步，见图 5.10。

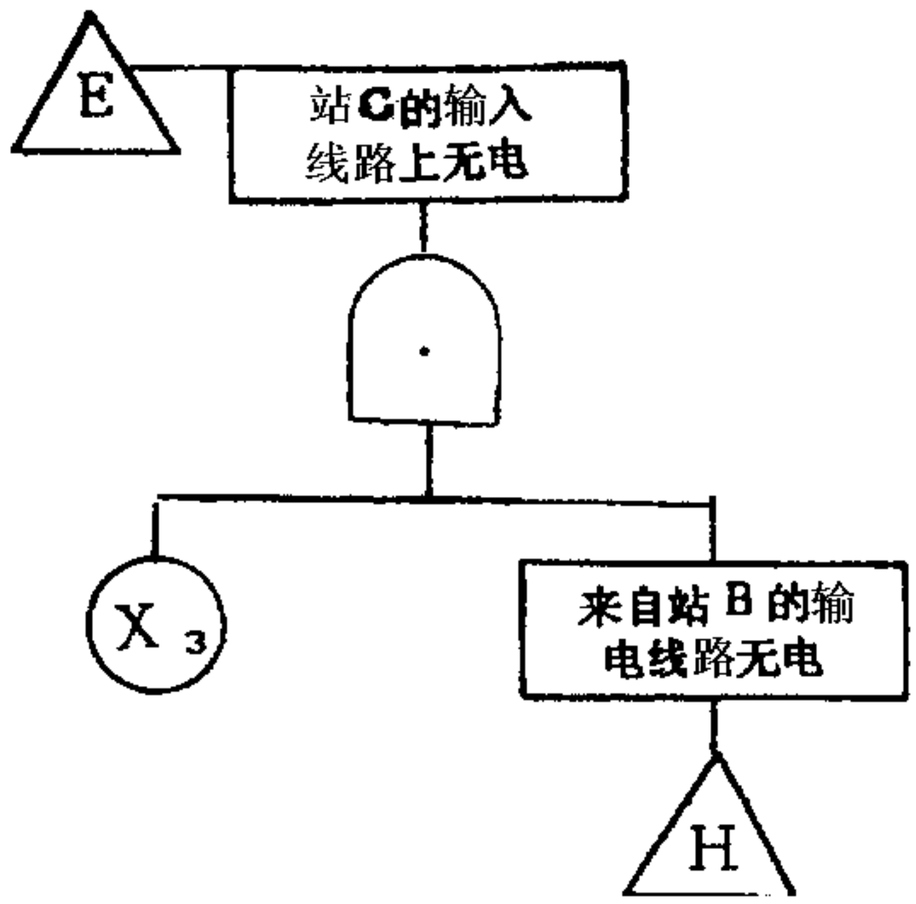


图 5.10　建树第七步

建树第八步是将子树 H 发展到底事件。由该子树的结果事件的定义可看出这是一个“或门”构成的子树，该或门的输入为两个事件，或者“线路 4 和线路 5 均故障断电”或者“线路 1 和线路 2 均故障断电”。而事件“线路 4 和线路 5 均故障断电”为一“与门”结构，“与门”之下的输入事件为底事件 X_4 和 X_5，事件“线路 1 和线路 2 均故障断电”也为一“与门”结构，“与门”之下的输入事件为底事件 X_1 和 X_2。第八步建成子树见图 5.11。

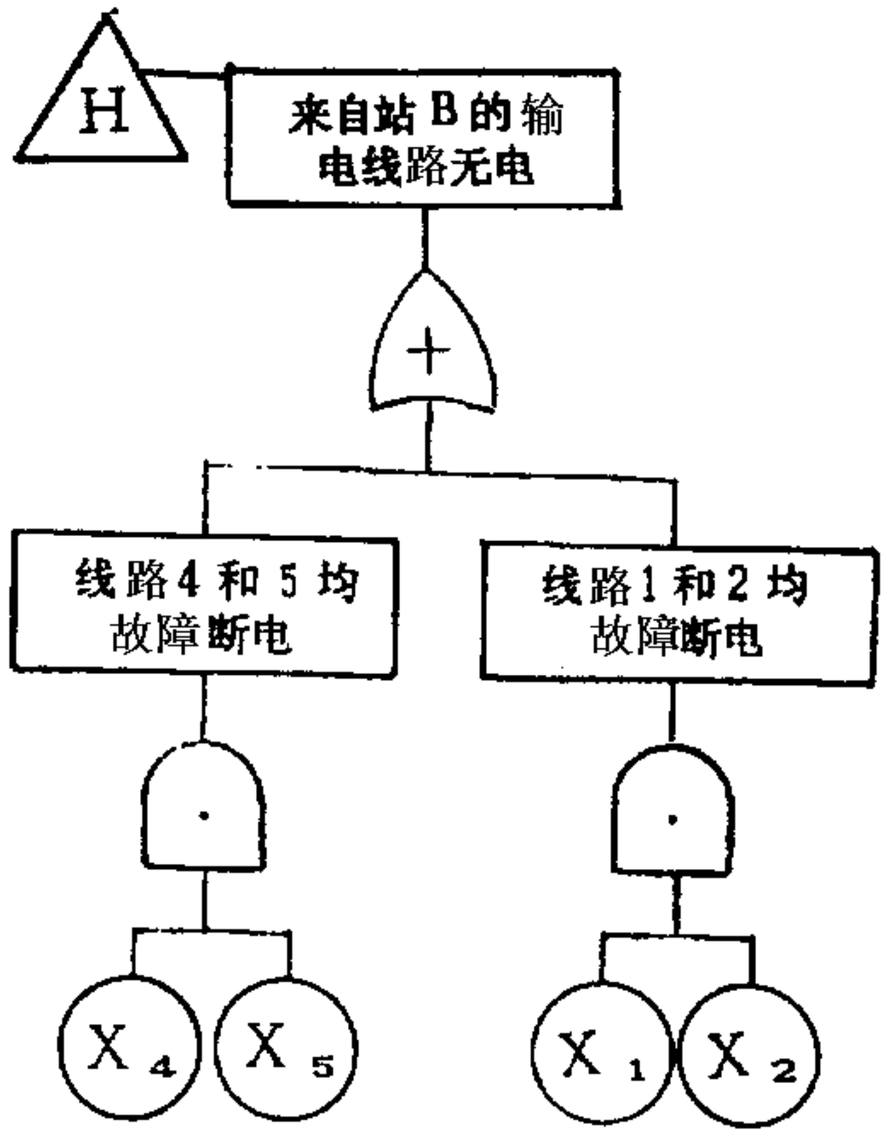


图 5.11　建树第八步

检查以上八步建成的故障树中，还有子树 F 等待发展。子树 F 的顶部事件为："站 B 或站 C 的负荷仅由同一条输电线承担"。由系统图可以看出，三条供电线，线路 1、2、3 中的任意两根若同时故障则该顶部事件发生而不必考虑 B、C 间联线路 4、5 的状况如何。建树第九步即是将子树 F 发展到底事件，显然逻辑门为 2/3 表决门，其输入事件为 X_1、X_2 和 X_3。建树第九步见图 5.12。

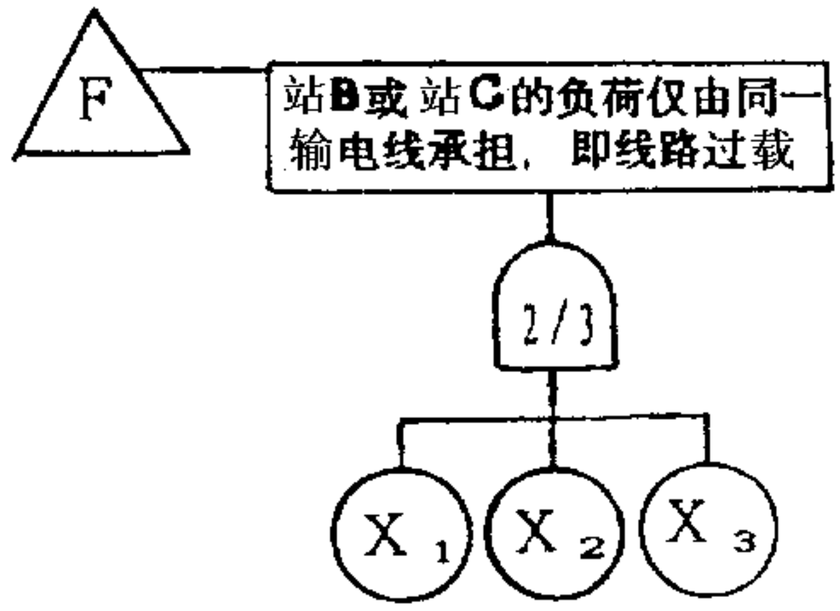


图 5.12　建树第九步

去掉以上九步得到的局部故障树中的转移符号，按照转移符号指明的联接位置即可拼成完整故障树，见图 5.13。

5.2　故障树规范化、简化和模块分解

5.2.1　目的

本节给出了将建好的故障树规范化的基本规则、故障树的简化和模块分解的原则。

5.2.2　规范化、简化和模块分解指南

由于现实的系统错综复杂，按上面方法建造出来的故障树也大不相同，因人而异。为了能用标准的程序对各种不同的故障树作统一的描述和分析，必须将建好的故障树变为规范化故障树，并尽可能对故障树进行简化和模块化，以便减少分析的工作量。

5.2.3　故障树规范化的基本规则

要将建好的故障树变为规范化的故障树，必须确定对特殊事件的处理规则和对特殊逻辑门进行逻辑等效变换的规则。

5.2.3.1　特殊事件的处理规则

5.2.3.1.1　未探明事件的处理规则

未探明事件可根据其重要性(如发生概率的大小，后果严重程度，等等)和数据的完备性，或者当作基本事件对待或者删去。重要且数据完备的未探明事件当作基本事件对待；不重要且数据不完备的未探明事件则删去；其它情况由分析者酌情决定。

5.2.3.1.2　开关事件的处理规则

将开关事件当作基本事件对待。

5.2.3.1.3　条件事件的处理规则

条件事件总是与特殊门联系在一起的，它的处理规则见 5.2.3.2 条特殊门的等效变换规

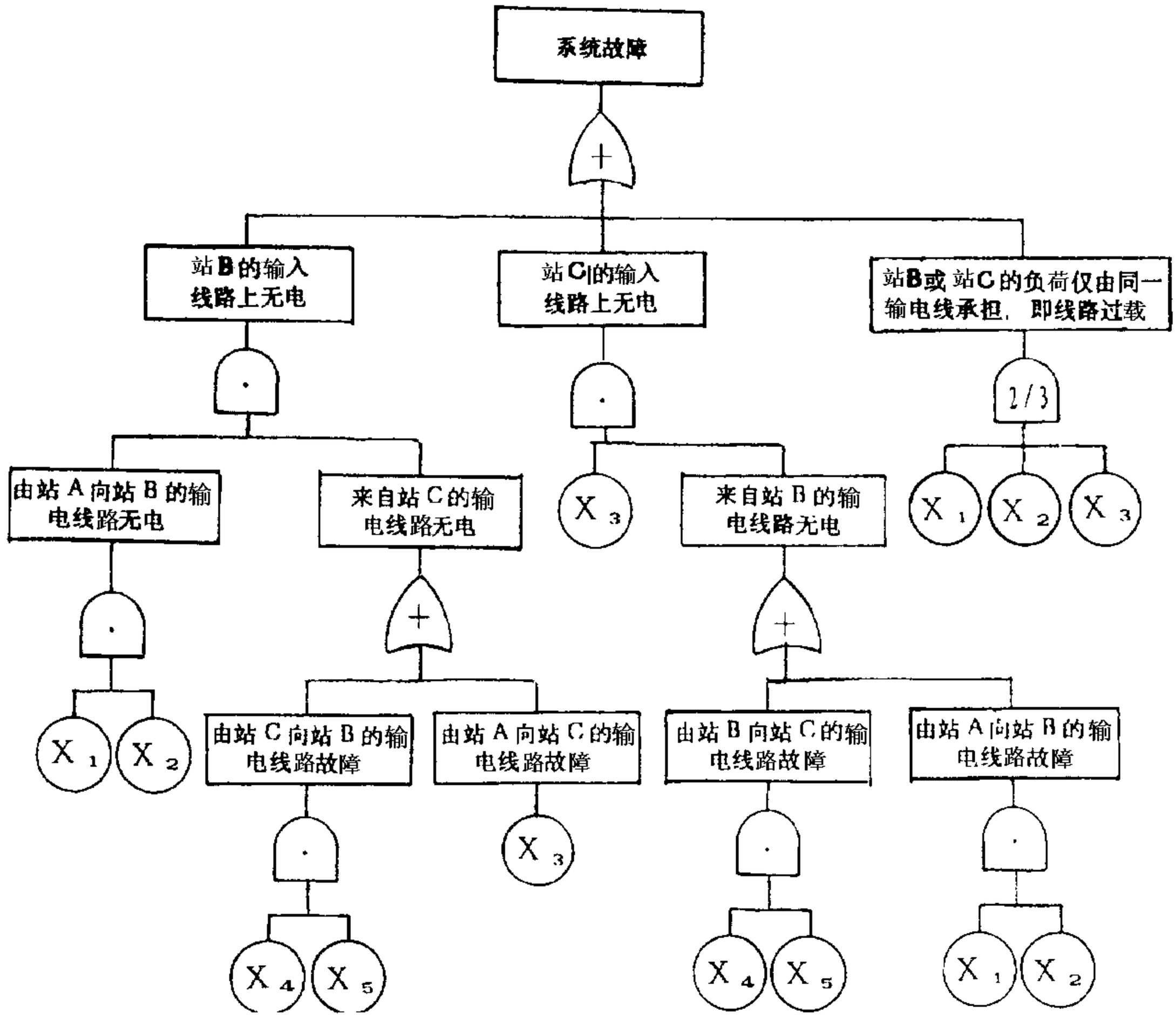


图 5.13　完整故障树

则。

5.2.3.2　特殊门的等效变换规则

5.2.3.2.1　顺序与门变换为与门的规则

输出不变，顺序与门变换为与门，其余输入不变，顺序条件事件作为一个新的输入事件。这条规则举例见图 5.14。

5.2.3.2.2　表决门变换为或门和与门的组合的规则

一个 r/n 表决门有以下两种或门和与门的组合等效变换：

a. 原输出事件下接一个或门，或门之下有 $\binom{n}{r}$ 个输入事件，每个输入事件之下再接一个与门，每个与门之下有 r 个原输入事件。举例见图 5.15；

b. 原输出事件下接一个与门，与门之下有 $\binom{n}{n-r+1}$ 个输入事件，每个输入事件之下再接一个或门，每个或门之下有 $n-r+1$ 个原输入事件。举例见图 5.16。

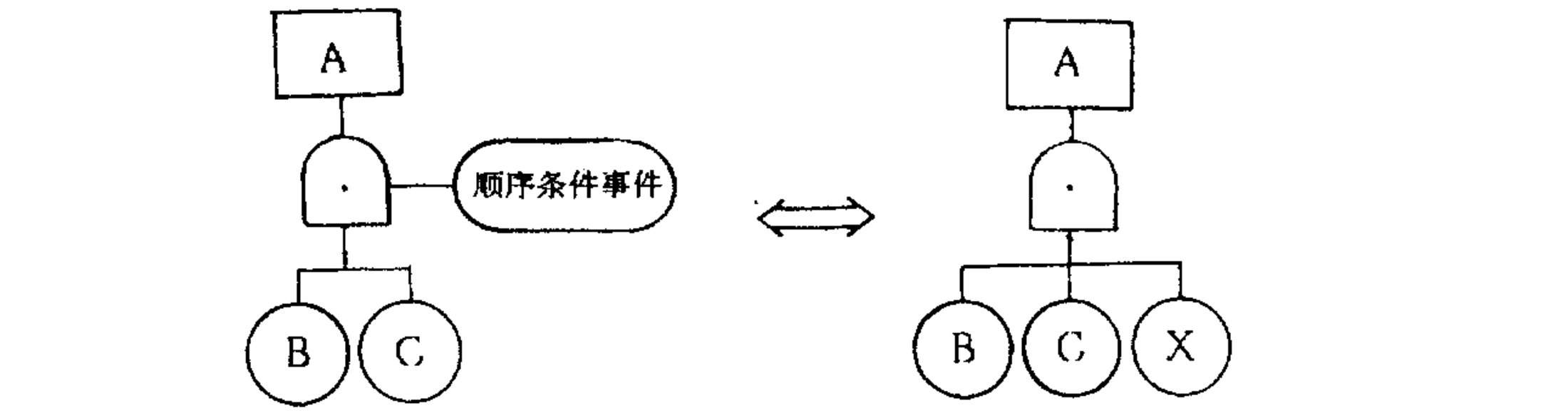


图 5.14　顺序与门变换为与门

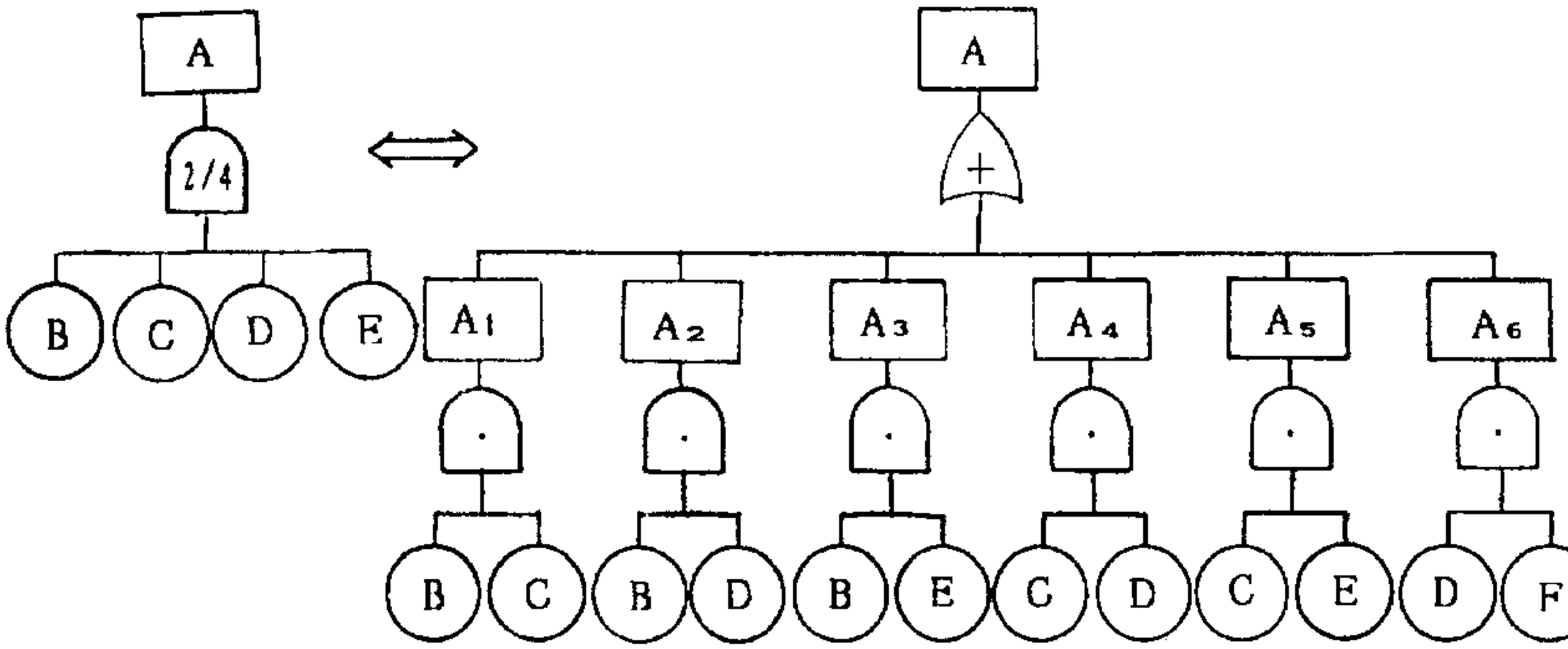


图 5.15　2/4 表决门变换为或门与门的组合

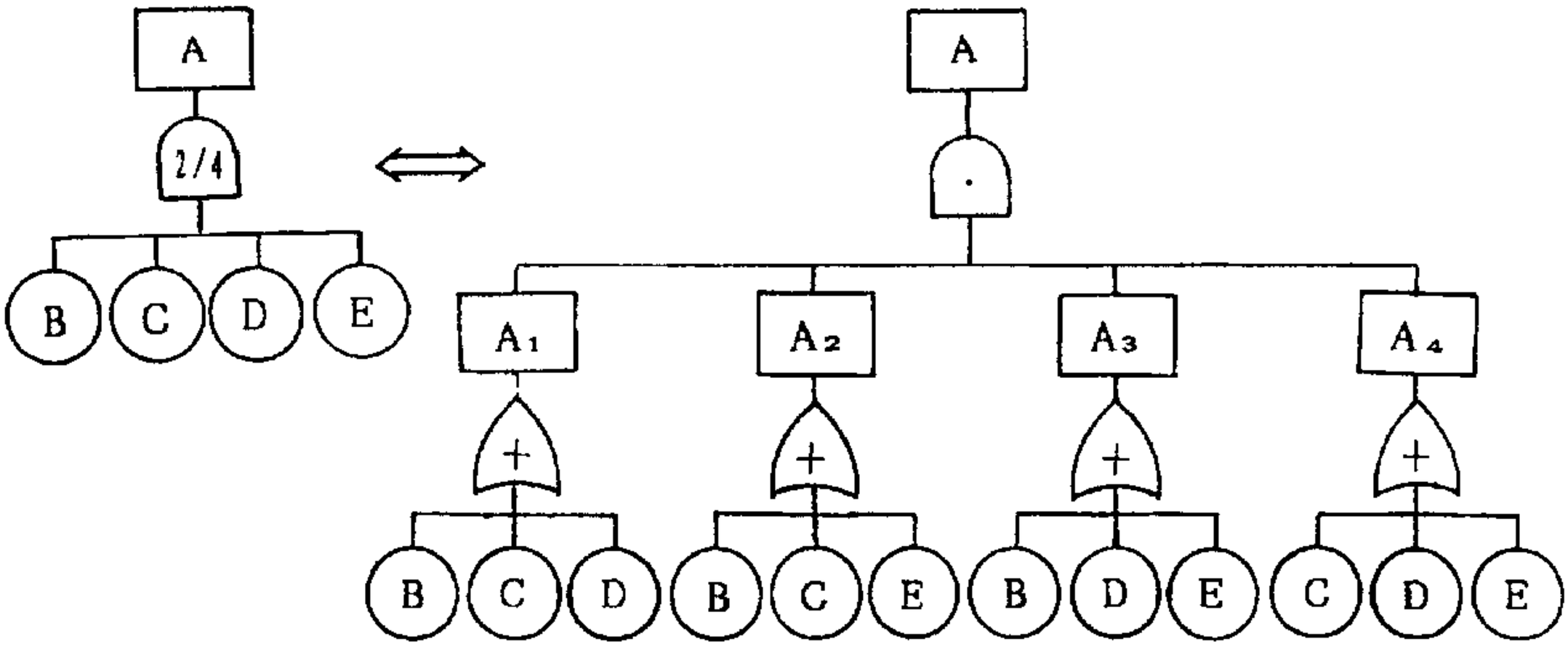


图 5.16　2/4 表决门变换为与门或门的组合

5.2.3.2.3　异或门变换为或门、与门和非门组合的规则

原输出事件不变，异或门变为或门，或门下接两个与门，每个与门之下分别接一个原输入事件和一个非门，非门之下接一个原输入事件。这条规则举例见图 5.17。

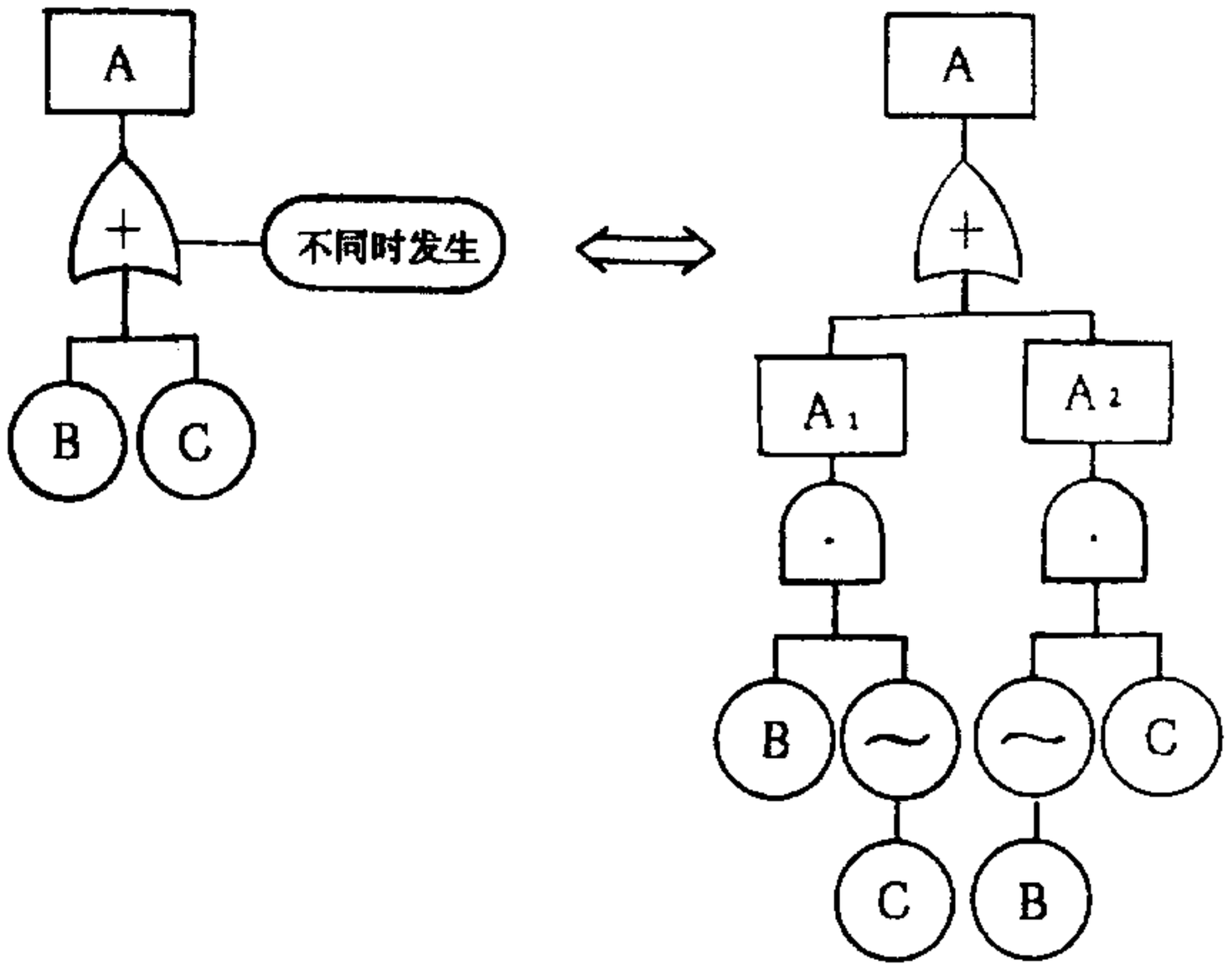


图 5.17　异或门变换为或门、与门和非门的组合

5.2.3.2.4　禁门变换为与门的规则

原输出事件不变，禁门变换为与门，与门之下有两个输入，一个为原输入事件，另一个为禁门打开条件事件。这条规则举例见图 5.18。

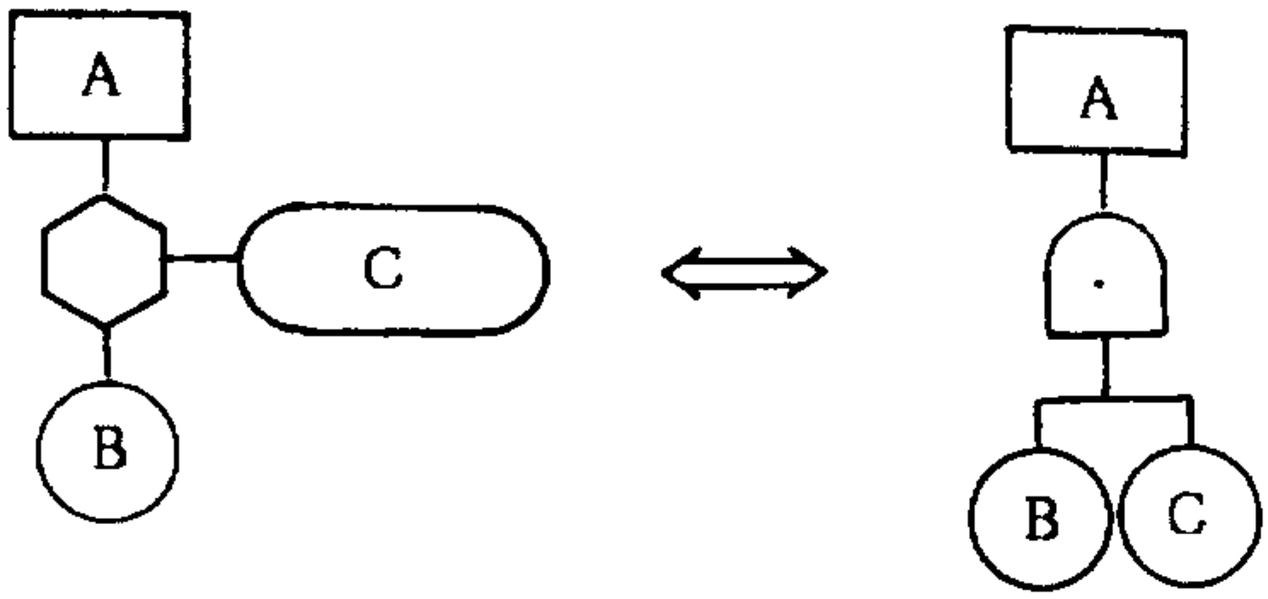


图 5.18　禁门变换为与门

5.2.4　故障树的简化和模块分解

故障树的简化和模块分解并不是故障树分析的必要步骤。对故障树不作简化和模块分解，或简化和模块分解不完全，并不会影响以后定性分析和定量分析的结果。然而，对故障树尽可能的简化和模块分解是减小故障树的规模，从而减少分析工作量的有效措施。

5.2.4.1 故障树的简化

5.2.4.1.1 用相同转移符号表示相同子树，用相似转移符号表示相似子树。

例如，使用相同转移符号将图 5.19 变为图 5.20。

使用相似转移符号将图 5.21 变为图 5.22。

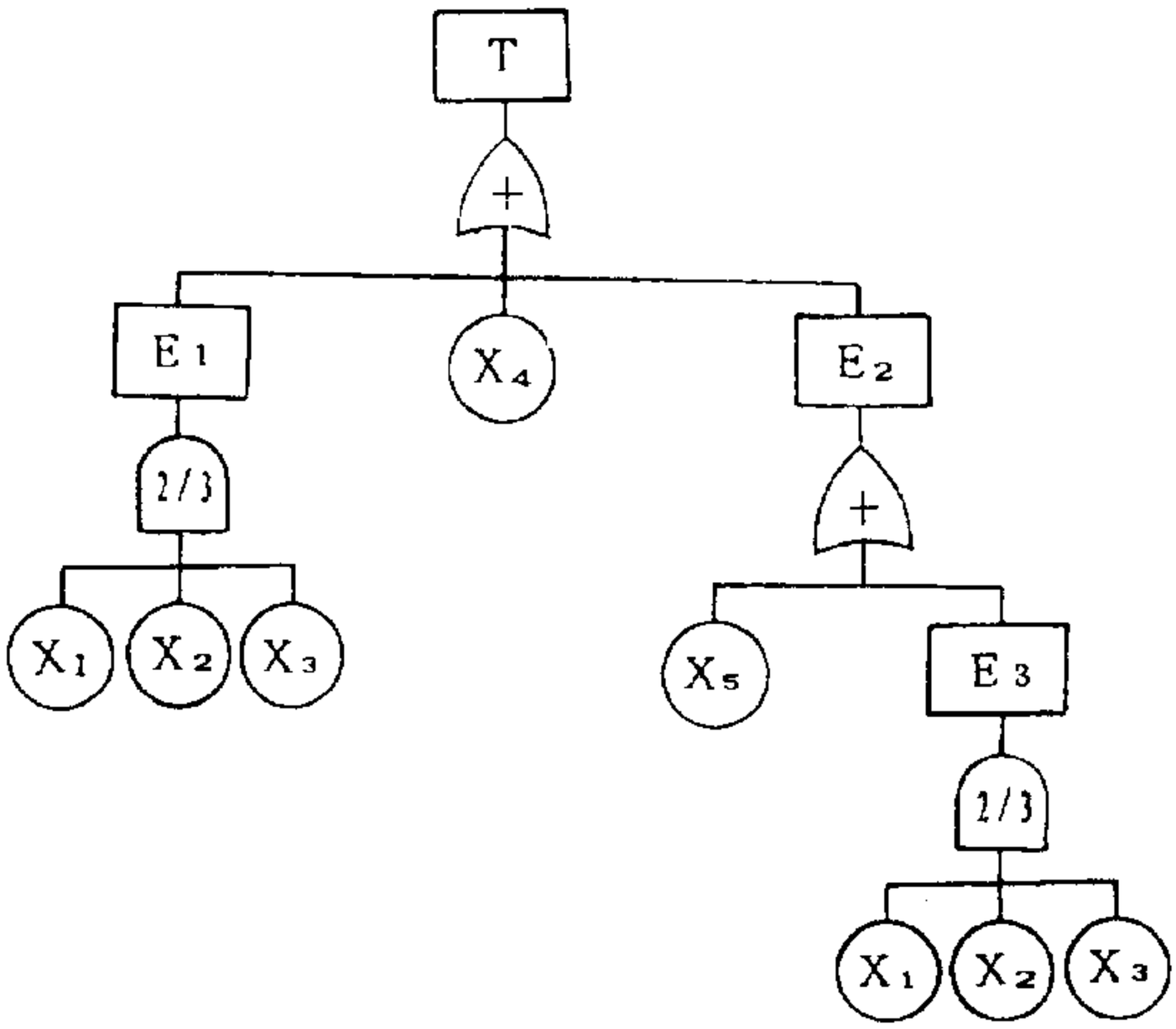


图 5.19

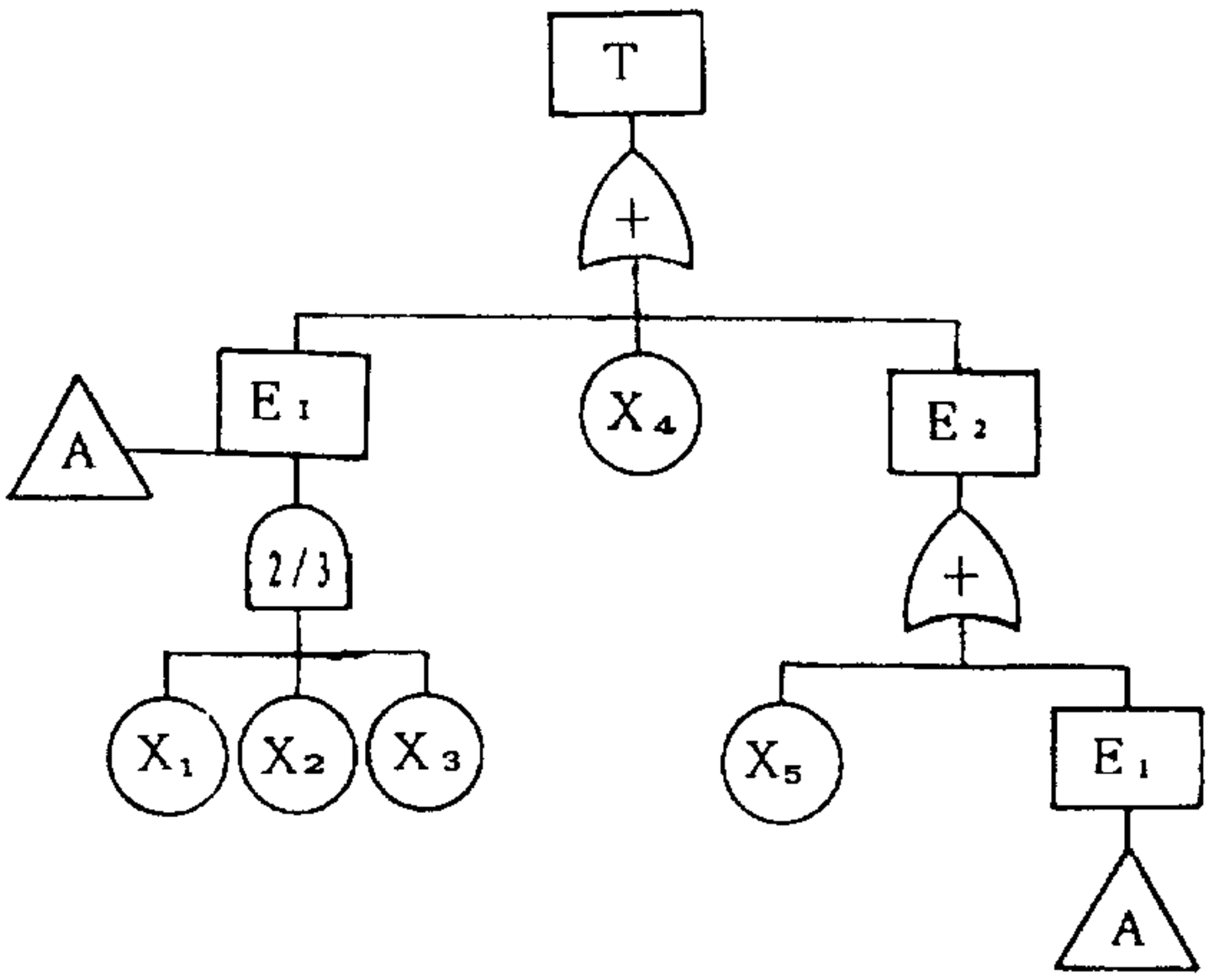


图 5.20

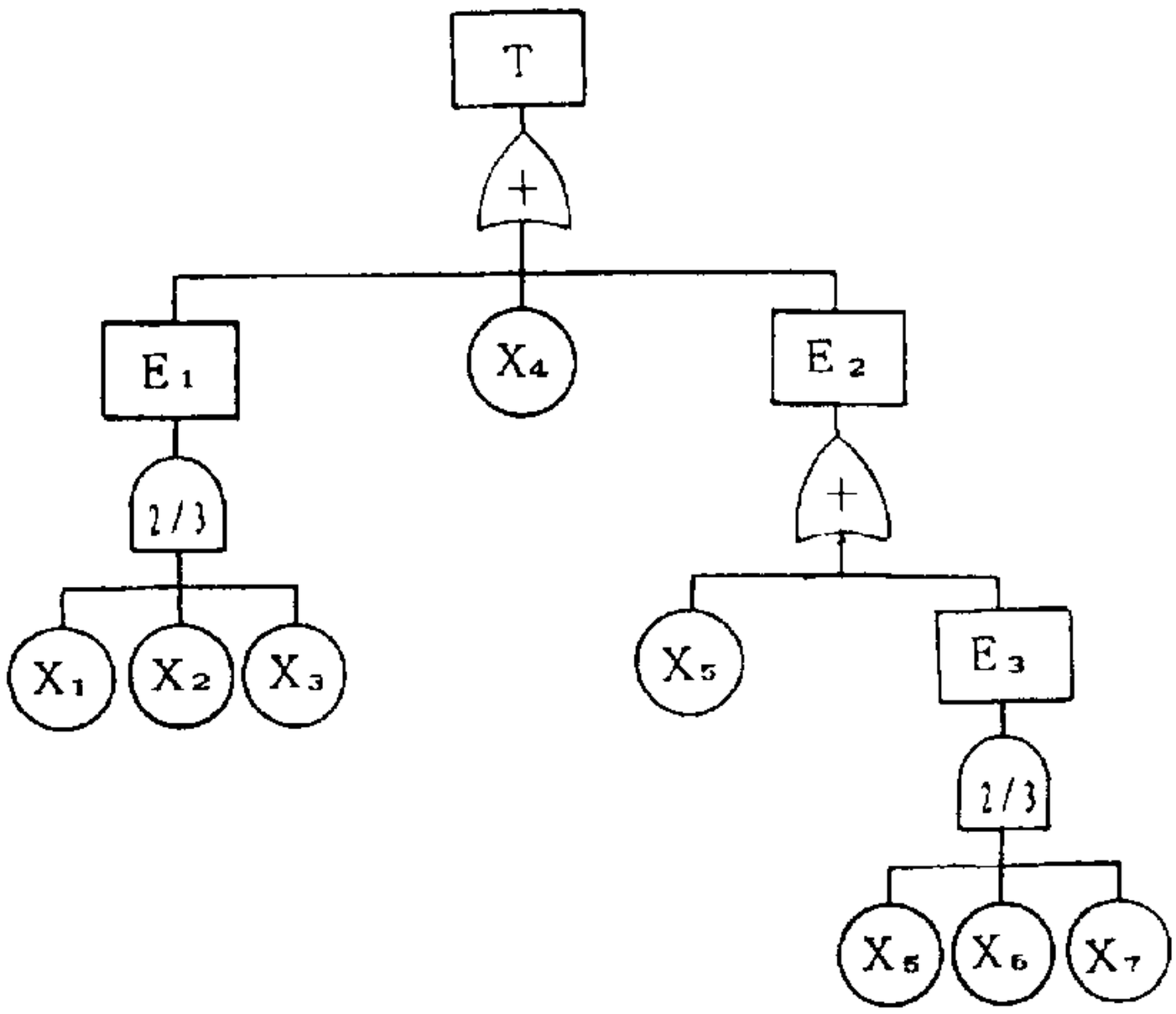
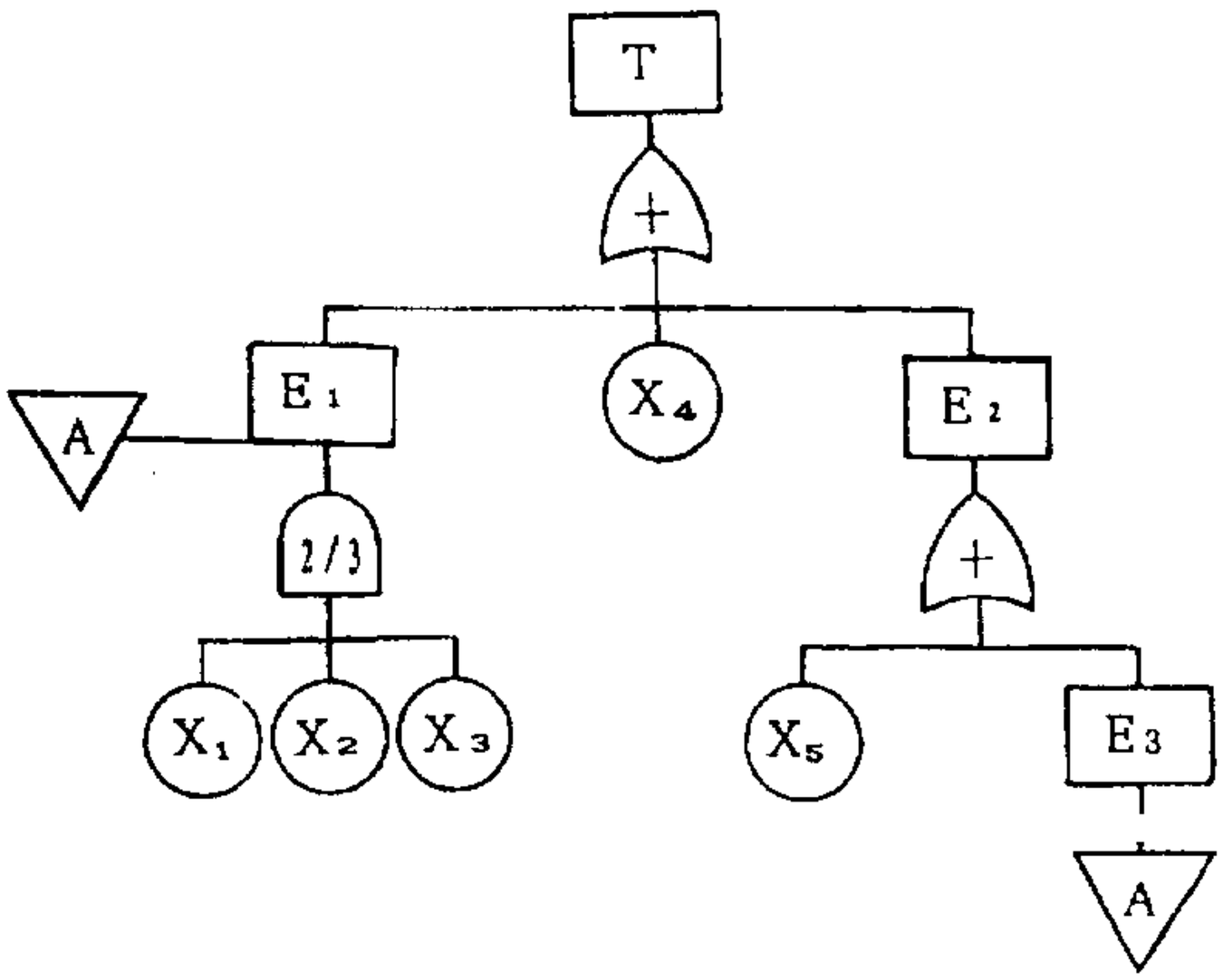


图 5.21



底事件标号：$X_5 \sim X_7$

图 5.22

5.2.4.1.2　去掉明显的逻辑多余事件和明显的逻辑多余门

按照集合(事件)运算规则,可得以下简化故障树的基本原理：

a. 按结合律

$$(A+B)+C=A+B+C$$

可作如图 5.23 的简化：

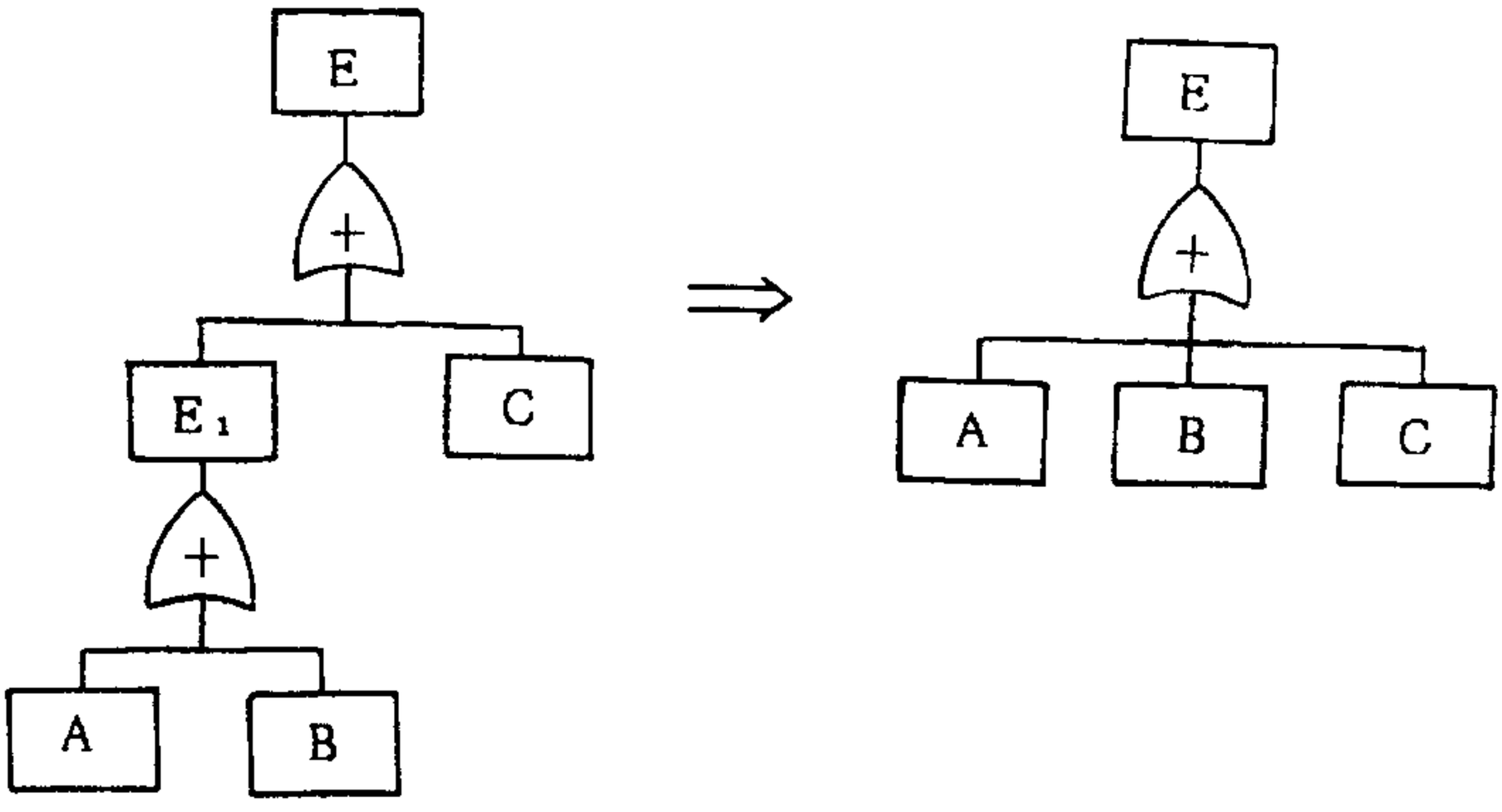


图 5.23

b. 按结合律

$$(AB)C=ABC$$

可作如图 5.24 的简化：

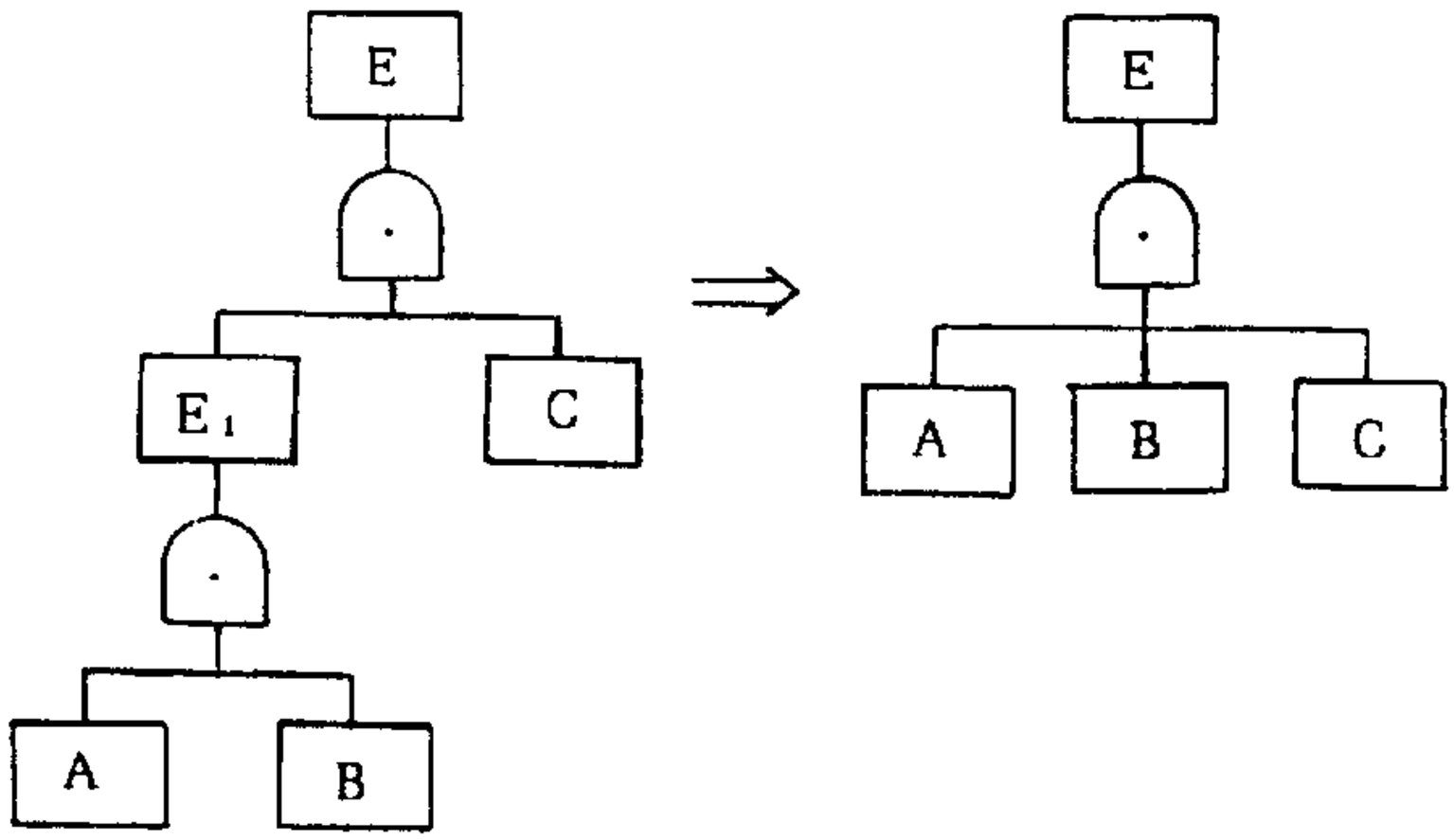


图 5.24

c. 按分配律

$$AB+AC=A(B+C)$$

可作如图 5.25 的简化：

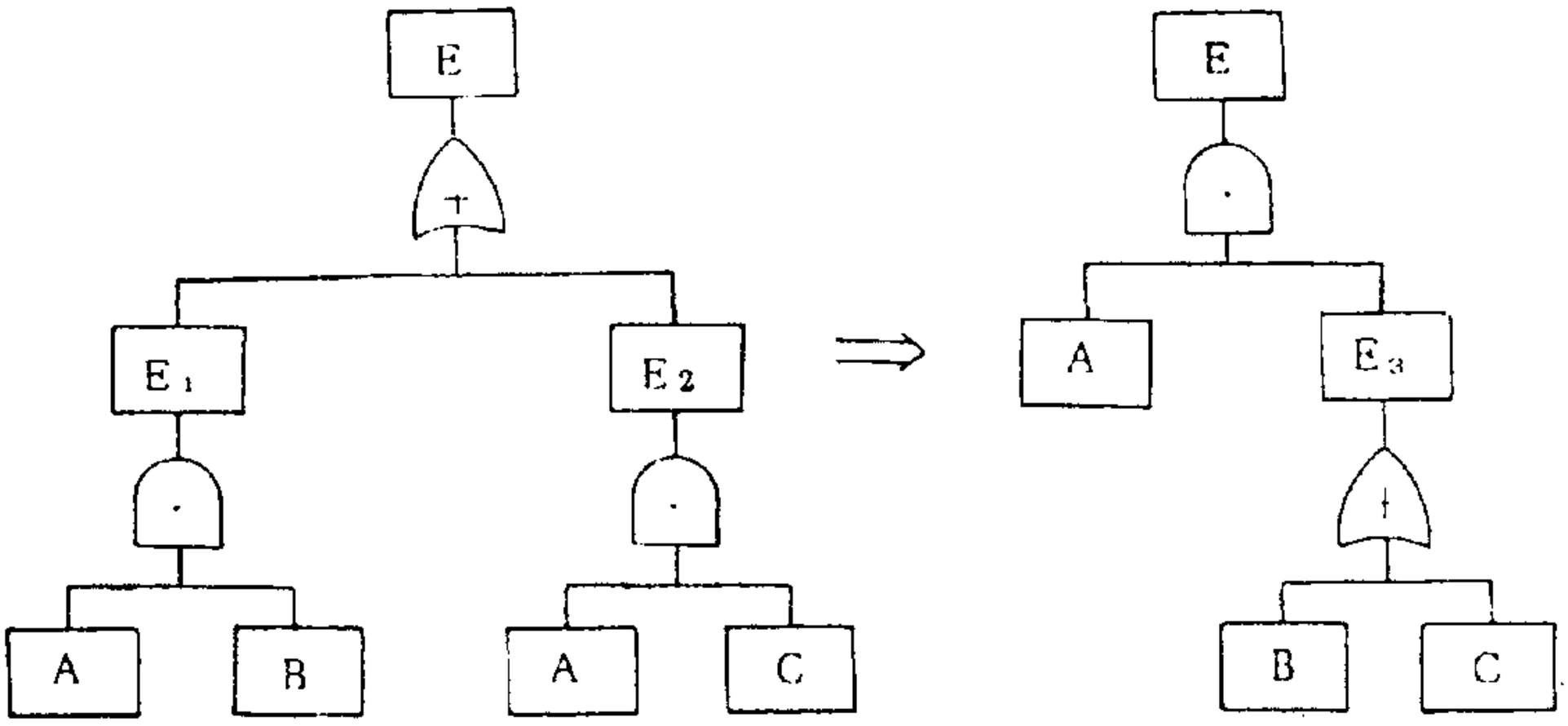


图 5.25

d. 按分配律

$$(A+B)(A+C)=A+BC$$

可作如图 5.26 的简化：

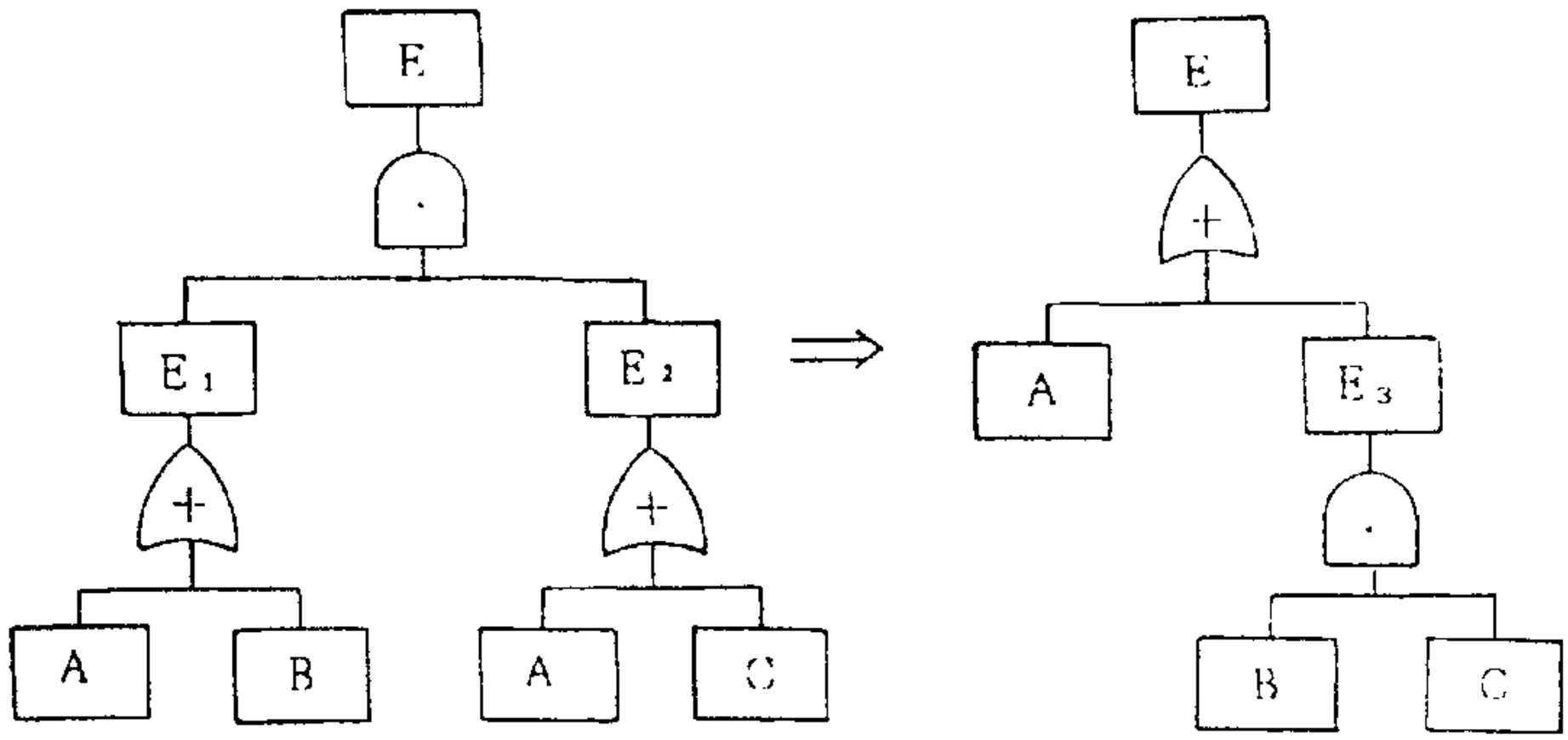


图 5.26

e. 按吸收律

$$A(A+B)=A$$

可作如图 5.27 的简化：

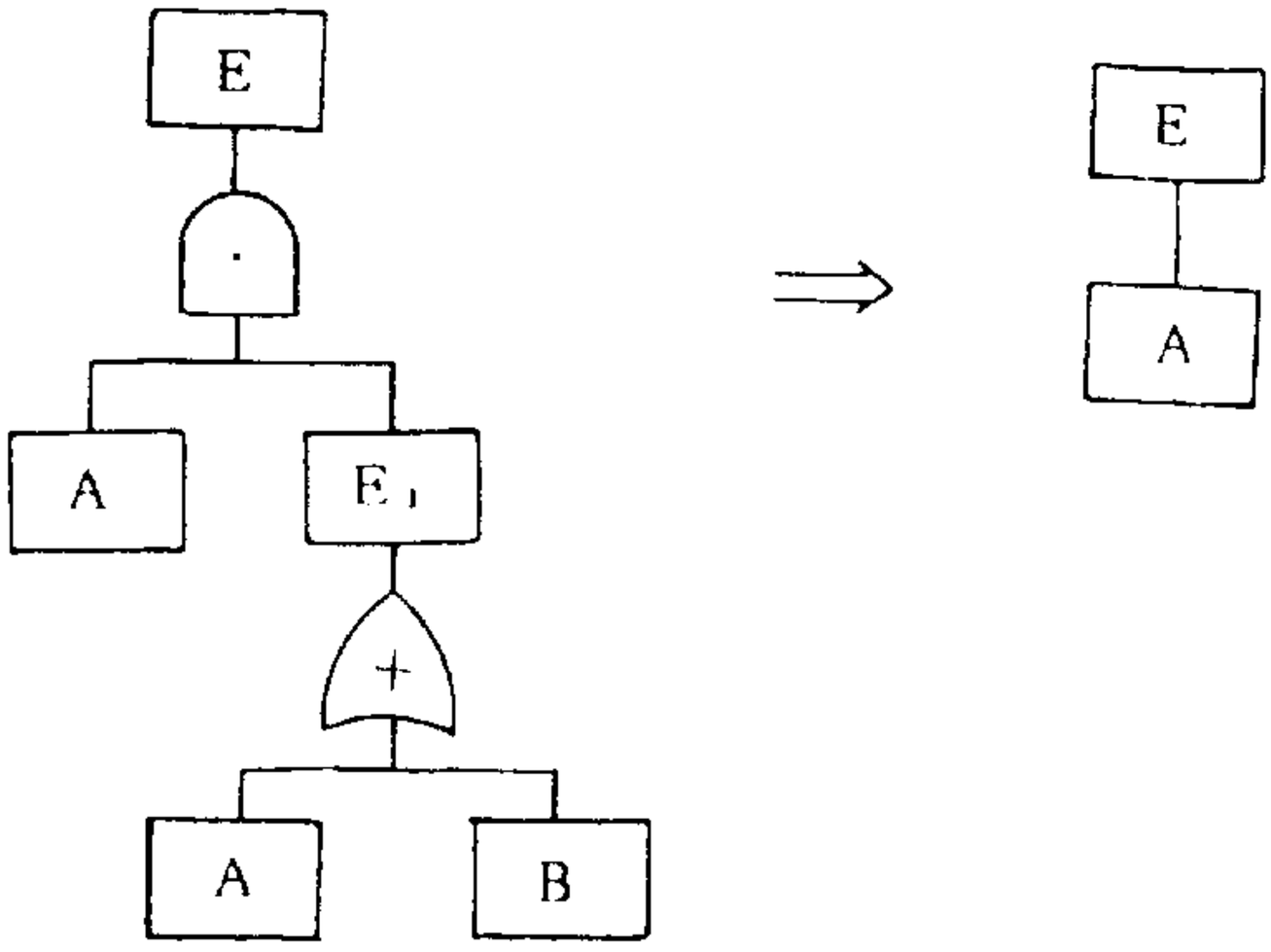


图 5.27

f. 按吸收律

$$A + AB = A$$

可作如图 5.28 的简化：

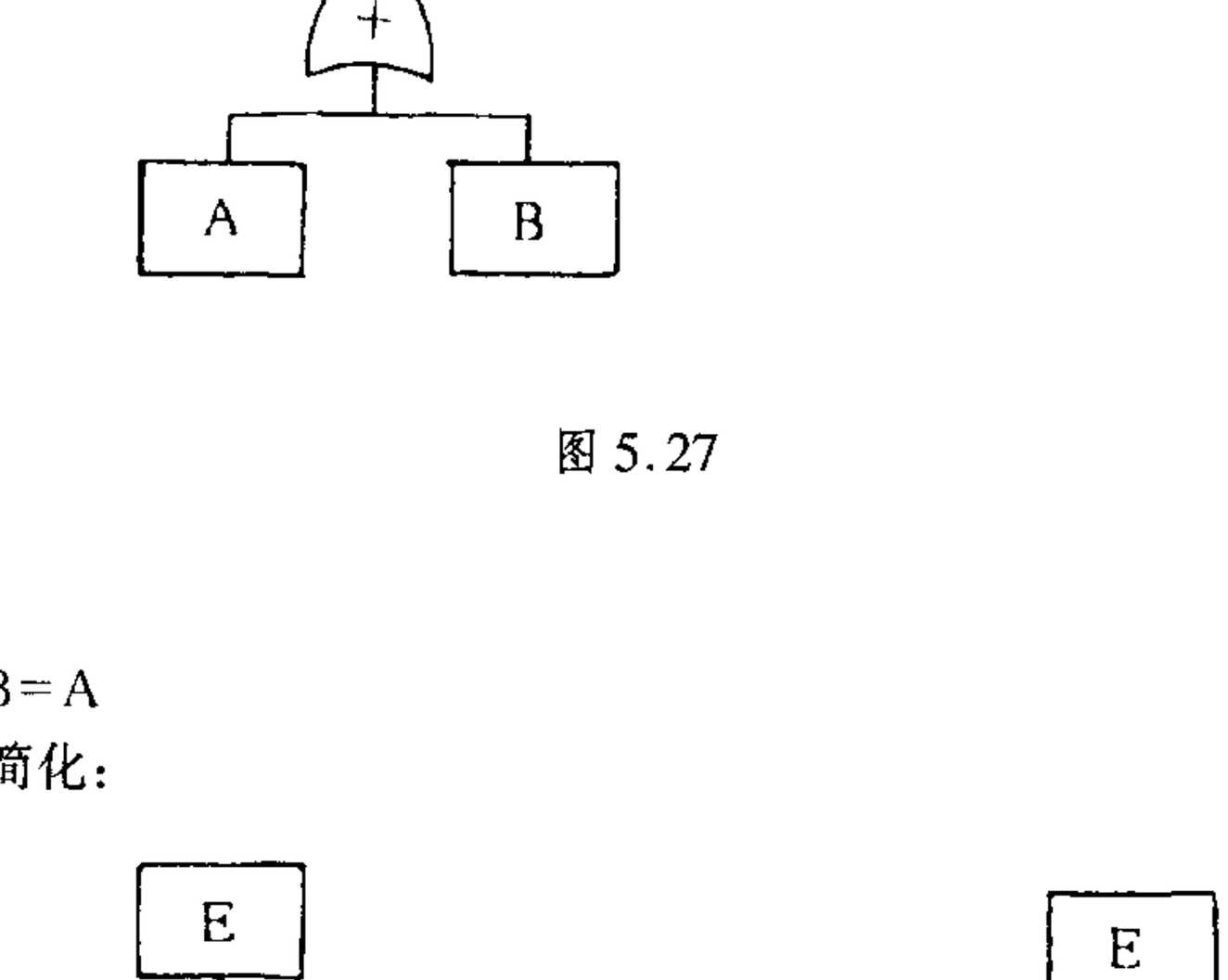


图 5.28

g. 按幂等律

$$A + A = A$$

可作如图 5.29 的简化：

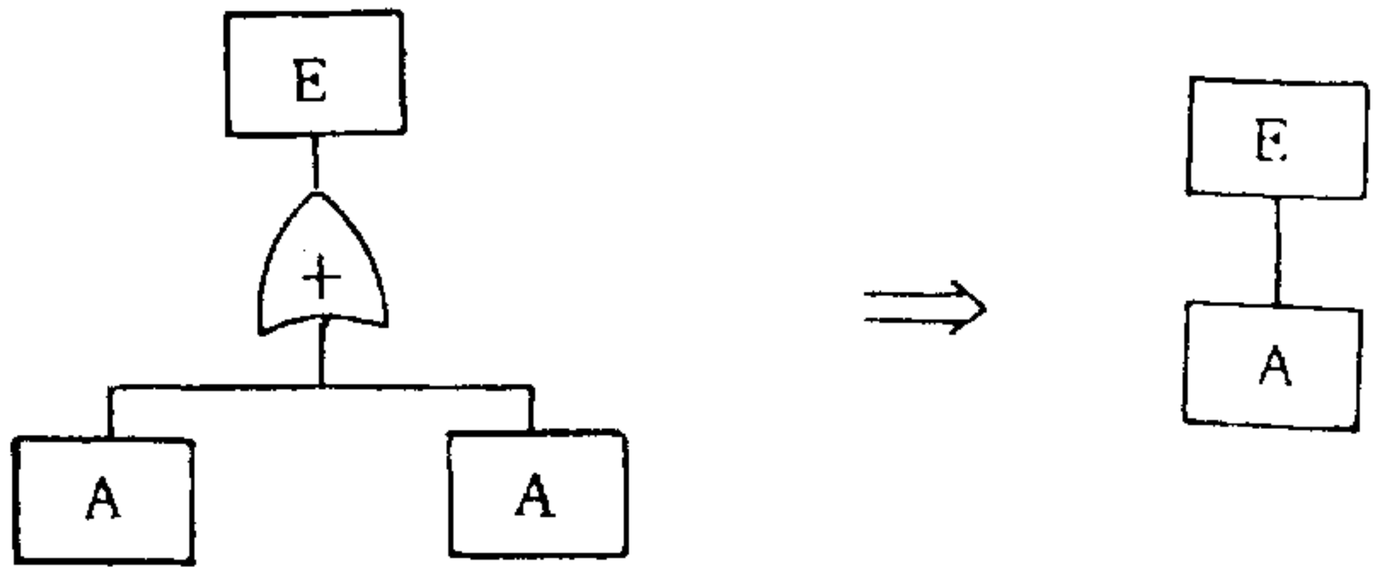


图 5.29

h. 按幂等律

$$AA = A$$

可作如图 5.30 的简化：

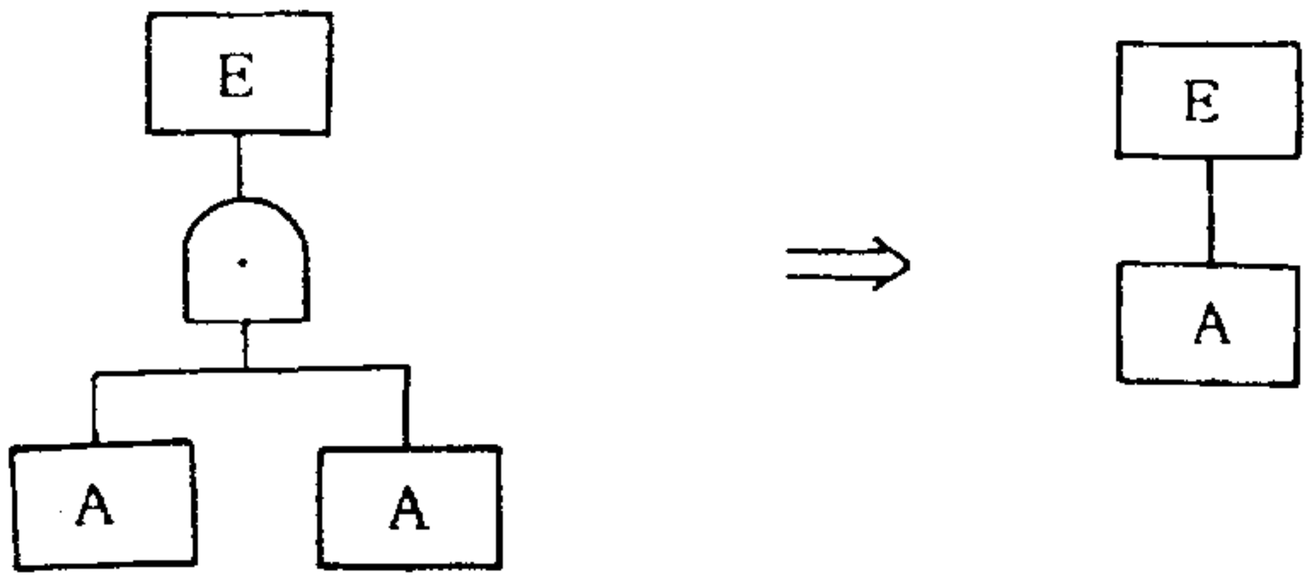


图 5.30

i. 按互补律

$$A\bar{A} = \Phi$$

其中 Φ 为空集，图 5.31 中事件 E 是不可能发生的事件，因此事件 E 以下的部分可以全部删去。

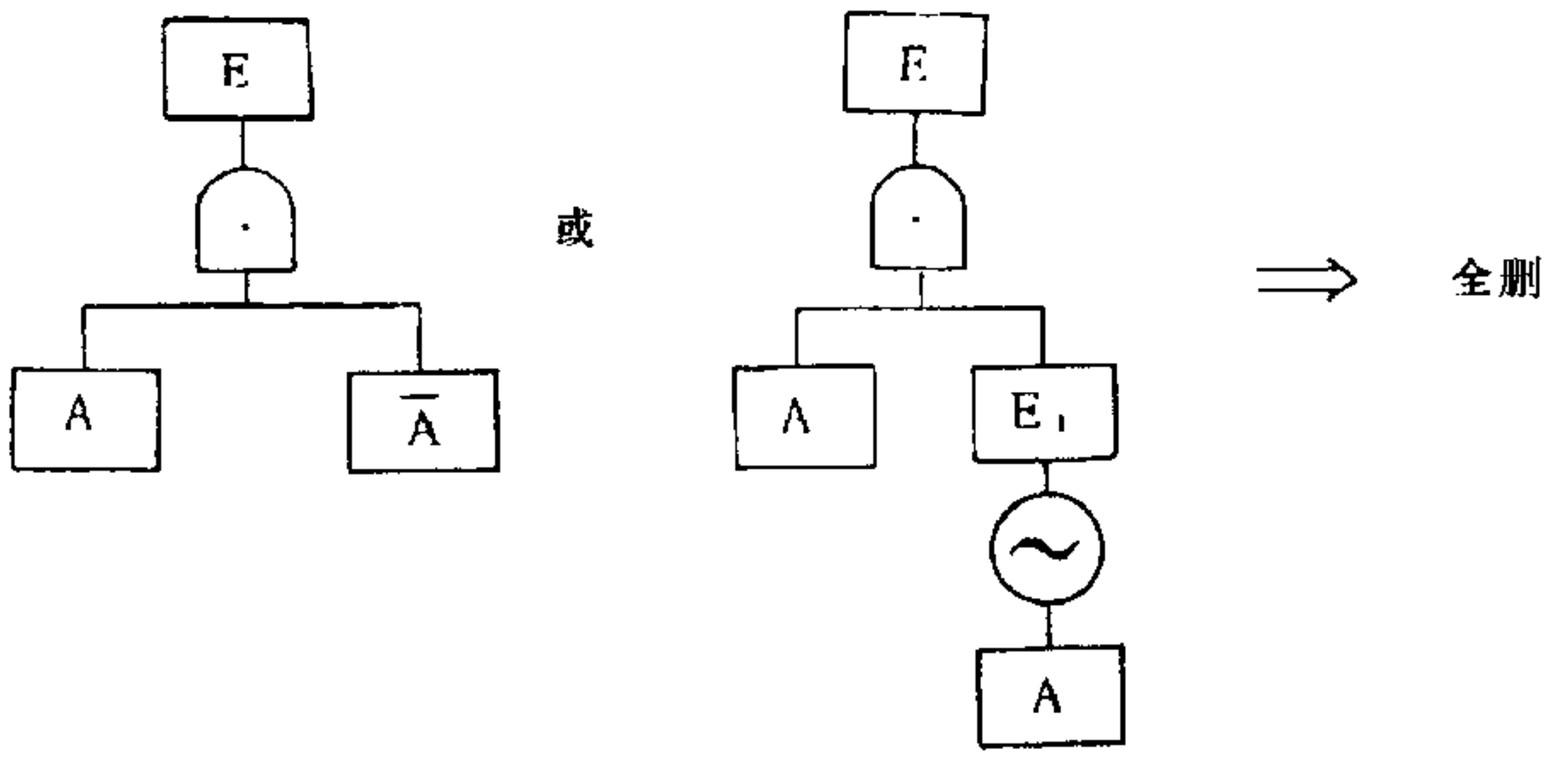


图 5.31

下面给出一个简化故障树的示例。

图 5.32 给出了包括逻辑多余部分的故障树。

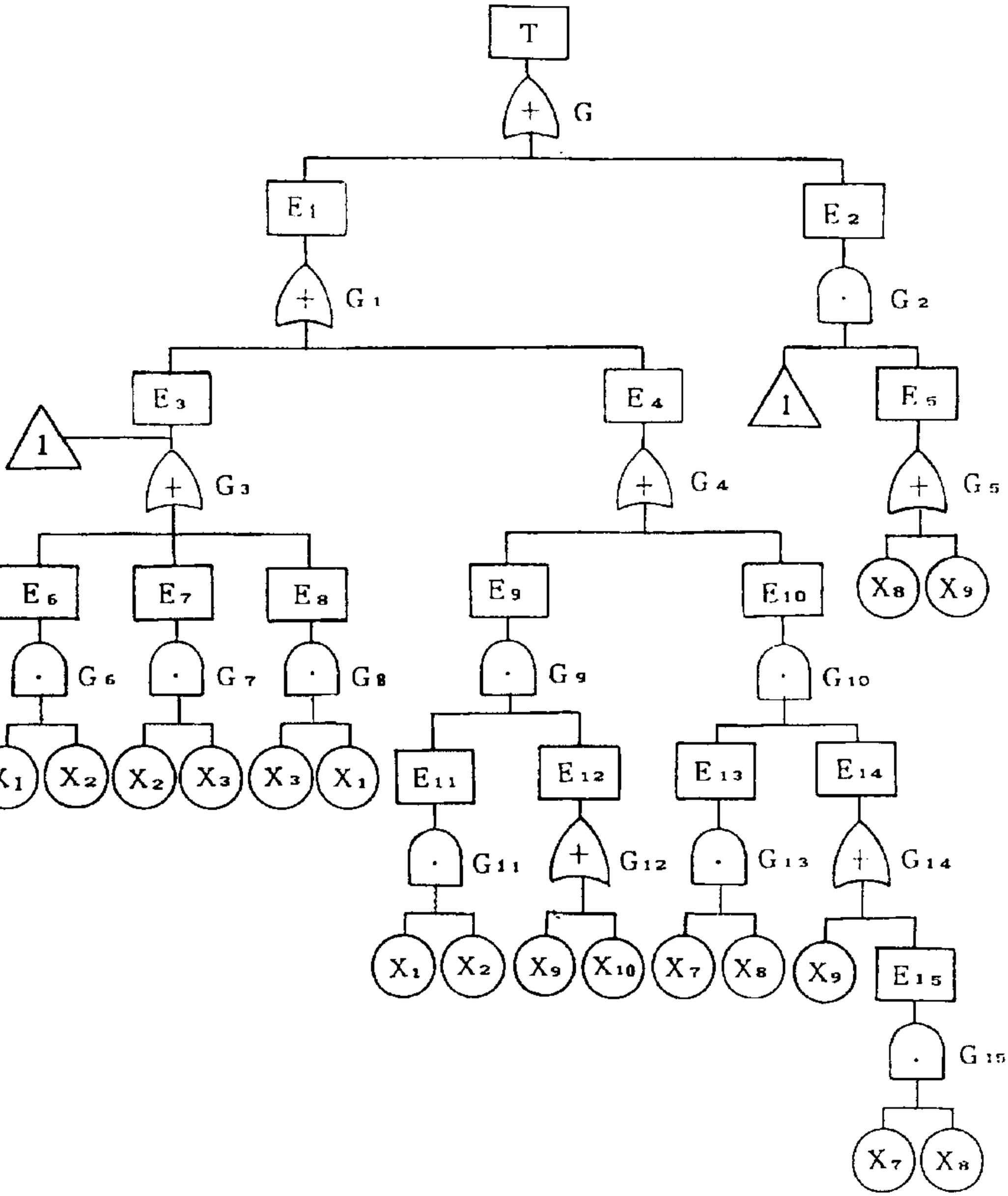


图 5.32 含逻辑多余部分的故障树

E_9 和 E_6 通过一系列或门向上到达或门 G_1，按图 5.23，E_9 和 E_6 可简化成 G_1 的直接输入；又因为 E_6 和 E_{11} 是相同事件，而 G_1 是或门、G_9 是与门，故按图 5.28，E_9 以下可以全部删去。按图 5.23，E_2 和 E_3 可简化成或门 G 的直接输入，注意到相同转移符号，故按图 5.28，E_2 以下可以全部删去。注意到 E_{13} 和 E_{15} 是相同事件，按图 5.27，E_{14} 以下可以全部删去，最后，再按图

5.23 和图 5.24,图 5.32 简化为图 5.33。

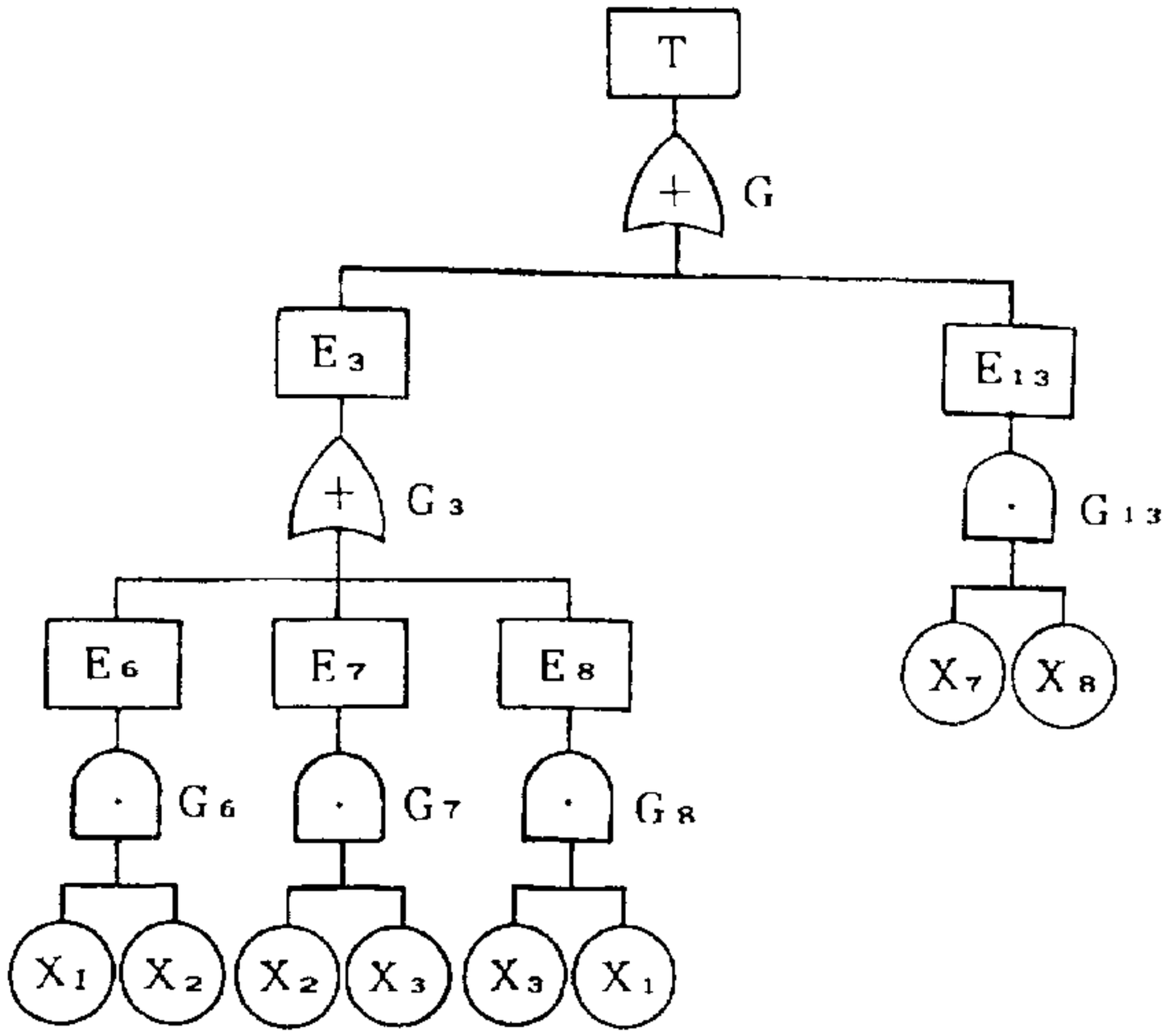


图 5.33

通过上述例子,可看出上述基本原理在故障树简化中的具体运用。

5.2.4.2 故障树的模块分解

故障树的模块分解按下述步骤进行:

a. 按模块和最大模块的定义(见 3.11 条),找出故障树中尽可能大的模块。如果有计算机软件可用的话,求出故障树的所有最大模块;

b. 每个模块构成一个模块子树,可单独地进行定性分析和定量分析;

c. 对每个模块子树用一个等效的虚设底事件来代替,使原故障树的规模减小;

d. 在故障树定性分析和定量分析后,可根据实际需要,将顶事件与各模块之间的关系,转换为顶事件与底事件之间的关系。

下面给出一个故障树模块分解的示例。

对图 5.34 的故障树用相同转移符号简化后,可得图 5.35 的故障树。

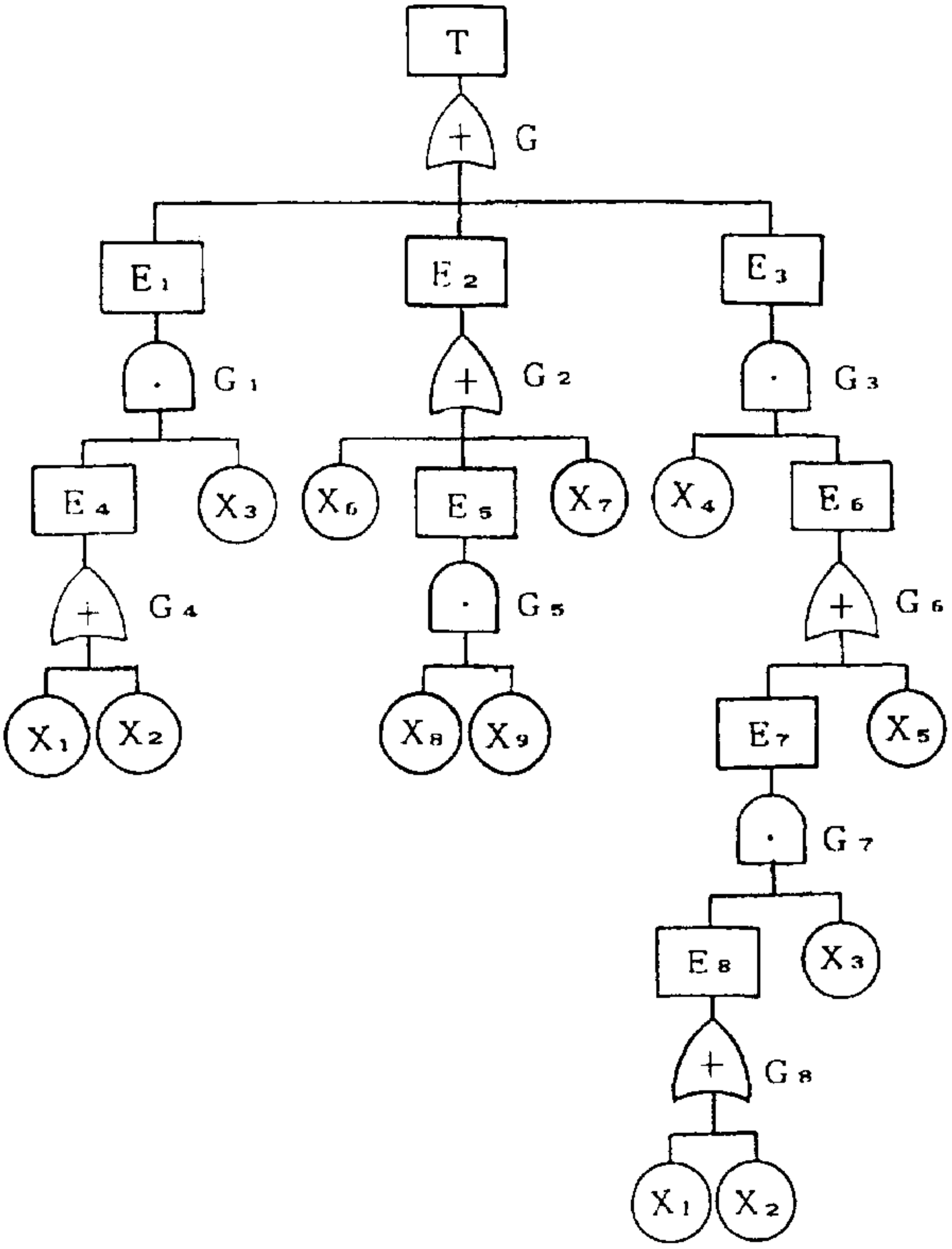

T
+
G
E1
E2
E3
·
G1
+
G2
·
G3
E4
X3
X6
E5
X7
X4
E6
+
G4
·
G5
+
G6
X1
X2
X8
X9
E7
X5
·
G7
E8
X3
+
G8
X1
X2

图 5.34

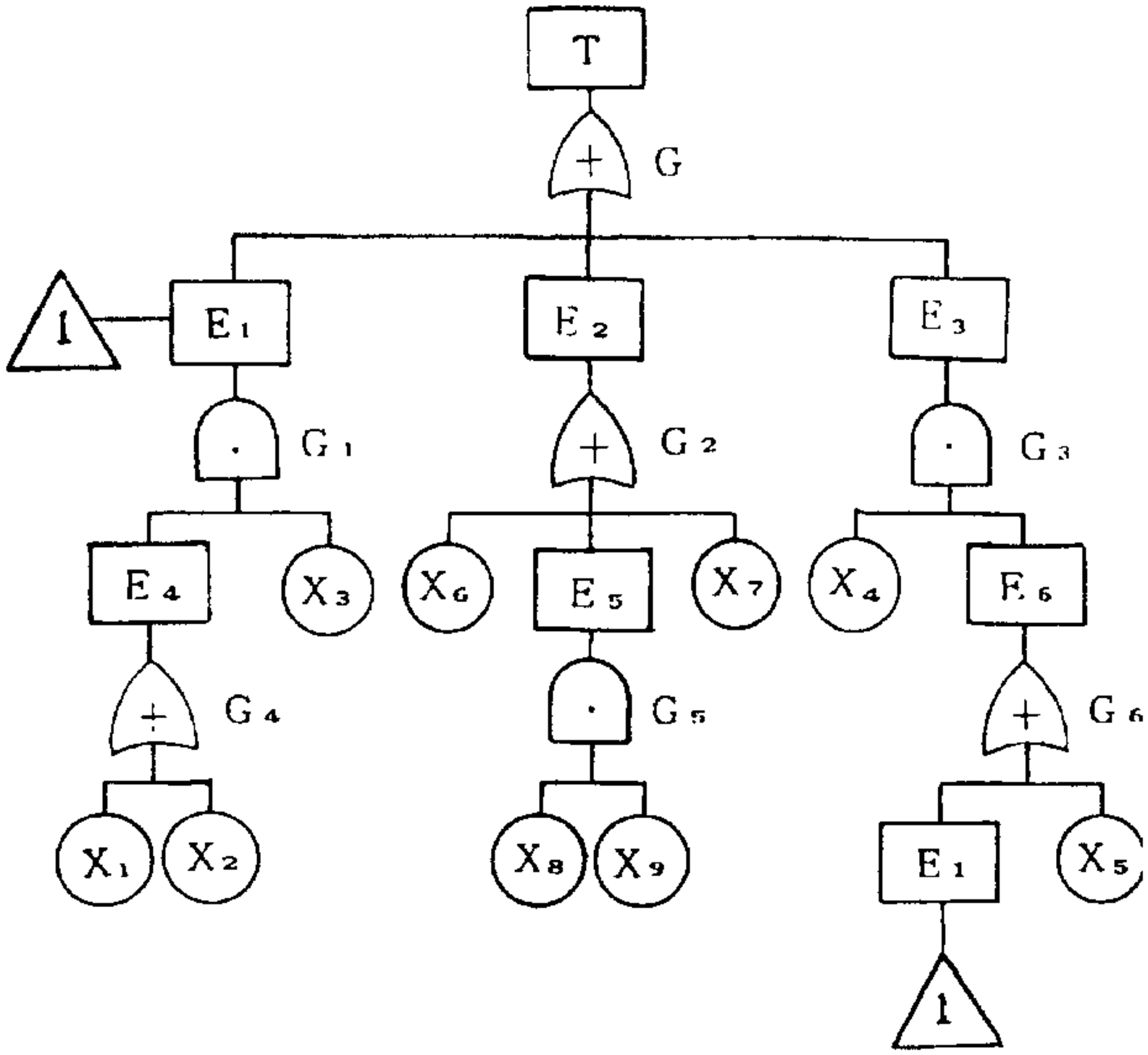


图 5.35

按 3.11 条中故障树模块和最大模块的定义，在图 5.35 中底事件 X_1, X_2, X_3 向上到达同一逻辑门 G_1 才能到达顶事件，故障树所有其它底事件向上均不能到达 G_1。因此，底事件集合 $\{X_1, X_2, X_3\}$ 为故障树的一个模块。同样，底事件集合 $\{X_6, X_7, X_8, X_9\}$ 也为故障树的一个模块。底事件 $\{X_1, X_2\}$ 和 $\{X_8, X_9\}$ 也是故障树的模块，但它们不是最大模块。进一步由定义可验证，$\{X_1, X_2, X_3\}$ 和 $\{X_6, X_7, X_8, X_9\}$ 为故障树的最大模块。故 E_1 以下构成一个模块子树，E_2 以下也构成一个模块子树。

在图 5.36 中用相同转移符号表示事件 E_1 和事件 E_2。我们可单独地对它们进行定性分析和定量分析。对这两个模块子树 E_1 和 E_2 可看成两个虚设底事件，使原故障树的规模减小，以节省分析工作量。

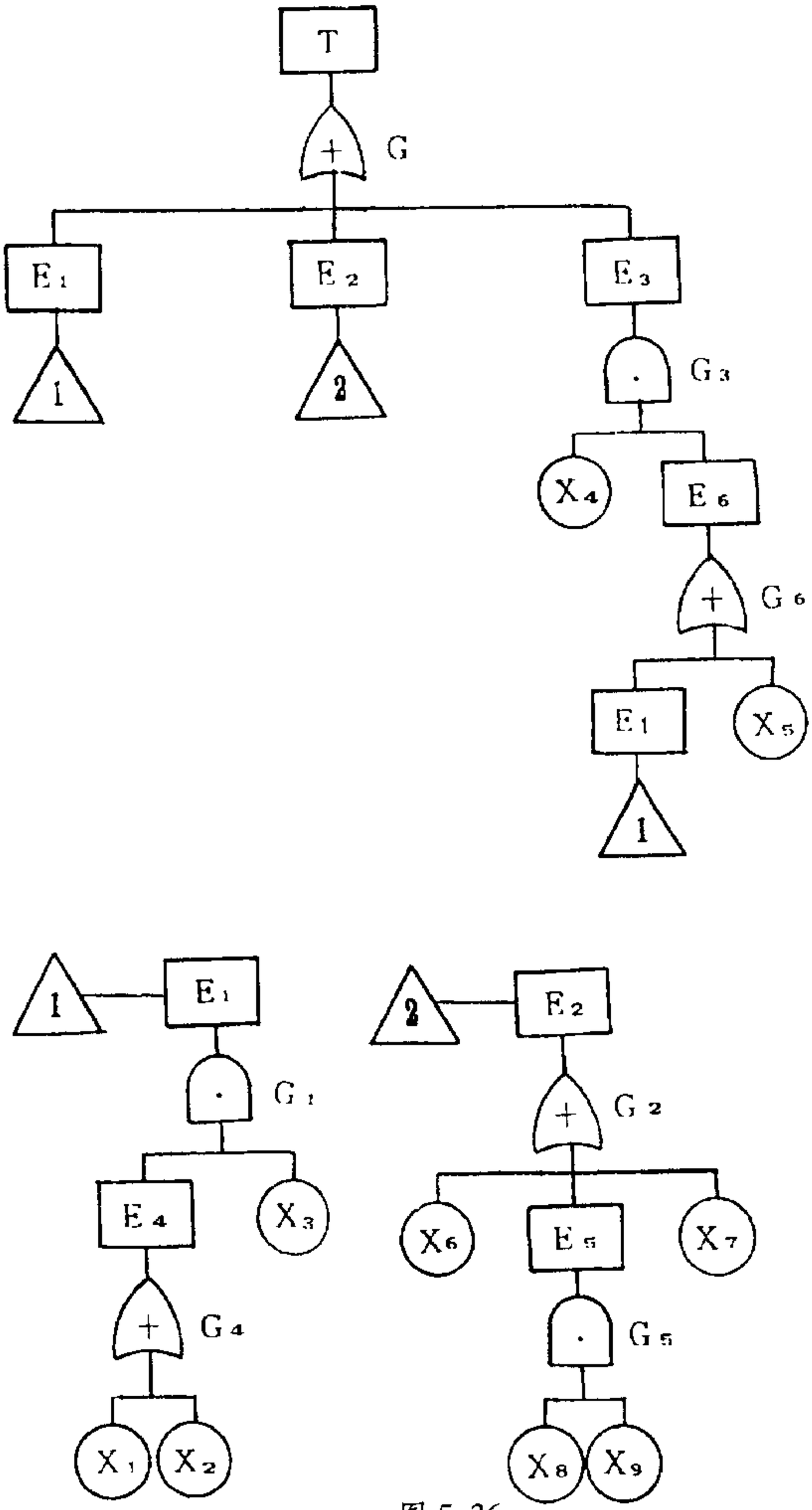


图 5.36

5.3 单调故障树定性分析

5.3.1 目的

故障树定性分析的目的在于寻找导致顶事件发生的原因和原因组合，即识别导致顶事件发生的所有故障模式，它可以帮助判明潜在的故障，以便改进设计；可以用于指导故障诊断，改进使用和维修方案。

5.3.2 定性分析指南

单调故障树的(系统)故障模式通过最小割集表示。单调故障树定性分析的基本任务在于找出故障树的所有最小割集。对于给定的单调故障树,由所有最小割集组成的最小割集族是唯一确定的。

5.3.3 步骤

5.3.3.1 预备步骤

对于已经建造的故障树,应当进行规范化、简化和模块分解。

5.3.3.2 求最小割集

根据故障树结构,用下行法或上行法求故障树的所有最小割集。

5.3.3.3 把“模块最小割集”转换为“底事件最小割集”

对于已经模块化的故障树求得的一般是“模块最小割集”。它是若干“底事件最小割集”的代表,数量集中,便于掌握。但为了进行底事件重要度的定性比较和定量计算,还应当把“模块最小割集”转换为“底事件最小割集”。示例见 5.3.6.4。

5.3.3.4 用最小割集表示故障树顶事件

在求得全部最小割集 $C_1, C_2, \cdots, C_r$ 的基础上,可将故障树的顶事件表示为

$$T = \sum_{j=1}^{r} \prod_{X_i \in C_j} X_i \qquad (12)$$

式中:$\sum$ ——并集;

$\prod$ ——交集;

X_i ——第 i 个底事件。

为了节省分析工作量,在工程上可以略去阶数(最小割集中所含底事件的个数)大于指定值的所有最小割集来进行近似分析。

5.3.4 方法

5.3.4.1 下行法

这个方法的特点是根据故障树的实际结构,从顶事件开始,逐级向下寻查,找出割集。规定在下行过程中,顺次将逻辑门的输出事件置换为输入事件。遇到与门就将其输入事件排在同一行(取输入事件的交(布尔积)),遇到或门就将其输入事件各自排成一行(取输入事件的并(布尔和)),这样直到全部换成底事件为止,这样得到的割集再通过两两比较,划去那些非最小割集,剩下即为故障树的全部最小割集。

下行法求所有最小割集的示例见 5.3.6.1 条。

5.3.4.2 上行法

上行法是从底事件开始,自下而上逐步地进行事件集合运算,将或门输出事件表示为输入事件的并(布尔和),将与门输出事件表示为输入事件的交(布尔积)。这样向上层层代入,在逐步代入过程中或者最后,按照布尔代数吸收律和等幂律来化简,将顶事件表示成底事件积之和的最简式。其中每一积项对应于故障树的一个最小割集,全部积项即是故障树的所有最小割集。

上行法求最小割集的示例见 5.3.6.2 条。

5.3.5 定性分析结果的应用

故障树定性分析的基本结果是求得的全部最小割集。它的基本用途在于识别导致顶事件发生的所有可能的系统故障模式，这种基于严格逻辑演绎求得的所有故障模式和根据系统故障履历或者个人经验所得到的认识有原则性差别：后者限于事后经验，前者可以事前推理；后者可能有所遗漏，前者在原则上可以保证完整性。因而有助于判明潜在的故障，避免遗漏重要的故障模式；有助于指导故障诊断和制订使用维修方案，故障树定性分析结果也是进一步定量分析的基础。

如果数据不足，则进行定性比较：根据每个最小割集阶数排序。在各个底事件发生概率比较小、其差别相对地不大的条件下，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；在考虑最小割集阶数的条件下，在不同最小割集中重复出现次数越多的底事件越重要。定性比较结果可用于指导故障诊断、确定维修次序、或者指示改进系统的方向。

5.3.6 输变电网络故障树定性分析示例

输变电网络见图 5.3 所示。系统的故障树见图 5.13 所示。经规范化整理后的故障树见图 5.37 所示。前图中的“三中取二”表决门已经变换为后图中 E4 以下子树，后图中各事件符号代表意义如下。

E_1——电网失效；

E_2——B 站无输入；

E_3——C 站无输入；

E_4——站 B 和站 C 仅由同一单线供电；

E_5——来自站 C 的输电线路无电；

E_6——来自站 B 的输电线路无电；

E_7——输电线 2、3 同时故障；

E_8——输电线 1、3 同时故障；

E_9——输电线 1、2 同时故障；

E_{10}——输电线 4、5 同时故障；

5.3.6.1 用下行法找所有最小割集

步骤 1　顶事件 E_1 下面是或门，将其输入事件 E_2，E_3，E_4 各自成一行；

步骤 2　事件 E_2 下面是与门，将其输入 X_1、X_2、E_5 排在同一行；事件 E_3 下面是与门，将其输入 X_3、E_6 排成另一行；事件 E_4 下面是或门，将其输入 E_7、E_8、E_9 各自排成一行；

步骤 3　事件 E_5 下面是或门，将其输入 X_3、E_{10} 各自排成一行并分别与 X_1、X_2 组合成为 X_1、X_2、X_3；X_1、X_2、E_{10}；

事件 E_6 下面是或门，将其输入 E_{10}、E_9 各自排一行并分别与 X_3 组合成为 X_3、E_{10}；X_3、E_9；

事件 E_7 下面是与门，将其输入 X_2、X_3 写成同一行。

事件 E_6 下面是与门，将其输入 X_1、X_3 写成同一行。

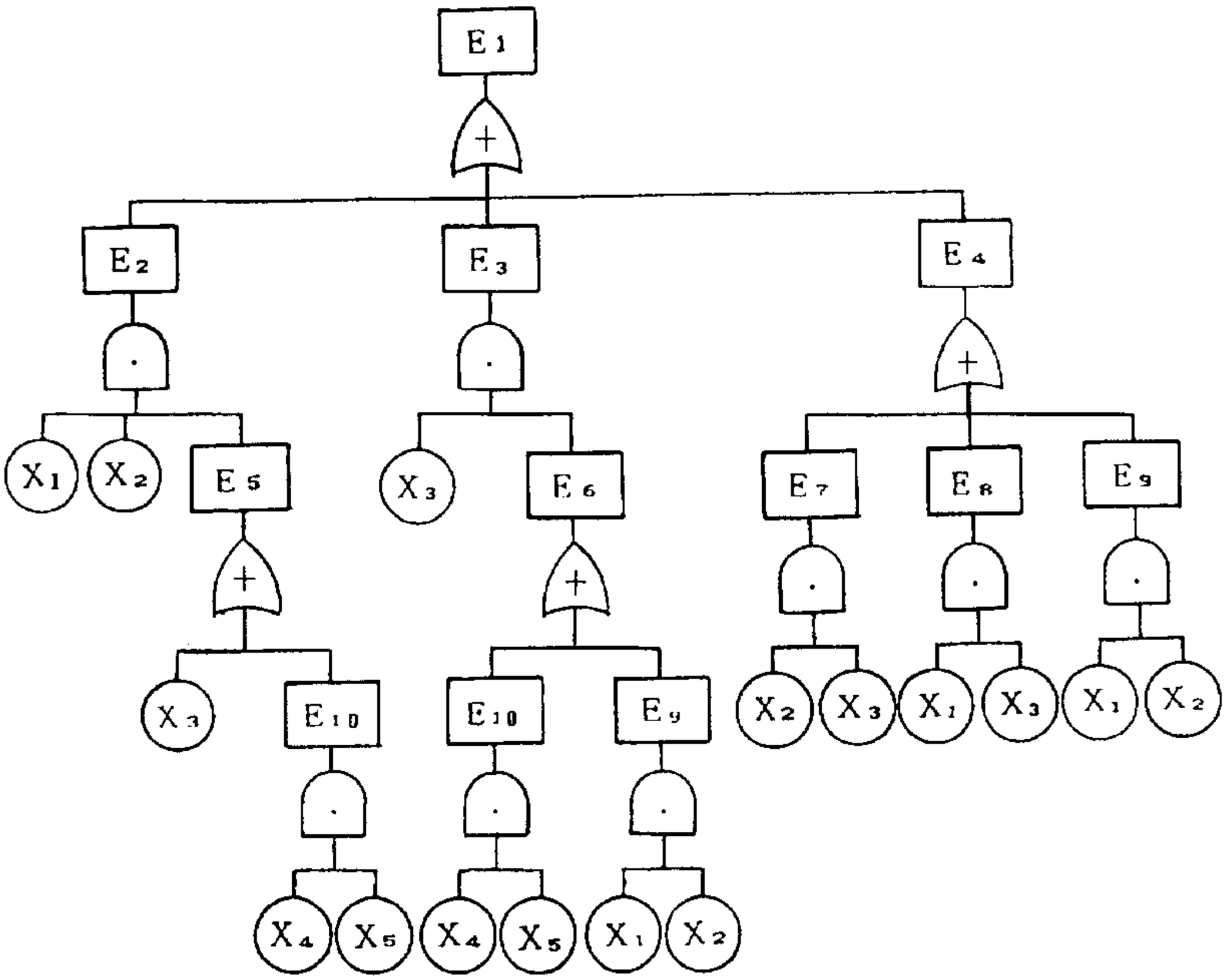


图 5.37　输变电网络规范化故障树

事件 E_9 下面是与门，将其输入 X_1、X_2 写成同一行。

步骤 4　事件 E_{10} 下面是与门，将其输入 X_4、X_5 写成同一行、并与 X_1、X_2 组合成 X_1、X_2、X_4、X_5；

将 E_{10} 下面与门输入 X_4、X_5 和 X_3 组合成 X_3、X_4、X_5；

将 E_9 下面与门输入 X_1、X_2 和 X_3 组合成 X_3、X_1、X_2；至此，故障树的所有逻辑门的输出事件都已被处理，步骤 4 所得到的每一行都是一个割集，共得七个割集。

步骤 5　进行两两比较：

因为$\{X_1、X_2\}$是割集，所以$\{X_1、X_2、X_3\}$，$\{X_1、X_2、X_4、X_5\}$和$\{X_3、X_1、X_2\}$都不是最小割集，应当删去，所以最后求得全部最小割集 4 个：

$\{X_3、X_4、X_5\}$、$\{X_2、X_3\}$、$\{X_1、X_3\}$、$\{X_1、X_2\}$。

上述步骤可表示为：

	步骤 1	步骤 2	步骤 3	步骤 4	步骤 5
$E_1\rightarrow$	$E_2\rightarrow$	$X_1、X_2、E_5\rightarrow$	$X_1、X_2、X_3\rightarrow$	$X_1、X_2、X_3\rightarrow$	$X_3、X_4、X_5$
	E_3	$X_3、E_6$	$X_1、X_2、E_{10}$	$X_1、X_2、X_4、X_5$	$X_2、X_3$
	E_4	E_7	$X_3、E_{10}$	$X_3、X_4、X_5$	$X_1、X_3$
		E_8	$X_3、E_9$	$X_3、X_1、X_2$	$X_1、X_2$
		E_9	$X_2、X_3$	$X_2、X_3$	
		$X_1、X_3$	$X_1、X_3$		
		$X_1、X_2$	$X_1、X_2$		

故障树顶事件可表示为：

$$T=E_1=X_3X_4X_5+X_1X_3+X_1X_2+X_2X_3$$

5.3.6.2　用上行法求所有最小割集

$E_{10}=X_4X_5$

$E_9=X_1X_2$

$E_8=X_1X_3$

$E_7=X_2X_3$

$E_6=E_9+E_{10}=X_1X_2+X_4X_5$

$E_5=X_3+E_{10}=X_3+X_4X_5$

$E_4=E_7+E_8+E_9=X_2X_3+X_1X_3+X_1X_2$

$E_3=X_3E_6=X_3X_1X_2+X_3X_4X_5$

$E_2=X_1X_2E_5=X_1X_2X_3+X_1X_2X_4X_5$

$T=E_1=E_2+E_3+E_4=X_1X_2X_3+X_1X_2X_4X_5+X_3X_1X_2+X_3X_4X_5+X_2X_3+X_1X_3+X_1X_2$

$=X_3X_4X_5+X_2X_3+X_1X_3+X_1X_2$

最后得到与下行法相同的 4 个最小割集和故障树顶事件表示式。

5.3.6.3　定性比较

以上得到图 5.37 故障树的四个最小割集代表系统的四种故障模式，其中有三个最小割集的阶数为 2、一个最小割集的阶数为 3。因为根据现有数据还不足以推断各条线路的故障概率值，所以不能做进一步的定量分析，此时应作以下定性比较：

三个 2 阶最小割集的重要性较大，一个 3 阶最小割集的重要性较小；

从单元重要性来看，线路 3 最重要，因为 X_3 在三个最小割集中出现；线路 1、2 的重要性次之，因为 X_1、X_2 在二个最小割集中出现；线路 4、5 的重要性最小，因为 X_4、X_5 只在一个三阶最小割集中出现。

根据这些定性分析结果可知：

a. 如果仅知输变电网络出了故障，原因待查，那么首先应检查线路 3、再检查线路 1 和 2，最后检查线路 4 和 5；

如果已知网络状态是 B 站不能向负荷供电，而 C 站仍能供电，那么根据图 5.37 故障树结

构,不经检查可以判定线路 1、2、4、5 都出了故障,修理次序应先修线路 1 或 2,后修其他;

如果 C 站不能供电而 B 站仍能供电,则从故障树可以判定线路 3、4、5 出了故障。此时修理的顺序,应当先修线路 3,后修线路 4、5。

这样,故障树定性分析结果可以指导故障诊断,并有助于制订维修方案和确定维修次序。

b. 为了改进系统,从上述定性分析结果可以得到重要启示:提高系统可靠性的关键在于提高三个二阶最小割集的阶数和加强对于线路 3 的备份。因此 A、C 站之间应增设备用线路 6,如图 5.38 所示。它可以同时达到提高最小割集阶数的目的。

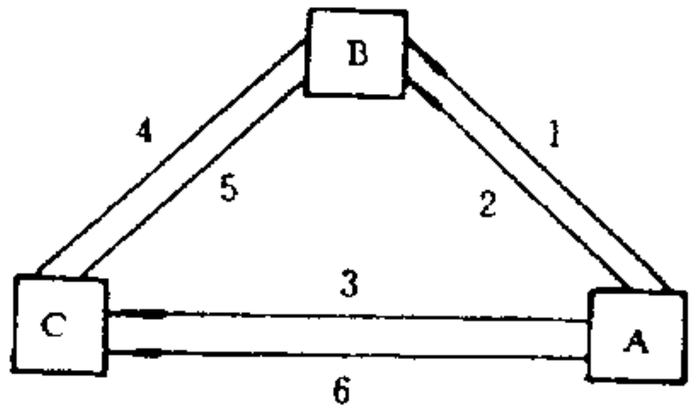


图 5.38　输变电网络改进方案之一

对图 5.38 系统建造故障树并进行定性分析可得全部最小割集为:

$\{X_1、X_2、X_3\}$、$\{X_1、X_2、X_6\}$、$\{X_1、X_3、X_6\}$

$\{X_2、X_3、X_6\}$、$\{X_3、X_6、X_4、X_5\}$、$\{X_1、X_2、X_4、X_5\}$

和改进前比较易见,系统可靠性将得到显著提高。

c. 如果系统的改进受投资的约束,上述 A、B、C 各站之间都用备份线路的方案(图 5.38)投资过大,那么根据此方案的定性分析结果,二个 4 阶最小割集的重要性较小,所以可以取消线路 4 或线路 5 以节省投资,此时系统结构如图 5.39 所示。

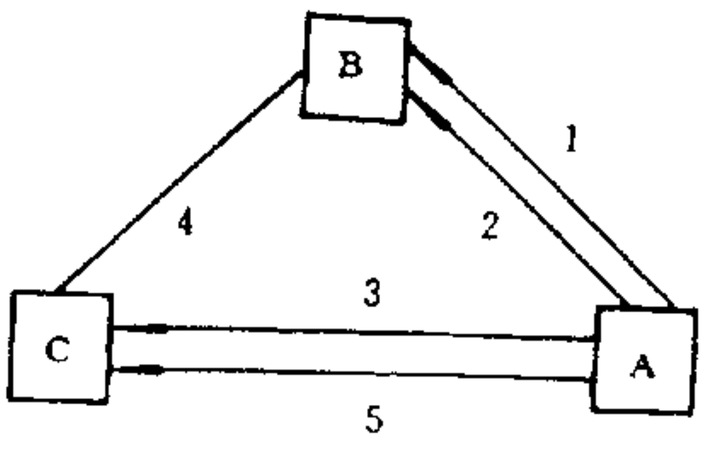


图 5.39　输变电网络改进方案之二

图 5.39 系统的故障树有六个 3 阶最小割集,和图 5.3 原系统的三个 2 阶最小割集加一个 3 阶最小割集相比较,显然图 5.39 系统的故障概率更低,可靠性更高。这就说明,故障树定性分析结果可以有助于系统方案的比较和论证,指导投资的合理分配。当然,这是在工程判断基础上,根据故障树定性比较结果,得出的提示性意见。实际系统的优化设计应当进一步细致调查研究,综合考虑可靠性和其他经济、技术因素,然后再做决策。

5.3.6.4　把“模块最小割集”转换为“底事件最小割集”

模块是子故障树,它可以独立进行定性和定量分析。对原故障树来说,把“模块最小割集”转换为“底事件最小割集”的规则如下。

如果“模块最小割集”仅由一个模块组成,则根据模块和底事件之间的关系直接代入。如

图 5.40(b)中的{M_3}；

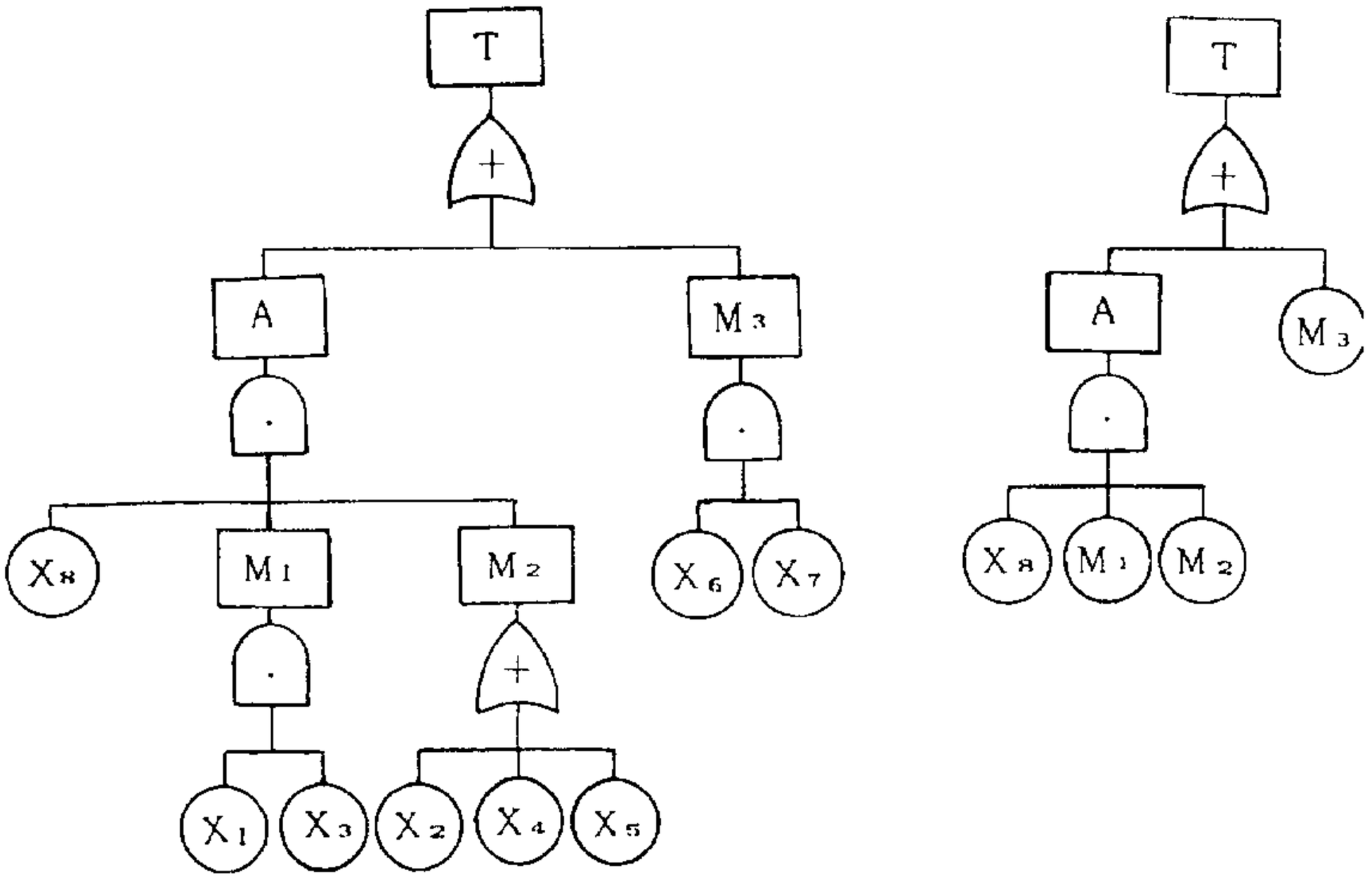


图 5.40　故障树模块化的例子

如果“模块最小割集”中包含多个模块，则根据模块和底事件之间的关系，按照事件结合律，求得和“模块最小割集”对应的“底事件最小割集”。见图 5.40(b)中的{X_8、M_1、M_2}，这里为说明方便，未把{X_8、M_1、M_2}看作一个更大的模块。

模块化故障树的“模块最小割集”有二：{X_8、M_1、M_2}、{M_3}。

各模块及其对应底事件关系如下表

模块编号	M_1	M_2	M_3
对应底事件	X_1、X_3 的交	X_2、X_4、X_5 的并	X_6、X_7 的交

按上述规则求得二个“模块最小割集”对应的四个“底事件最小割集”如下：

{X_8、X_1、X_3、X_2}，{X_8、X_1、X_3、X_4}，{X_8、X_1、X_3、X_5}、{X_6、X_7}。

故障树顶事件可表示为

$$T = X_8X_1X_3X_2 + X_8X_1X_3X_4 + X_8X_1X_3X_5 + X_6X_7$$

5.4　单调故障树定量分析

5.4.1　目的

单调故障树定量分析的目的是，在底事件互相独立和已知其发生概率的条件下，计算顶事件发生概率和底事件重要度等定量指标。

5.4.2　定量分析指南

在单调故障树定性分析求出全部最小割集的基础上，把故障树顶事件表示为最小割集中

底事件积之和的最简布尔表达式，即可对顶事件发生的概率进行定量计算。

计算有两个条件，即底事件相互独立和已知底事件发生概率。底事件之间的统计独立性主要从工程实际的角度进行判断，若某些底事件互相不独立，按照统计独立的假设进行计算将出现难以接受的误差，则应参考专门文献进行不独立所需的修正。若工程实际能给出大部分底事件发生概率的数据，则建议参照类似情况对少数缺乏数据的底事件给出估计值。若相当多的底事件缺乏数据且又不能给出恰当的估计值，则不适宜进行定量分析，只进行定性分析，仍可得到许多有助于提高安全性和可靠性的信息。

在工程上可以略去高阶最小割集来节省分析工作量，此时仍适用以下的定量分析方法。

5.4.3 顶事件发生概率计算

故障树顶事件发生概率是各个底事件发生概率的函数，即

$$P(T) = Q(q_1, q_2, \cdots, q_n) \cdots\cdots (13)$$

由 5.3.3.4，顶事件可表示为

$$T = \sum_{j=1}^{r} \prod_{X_i \in C_j} X_i \cdots\cdots (14)$$

从式(13)和(14)出发可精确计算顶事件发生概率(见附录 A(补充件))，也可近似计算顶事件发生概率。工程上往往没有必要精确计算，因为计算量大且底事件发生概率值的误差大到使精确计算变得无意义。本指导性技术文件所推荐采用的几种近似计算方法，一般可满足工程上的要求。

5.4.3.1 容斥定理取部分项近似计算

容斥定理为：

$$Q = \sum_{i=1}^{r} (-1)^{i-1} \sum_{1 \leqslant j_1 < \cdots < j_i \leqslant r} P\left(\prod_{l=1}^{i} K_{j_l}\right) \cdots\cdots (15)$$

其中，K_{j_l} 为第 j_l 个最小割集 C_{j_l} 的所有底事件的交，

$$K_{j_l} = \prod_{X_p \in C_{j_l}} X_p$$

r 为最小割集数。令

$$S_i = \sum_{1 \leqslant j_1 < \cdots < j_i \leqslant r} P\left(\prod_{l=1}^{i} K_{j_l}\right) \cdots\cdots (16)$$

则

$$Q = \sum_{i=1}^{r} (-1)^{i-1} S_i \cdots\cdots (17)$$

5.4.3.1.1 容斥定理的首项近似计算式

当 $i = 1$ 时

$$Q = S_1 = \sum_{1 \leqslant j_1 \leqslant r} P(K_{j_1}) = \sum_{j=1}^{r} P(K_j) \cdots\cdots (18)$$

式(18)也常称为相斥近似计算式，因为一旦各最小割集事件之间互斥，则各最小割集事件的和的概率等于各最小割集事件的概率之和。从工程上看，只要各最小割集事件的发生概率

足够小，便可以把各割集事件之间视为互斥，从而可以利用容斥定理首项近似计算式实现顶事件发生概率的计算。

5.4.3.1.2　容斥定理上下限平均近似计算式

理论上不难证明

$$S_1 \geqslant Q \geqslant S_1 - S_2 \quad \cdots\cdots (19)$$

于是可以用上下限的平均值来近似顶事件发生概率值，有

$$Q = \frac{1}{2}(S_1 + S_1 - S_2)$$

$$= S_1 - \frac{1}{2}S_2 \quad \cdots\cdots (20)$$

其中，S_1 如式(18)

$$S_2 = \sum_{1 \leqslant j_1 < j_2 \leqslant r} \mathrm{P}\left(\prod_{t=1}^{2} K_{j_t}\right) = \sum_{1 \leqslant i < j \leqslant r} \mathrm{P}(K_i K_j) \quad \cdots\cdots (21)$$

5.4.3.2　独立近似计算

当各个最小割集中相同的底事件较少且发生概率较低时，可以假设各个最小割集之间互相独立，各个最小割集发生(或不发生)互不相关。而所有最小割集都不发生即顶事件不发生，根据概率论乘法定理，下式成立：

$$1 - Q = \prod_{i=1}^{r}[1 - \mathrm{P}(K_i)] \quad \cdots\cdots (22)$$

据此可以近似计算顶事件发生概率 Q，这就是独立近似计算式，在工程上也比较常用。

5.4.3.3　近似计算示例

图 5.37 故障树的最小割集有四个，它们是

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

则

$$K_1 = X_1 X_2$$

$$K_2 = X_1 X_3$$

$$K_3 = X_2 X_3$$

$$K_4 = X_3 X_4 X_5$$

设底事件 X_1、X_2、X_3、X_4、X_5 的发生概率

$$q_1 = q_2 = q_3 = q_4 = q_5 = 0.01$$

采用近似计算方法，其结果如下。

a. 容斥定理首项近似计算结果

$$Q = \sum_{j=1}^{4} \mathrm{P}(K_j)$$

$$= \mathrm{P}(K_1) + \mathrm{P}(K_2) + \mathrm{P}(K_3) + \mathrm{P}(K_4)$$

$$= \mathrm{P}(X_1 X_2) + \mathrm{P}(X_1 X_3) + \mathrm{P}(X_2 X_3) + \mathrm{P}(X_3 X_4 X_5)$$

$$= q_1 q_2 + q_1 q_3 + q_2 q_3 + q_3 q_4 q_5$$

$$= 0.000\ 301$$

b. 容斥定理上下限平均近似计算结果

$$S_1 = 0.000\ 301$$

$$S_2 = \sum_{1 \leqslant i < j \leqslant 4} P(K_iK_j)$$

$$= P(K_1K_2) + P(K_1K_3) + P(K_1K_4) + P(K_2K_3) + P(K_2K_4) + P(K_3K_4)$$

$$= q_1q_2q_3 + q_1q_2q_3 + q_1q_2q_3q_4q_5 + q_1q_2q_3 + q_1q_3q_4q_5 + q_2q_3q_4q_5$$

$$= 0.000\ 003\ 020\ 1$$

$$Q = S_1 - \frac{1}{2}S_2$$

$$= 0.000\ 299\ 49$$

c. 用独立近似式计算

$$1 - Q = [1 - P(K_1)][1 - P(K_2)][1 - P(K_3)][1 - P(K_4)]$$

即 $$Q = 1 - (1 - q_1q_2)(1 - q_1q_3)(1 - q_2q_3)(1 - q_3q_4q_5) = 0.000\ 300\ 97$$

由附录 A(补充件)可得上例的精确解为 0.000 298 98,故上下限平均值近似解的相对误差为 0.17%,首项近似相对误差为 0.68%,独立近似的相对误差为 0.67%。

就本例而言,底事件发生概率为 1×10^{-2},最小割集的发生概率小于或等于 10^{-4},因此三种近似计算均可接受。

5.4.4 重要度计算

工程实践表明,从可靠性、安全性角度看,系统中各部件并不是同等重要的,因此,引入重要度的概念用以标明某个部件对顶事件发生的影响大小是很必要的。重要度是故障树分析中的一个重要概念,对改进系统设计,制订维修策略是十分有利的。对于不同的对象和要求,应采用不同的重要度。

本节主要介绍较常用的四种重要度,即概率重要度、结构重要度、相对概率重要度、相关割集重要度的计算方法。这些重要度从不同角度反映了部件对顶事件发生的影响大小。

在工程中,重要度分析一般用于以下几个方面:

a. 改进系统设计;

b. 确定系统运行中需监测的部位;

c. 制订系统故障诊断时的核对清单的顺序。

5.4.4.1 底事件概率重要度的计算

将故障概率函数表达式代入 3.14.2 条的概率重要度定义式,便得概率重要度的表达式。

从表达式即可求得指定底事件 i 的概率重要度。

5.4.4.2 底事件结构重要度的计算

从 3.14.1 条定义出发计算结构重要度是很繁的,只在系统部件数少时可行。下面介绍借助概率重要度计算结构重要度的方法:

若 q_l 为底事件 1 发生的概率,当 $q_l = \frac{1}{2}, l = 1,2,\cdots,i-1,i+1,\cdots,n$,则有

$$I_\Phi(i) = I_P(i)$$

因为对顶事件发生概率(故障概率函数)采用近似计算式的前提是底事件发生概率很小，而本法由概率重要度计算结构重要度则假设所有 $q_l = \frac{1}{2}$，与上述前提矛盾，所以由本法计算结构重要度时必须采用故障概率函数的精确表达式而不能再用容斥定理部分项的近似式。一般由附录 A(补充件)得到故障概率函数的不交化精确表达式，由概率重要度计算式便得概率重要度的精确表达式，然后令 $q_l = \frac{1}{2}, l = 1,2,\cdots,i-1,i+1,\cdots,n$，代入上述概率重要度的精确表达式，即得所求结构重要度。

5.4.4.3 底事件相对概率重要度的计算

从 3.14.3 条定义出发即可求得指定底事件 i 的相对概率重要度。

5.4.4.4 底事件的相关割集重要度

从 3.14.4 条定义出发即可求得指定底事件的相关割集重要度。

5.4.4.5 重要度计算示例

从附录 A(补充件)得到图 5.37 故障树的故障函数表达式为：

$$Q = q_1q_2 + q_1(1-q_2)q_3 + (1-q_1)q_2q_3 + (1-q_1)(1-q_2)q_3q_4q_5,$$

试计算底事件 3 的各种重要度值。

a. 底事件 3 的概率重要度：

$$\begin{aligned} I_P(3) &= \frac{\partial Q}{\partial q_3} \\ &= q_1(1-q_2) + (1-q_1)q_2 + (1-q_1)(1-q_2)q_4q_5 \\ &= 0.01 \times 0.99 + 0.99 \times 0.01 + 0.99 \times 0.99 \times 0.01 \times 0.01 \\ &= 0.019\ 898 \end{aligned}$$

b. 底事件 3 的结构重要度：

当 $q_1 = q_2 = q_4 = q_5 = \frac{1}{2}$ 时，

$$\begin{aligned} I_\Phi(3) &= I_P(3) \\ &= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \\ &= 0.562\ 5 \end{aligned}$$

c. 底事件 3 的相对概率重要度：

$$\begin{aligned} I_C(3) &= \frac{q_3}{q_1q_2 + q_1(1-q_2)q_3 + (1-q_1)q_2q_3 + (1-q_1)(1-q_2)q_3q_4q_5,} I_P(3) \\ &= \frac{0.01}{0.000\ 298\ 98} \times 0.019\ 9 \\ &= 0.665\ 5 \end{aligned}$$

d. 底事件 3 的相关割集重要度

$$Q_3(q_1,q_2,q_3,q_4,q_5) = \mathrm{P}\left(\sum_{k=1}^{3}\prod_{X_j\in C_k^{(3)}} X_j\right)$$
$$= q_1(1-q_2)q_3 + (1-q_1)q_2q_3 + (1-q_1)(1-q_2)q_3q_4q_5$$
$$= 0.01^2\times 0.99 + 0.99\times 0.01^2 + 0.992\times 0.01^3$$
$$= 0.000\ 198\ 98$$

$$I_{RC}(3) = \frac{0.000\ 198\ 98}{0.000\ 298\ 98}$$
$$= 0.665\ 5$$

5.5 多状态故障的处理

故障树分析通常只考虑系统和部件只有正常工作和发生故障两种状态，采用二元布尔代数作为进行分析的数学工具。但在工程实践中同一系统或同一部件往往存在着多种可能的故障模式和不同的故障程度必须加以考虑，附录 B(补充件)给出了处理多状态故障的方法。

附　录　A
单调故障树定量分析的精确方法
（补充件）

A1　方法

虽然从理论上说，容斥定理公式本身就是一个精确计算式，但实际上难以采用，因为容斥定理展开后所得多项式的项数为 2^r-1，r 为最小割集数，当 $r=10$ 时，$2^r-1=1023$，计算量太大，难以采用容斥定理计算精确解。当必须得到精确解时，常采用不交化算法。

所谓不交化算法，即是将故障树结构函数的最小割集表达式化为不交化表达式，然后计算顶事件发生概率。

不交化算法采用如下公式：

设 $C_1, C_2, \cdots, C_r$ 为全部最小割集，令 $K_j = \prod\limits_{X_i \in C_j} X_i$，则

$$\begin{aligned} & K_1 + K_2 + \cdots + K_r \\ = {} & K_1 + \overline{K}_1(K_2 + \overline{K}_2(K_3 + \overline{K}_3(K_4 + \cdots + \overline{K}_{r-2}(K_{r-1} + \overline{K}_{r-1}K_r)\cdots))) \quad \cdots\cdots (A1) \end{aligned}$$

$$\begin{aligned} & K_1 + K_2 + \cdots + K_r \\ = {} & K_1 + \overline{K}_1K_2 + \overline{K}_1\overline{K}_2K_3 + \cdots + \overline{K}_1\overline{K}_2\cdots\overline{K}_{r-1}K_r \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (A2) \end{aligned}$$

式(A1)和式(A2)中，设 $K_i = X_1X_2\cdots X_l$，则有

$$\begin{aligned} \overline{K}_i &= (\overline{X_1X_2\cdots X_l}) \\ &= \overline{X}_1 + X_1\overline{X}_2 + X_1X_2\overline{X}_3 + \cdots + X_1X_2\cdots X_{l-1}\overline{X}_l \quad \cdots\cdots\cdots\cdots\cdots\cdots (A3) \end{aligned}$$

将式(A1)、(A2)中的 K_i 和 $\overline{K}_i$ 按定义和式(A3)代入后展开即可得系统的不交化表达式。

式(A1)是递推的，从最内括号开始，由内向外逐层打开括号，最后得到的每一积项相互间为不交集，而不交集和的概率就是各项概率的和。采用式(A1)递推式，计算工作量小，但各项物理意义不明显。式(A2)是非递推的，其计算量比式(A1)大，但各项物理意义很明显，例如 $\overline{K}_1\overline{K}_2K_3$ 就是割集 C_3 发生，割集 C_1, C_2 均不发生。当最小割集数不是很大时，尤其采用手算时常用式(A2)作为不交化计算式。

A2　计算示例

图 5.37 故障树的最小割集有四个，它们为

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

设底事件 X_1, X_2, X_3, X_4, X_5 的发生概率均为 0.01，采用不交化方法精确计算顶事件发生概率。

(a)将 K_j 不交化

$$K_1 = X_1X_2$$

$$K_2 = X_1X_3$$

$$K_3 = X_2X_3$$

$$K_4 = X_3X_4X_5$$

$$K_1 + K_2 + K_3 + K_4$$

$$= K_1 + \overline{K}_1K_2 + \overline{K}_1\overline{K}_2K_3 + \overline{K}_1\overline{K}_2\overline{K}_3K_4$$

$$= X_1X_2 + (\overline{X_1X_2})X_1X_3 + (\overline{X_1X_2})(\overline{X_1X_3})X_2X_3$$

$$+ (\overline{X_1X_2})(\overline{X_1X_3})(\overline{X_2X_3})X_3X_4X_5$$

$$= X_1X_2 + (\overline{X}_1 + X_1\overline{X}_2)X_1X_3 + (\overline{X}_1 + X_1\overline{X}_2)(\overline{X}_1 + X_1\overline{X}_3)X_2X_3$$

$$+ (\overline{X}_1 + X_1\overline{X}_2)(\overline{X}_1 + X_1\overline{X}_3)(\overline{X}_2 + X_2\overline{X}_3)X_3X_4X_5$$

$$= X_1X_2 + X_1\overline{X}_2X_3 + \overline{X}_1X_2X_3 + \overline{X}_1\overline{X}_2X_3X_4X_5$$

(b) 计算顶事件发生概率

$$Q = \mathrm{P}(X_1X_2 + X_1\overline{X}_2X_3 + \overline{X}_1X_2X_3 + \overline{X}_1\overline{X}_2X_3X_4X_5)$$

$$= q_1q_2 + q_1(1 - q_2)q_3 + (1 - q_1)q_2q_3 + (1 - q_1)(1 - q_2)q_3q_4q_5$$

$$= 10^{-2} \times 10^{-2} + 10^{-2} \times 10^{-2} \times 0.99 + 0.99 \times 10^{-2} \times 10^{-2}$$

$$+ 0.99 \times 0.99 \times 10^{-2} \times 10^{-2} \times 10^{-2}$$

$$= 0.000\ 298\ 98$$

附　录　B
多状态故障的处理方法
（补充件）

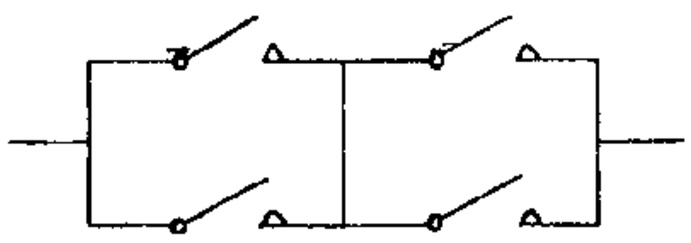
B1　方法

B1.1　处理系统的多个故障模式的方法是，把每一故障模式作为同一个假设的顶事件的直接原因事件，且用逻辑“或”门相连，从而把问题归结为仅有一个假设的顶事件的情形来处理。这个假设的顶事件即“系统发生任一种可能模式的故障”。假设的顶事件不发生表示系统正常。

B1.2　处理系统中第 i 个部件第 m 个故障模式的方法是，把这一故障模式作为一个故障事件 X_{im}，它只可能发生或不发生，并用多元布尔变量 x_{im} 描述这一故障事件的状态，相应地它只可能取值1或0。这样就可以按照5.1条至5.4条的方法进行建树和分析，只是在分析中要注意到同一部件的多个故障模式之间是两两互不相容的，所有故障模式都不发生即该部件正常。一般地，多状态故障树定量分析要把含多状态故障事件的最小割集用多元布尔代数不交化。特殊情况下如果互不相容的多状态故障事件出现在不同的子树中，而这些子树所代表的中间事件也是互不相容的，则不交化过程简化到可用二元布尔代数处理。下面通过示例分别作出说明。

B2　示例

设交通信号线路中红灯开关线路用图 B1 中的四开关组成，要求考虑每个开关和整个开关线路接不通、断不开和正常工作三种状态，试作故障树分析。



SW_1　SW_2　SW_3　SW_4

图 B1　开关线路

B2.1　建树

建造多状态故障树时首先按 B1.1 的方法处理系统的多个故障模式，在本例中顶事件 T 是开关线路故障，它下面接一个逻辑门，此“或”门输入有二，代表系统的两种故障模式：开关线路接不通 E_1 和断不开 E_2。当然，这两种故障模式是互不相容的。而且只有两种故障模式均不发生，系统才能正常工作。

往下追溯开关线路不通的必要而充分的直接原因，是开关 SW_1 和 SW_3 同时接不通，或者开关 SW_2 和 SW_4 同时接不通；

开关线路断不开的必要而充分的直接原因，是在开关 SW_1 断不开的同时 SW_2 或 SW_4 断不开；或者在 SW_3 断不开的同时 SW_2 或 SW_4 断不开。

用 B1.2 的方法，引入底事件 X_{i1}、$X_{i2}(i=1,2,3,4)$ 分别表示四个开关各自接不通或断不

开的故障，X_{i0} 表示开关 i 正常工作。就可以画出图 B2 所示开关线路的多状态故障树。

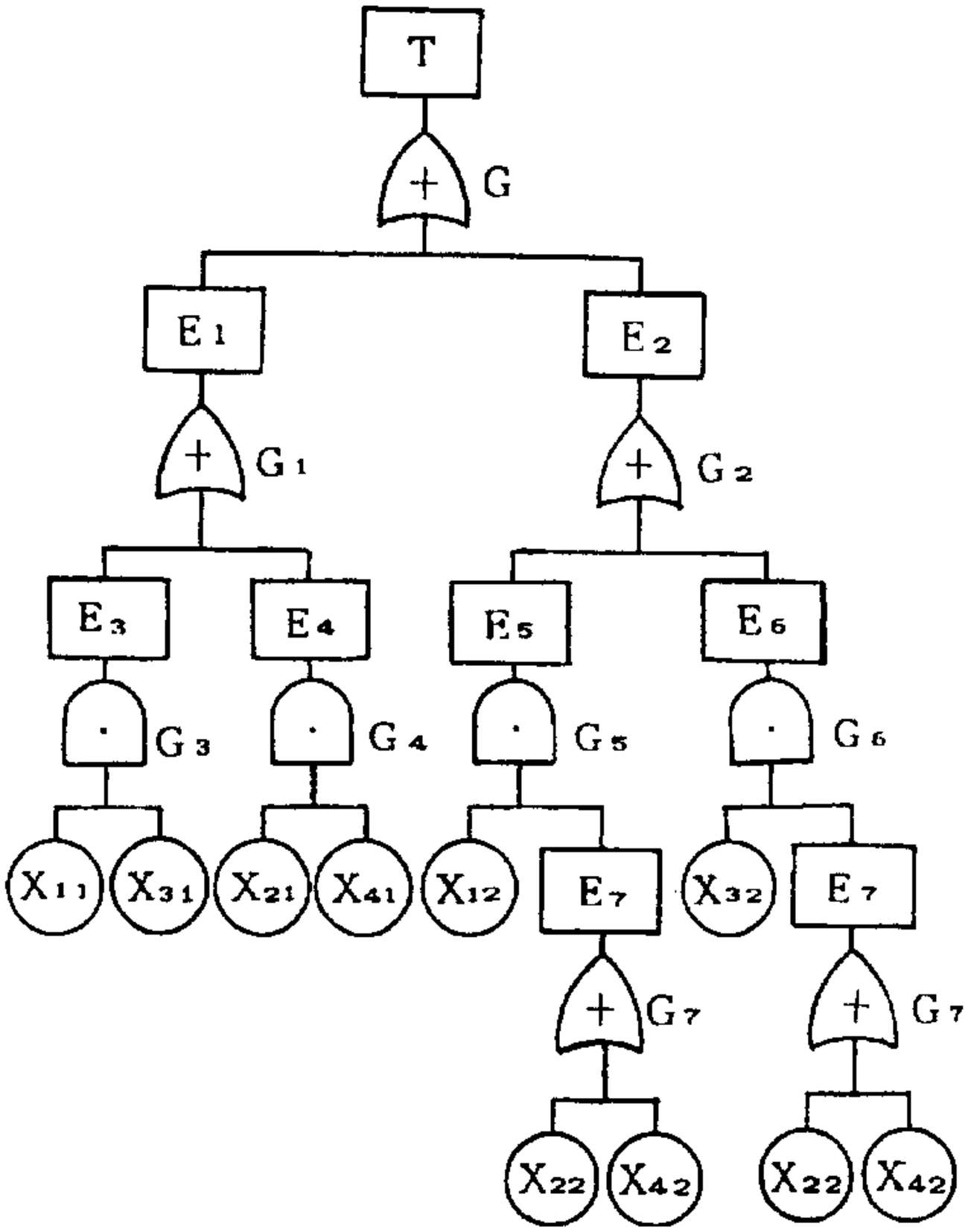


图 B2　开关线路故障树

B2.2　定性分析

用下行法或上行法容易求得图 B2 故障树的全部最小割集：

$$\{X_{11},X_{31}\},\{X_{21},X_{41}\},\{X_{12},X_{22}\},\{X_{12},X_{42}\},\{X_{32},X_{22}\},\{X_{32},X_{42}\},$$

注意到互不相容的底事件 X_{i1} 和 $X_{i2}(i=1,2,3,4)$ 出现在不同的子树 E_1 和 E_2 中，而 E_1、E_2 代表开关线路的互不相容的两种故障模式，可以独立定量分析，不交化过程简化到可用二元布尔代数处理。注意，“互不相容的底事件出现在互不相容的子树中”这个条件是一般的，并不要求都如本例这样，所有 X_{i1} 出现在 E_1，而所有 X_{i2} 出现在 E_2 中。

B2.3　定量分析

B2.3.1　顶事件发生概率计算

先按模块 E_1 和 E_2 分别分析如下：

$$E_1 = X_{11}X_{31} + X_{21}X_{41}$$

按照附录 A(补充件)采用不交化方法

$$E_1 = X_{11}X_{31} + \overline{X_{11}X_{31}}X_{21}X_{41}$$

$$= X_{11}X_{31} \quad \overline{X_{11}}X_{21}X_{41} + X_{11}\overline{X_{31}}X_{21}X_{41}$$

$$P_r(E_1) = q_{11}q_{31} + (1 - q_{11})q_{21}q_{41} + q_{11}(1 - q_{31})q_{21}q_{41}$$

设每个开关接不通概率 $q_{i1} = 0.001(i = 1,2,3,4)$，则开关线路接不通概率

$$P_r(E_1) = 2 \times 10^{-6}$$

$$E_2 = X_{12}X_{22} + X_{12}X_{42} + X_{32}X_{22} + X_{32}X_{42}$$

不交化

$$E_2 = X_{12}X_{22} + \overline{X_{22}}X_{12}X_{42} + \overline{X_{12}}X_{32}X_{22} + \overline{X_{12}X_{22}}X_{32}X_{42}$$

$$P_r(E_2) = q_{12}q_{22} + (1 - q_{22})q_{12}q_{42} + (1 - q_{12})q_{32}q_{22}$$

$$+ (1 - q_{12})(1 - q_{22})q_{32}q_{42}$$

设每个开关断不开概率 $q_{i2} = 0.001(i = 1,2,3,4)$，则开关线路断不开概率

$$P_r(E_2) = 4 \times 10^{-6}$$

顶事件发生概率 $Q = P_r(E_1) + P_r(E_2) = 6 \times 10^{-6}$

图 B1 线路把四个开关等效为一个开关使用，在每个开关接不通或断不开概率都是 0.001 的条件下，把整个开关线路故障概率降低到 6×10^{-6}。其中开关线路接不通概率降低到 2×10^{-6}，比单个开关接不通概率 0.001 降低 500 倍，优先保证交通信号红灯该通则通。

B2.3.2 重要度计算：

概率重要度 $I_P(X_{11}) = \dfrac{\partial Q}{\partial q_{11}} = \dfrac{\partial P_r(E_1)}{\partial q_{11}} = q_{31} - q_{21}q_{41} + (1 - q_{31})q_{21}q_{41} = 0.001$

同理 $I_P(X_{21}) = I_P(X_{31}) = I_P(X_{41}) = 0.001$

相对概率重要度

$$I_C(X_{11}) = I_C(X_{21}) = I_C(X_{31}) = I_C(X_{41}) = \frac{0.001}{6 \times 10^{-6}} \times 0.001 = \frac{1}{6} = 0.167$$

结构重要度

$$I_\Phi(X_{11}) = I_P(X_{11}) \mid_{\text{当所有}q=\frac{1}{2}} = \frac{1}{2} - \frac{1}{4} + \frac{1}{8} = \frac{3}{8} = 0.375$$

同理 $I_\Phi(X_{21}) = I_\Phi(X_{31}) = I_\Phi(X_{41}) = 0.375$

另一方面 $I_P(X_{12}) = \dfrac{\partial Q}{\partial q_{12}} = \dfrac{\partial P_r(E_2)}{\partial q_{12}} = q_{22} + (1 - q_{22})q_{42} - q_{32}q_{22} - (1 - q_{22})q_{32}q_{42}$

$$= 2 \times 10^{-3}$$

同理 $I_P(X_{22}) = I_P(X_{32}) = I_P(X_{42}) = 2 \times 10^{-3}$

$$I_C(X_{12}) = I_C(X_{22}) = I_C(X_{32}) = I_C(X_{42}) = \frac{0.001}{6 \times 10^{-6}} \times 2 \times 10^{-3} = 0.333$$

$$I_\Phi(X_{12}) = I_P(X_{12}) \mid_{\text{当所有}q=\frac{1}{2}} = \frac{3}{8} = 0.375$$

$$I_\Phi(X_{22}) = I_\Phi(X_{32}) = I_\Phi(X_{42}) = I_\Phi(X_{12}) = 0.375$$

B2.4 补充说明

B1.2 已经指出，多状态故障树定量分析一般地要把含多状态故障事件的最小割集用多元布尔代数不交化。在本例中子树 E_1 和 E_2 互不相容，所以上面所列的不交化过程简化。如果不做(假设不能做)这样的处理，则一般含多状态故障事件的最小割集不交化过程，可通过本例解释如下。

$$T = X_{11}X_{31} + X_{21}X_{41} + X_{12}X_{22} + X_{12}X_{42} + X_{32}X_{22} + X_{32}X_{42}$$

不交化

$$\begin{aligned} T = {} & X_{11}X_{31} + \overline{X_{11}X_{31}}X_{21}X_{41} + \overline{X_{11}X_{31}X_{21}X_{41}}X_{12}X_{22} \\ & + X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}X_{12}X_{42} + X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}X_{12}X_{42}X_{32}X_{22} \\ & + \overline{X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}X_{12}X_{42}X_{32}X_{22}}X_{32}X_{42} \end{aligned}$$

其中

$$\overline{X_{11}X_{31}}X_{21}X_{41} = \overline{X_{11}}X_{21}X_{41} + X_{11}\overline{X_{31}}X_{21}X_{41},$$

$$\begin{aligned} \overline{X_{11}X_{31}X_{21}X_{41}}X_{12}X_{22} &= (\overline{X_{11}} + X_{11}\overline{X_{31}})(\overline{X_{21}} + X_{21}\overline{X_{41}})X_{12}X_{22} \\ &= \overline{X_{11}X_{21}}X_{12}X_{22} = X_{12}X_{22}, \end{aligned}$$

$$\begin{aligned} \overline{X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}}X_{12}X_{42} &= (\overline{X_{11}} + X_{11}\overline{X_{31}})(\overline{X_{21}} + X_{21}\overline{X_{41}})(\overline{X_{12}} + X_{12}\overline{X_{22}})X_{12}X_{42} \\ &= \overline{X_{21}X_{22}}X_{12}X_{42} + X_{21}X_{12}X_{42}, \end{aligned}$$

$$\begin{aligned} & \overline{X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}X_{12}X_{42}}X_{32}X_{22} \\ &= (\overline{X_{11}} + X_{11}\overline{X_{31}})(\overline{X_{21}} + X_{21}\overline{X_{41}})(\overline{X_{12}} + X_{12}\overline{X_{22}})(\overline{X_{12}} + X_{12}\overline{X_{42}})X_{32}X_{22} \\ &= \overline{X_{11}X_{12}}X_{32}X_{22} + X_{11}X_{32}X_{22}, \end{aligned}$$

$$\begin{aligned} & \overline{X_{11}X_{31}X_{21}X_{41}X_{12}X_{22}X_{12}X_{42}X_{32}X_{22}}X_{32}X_{42} \\ &= (\overline{X_{11}} + X_{11}\overline{X_{31}})(\overline{X_{21}} + X_{21}\overline{X_{41}})(\overline{X_{12}} + X_{12}\overline{X_{22}})(\overline{X_{12}} + X_{12}\overline{X_{42}})(\overline{X_{32}} + X_{32}\overline{X_{22}})X_{32}X_{42} \\ &= (\overline{X_{11}X_{12}} + X_{11}\overline{X_{31}})\overline{X_{22}}X_{32}X_{42} = (\overline{X_{11}X_{12}X_{22}}X_{32}X_{42} + X_{11}\overline{X_{22}}X_{32}X_{42} \end{aligned}$$

这里用多元布尔代数不交化规则进行了展开和化简，这些规则形式举例如下：

$$\overline{X_{11}X_{31}} = \overline{X_{11}} + X_{11}\overline{X_{31}}$$

$$\overline{X_{11}} + X_{11} = 1$$

$$\overline{X_{11}}X_{11} = 0$$

$$\overline{X_{11}}X_{12} = X_{12}$$

$$X_{11}X_{11} = X_{11}$$

$$\overline{X_{11}X_{11}} = \overline{X_{11}}$$

$$X_{11}X_{12} = 0$$

所以顶事件发生概率

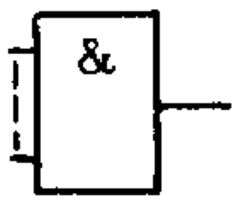
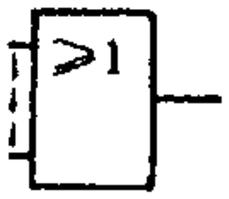
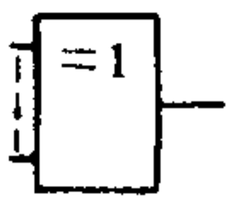
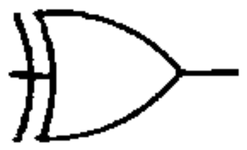
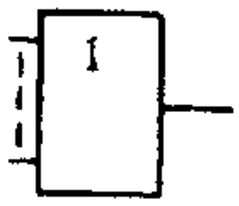
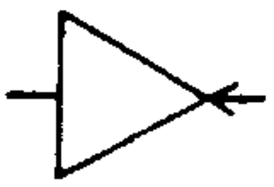
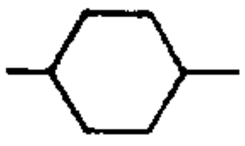
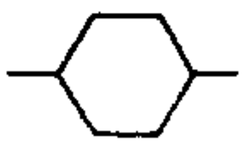
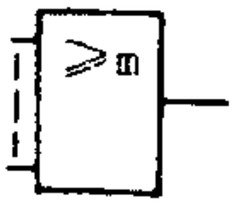
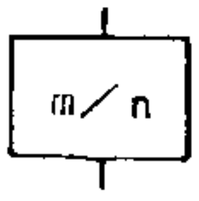
$$\begin{aligned} Q = P_r(T) = {} & q_{11}q_{31} + (1 - q_{11})q_{21}q_{41} + q_{11}(1 - q_{31})q_{21}q_{41} + q_{12}q_{22} \\ & + (1 - q_{21})(1 - q_{22})q_{12}q_{42} + q_{21}q_{12}q_{42} + (1 - q_{11})(1 - q_{12})q_{32}q_{22} \\ & + q_{11}q_{32}q_{22} + [(1 - q_{11})(1 - q_{12}) + q_{11}](1 - q_{22})q_{32}q_{42} \\ \approx {} & 6 \times 10^{-6} \end{aligned}$$

与 B2.3 的计算结果一致。

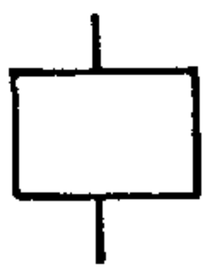
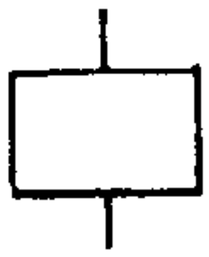
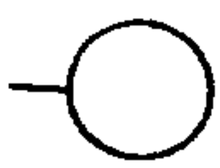
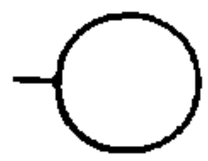
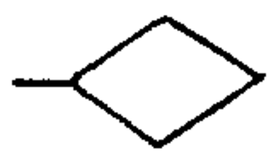
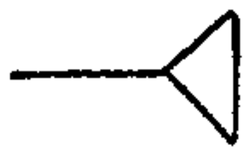
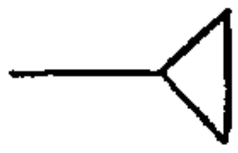
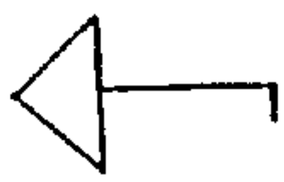
附　录　C
IEC－1025 推荐故障树符号表
（参考件）

C1　IEC－1025 推荐的故障树符号表见表 C1

表 C1

IEC-1025 推荐符号	代用符号	功　　能	说　　明
		与门	全部输入事件发生时输出事件才发生
		或门	至少一个输入事件发生时输出事件就发生
		异或门	当且仅当一个输入事件发生时输出事件才发生
		非门	输出事件是输入事件的逆事件
		禁门	仅当禁门打开条件事件发生时，输入事件的发生方导致输出事件发生
		表决门	n个输入事件中至少有m个发生时，输出事件才发生

续表 C1

IEC-1025 推荐符号	代用符号	功　　能	说　　明
		事件说明方框	方框内可包括事件名称、事件描述和事件编码及发生概率等
		基本事件	不能再分的事件，代表元部件失效或人的失误
		未探明事件	不作进一步分析的事件
		房形事件	已经发生或必将发生的事件
		转入符号	已在本故障树另外地方定义了的事件
		转出符号	用于另外地方的重复事件

附加说明：

本指导性技术文件由中国航天工业总公司提出。

本指导性技术文件由中国航天工业总公司七〇八所归口。

本指导性技术文件由中国航天工业总公司五〇二所、中国科学院应用数学研究所、清华大学核能技术设计研究院起草。

本指导性技术文件主要起草人：廖炯生、曹晋华、梅启智。

计划项目代号：4HT23。