割序集模型中顺序失效符的推演规则

刘 东[1] 王 波[2] 张红林[3]

(装备学院重点实验室 北京 101416)[1] (装备学院研究生院 北京 101416)[2] (63891 部队 洛阳 471003)[3]

摘 要 在割序集(CSS)模型中,为了将由动态故障树(DFT)转换得到的 CSS 初级形式整理为最小割序集(MCSS),提出了顺序失效符(SFS)的推演规则。该推演规则依据基本事件的发生顺序和顺序失效符建立,包含结合律、或分配律、与分配律、吸收律、CSP 规则、WSP 规则等。给出了各类规则的证明过程,并列举了现实中不存在的割序以及由推演规则推导得出的导出规则。SFS 推演规则是 CSS 模型定性分析的进一步形式化描述,它不仅解决了自动获得动态系统最小割序集的问题,而且可在此基础上开展计算机辅助工具的设计工作。

关键词 割序集,顺序失效符,推演规则,动态故障树,可靠性

中图法分类号 TP202.1 **文献标识码** A

Inference Rules of Sequence Failure Symbol in Cut Sequence Set Model

LIU Dong[1] WANG Bo[2] ZHANG Hong-lin[3]

(Key Laboratory, Academy of Equipment, Beijing 101416, China)[1]

(Department of Graduate, Academy of Equipment, Beijing 101416, China)[2]

(63891 Army, Luoyang 471003, China)[3]

Abstract In the cut sequence set(CSS) model, in order to get the minimal cut sequence set(MCSS) from the primary form of CSS, which is transformed from dynamic fault trees(DFT), the inference rules of sequence failure symbol(SFS) were put forward. The rules, which include combination law, or distribution law, and distribution law, absorption law, CSP law and WAP law, were established according to the sequence of basic events and the SFS. The proofs of the rules were provided. The paper also listed some inexistent cut sequences in reality, and provided some educed rules that can be got from the inference rules. SFS inference rules are the formalized qualitative analysis of CSS model. And they can be used to automatically get the MCSS of dynamic systems and design the computer assisted tools.

Keywords Cut sequence set, Sequence failure symbol, Inference rules, Dynamic fault tree, Reliability

1 引言

动态故障树(Dynamic Fault Tree, DFT)模型通过允许部件的动态行为和交互扩展了传统的静态故障树模型[1]。作为一种高级抽象模型,DFT 逐渐应用于可靠性工程领域,许多研究人员已经开展与之相关的研究工作,例如 DFT 的综合分析方法[2]、基于 Petri 网的分析方法[3]、基于贝叶斯网络推理算法的分析方法[4]、基于 Markov 链的分析方法[5]等。一些可靠性建模工具,例如 Galileo[6] 和 Relex[7],已经集成 DFT 并作为分析动态系统的可靠性模型。

DFT 使用 6 种动态门,即 PAND、WSP、CSP、HSP、FDEP 和 SEQ 门,描述动态系统的行为,每一种动态门都具有特定的含义。例如,PAND 门描述了部件失效的时间顺序;WSP、CSP、HSP 门刻画了工作部件及其备件的关系;FDEP 门能够描述部件之间的功能依赖关系;SEQ 门则用于限制事件的发生顺序。

基于 DFT,割序集(Cut Sequence Set, CSS)模型是本文作者提出的一种新的分析动态系统可靠性的定性和定量分析方法[8-10],它包含两个过程,即 CSS 的生成[8,9]和 CSS 定量分析[10]。割序集能够确切表示导致动态系统失效的基本事件发生顺序,其定量分析方法采用多重积分的方式避免了复杂的 Markov 方程组的计算求解过程。割序(Cut Sequence)是能够导致 DFT 中顶事件发生的基本事件序列,CSS 则是所有割序的集合。例如,对于一个含有部件 A 和 B 的并联系统,当 A 和 B 都失效时系统才会失效,则该系统的 CSS 为 $\{(A\rightarrow B)\cup(B\rightarrow A)\}$,其中顺序失效符(Sequence Failure Symbol, SFS)"$\rightarrow$"表示了事件的发生时间关系,而顺序失效表达式(Sequence Failure Expression, SFE) $\{A\rightarrow B\}$ 表示 B 在 A 之后失效。

在 CSS 模型中,为了生成由 SFE 组成的 CSS,需要将 DFT 中的静态门和动态门分解为 SFE,这一过程称为 SFS 转换。在文献[8]中,已经讨论过如何将 DFT 转换为 CSS,并指出 CSS 的生成包括如下 3 个步骤。

步骤 1 对静态门和动态门进行 SFS 转换,得到相应的 SFE;

步骤 2 根据步骤 1 的结果生成 CSS 的初级形式;

到稿日期:2011-06-29 返修日期:2011-10-19 本文受国家自然科学基金(60904082)资助。

刘 东(1981—),男,博士,助理研究员,主要研究方向为系统可靠性分析、系统综合集成,E-mail:LD5M@163.com。

步骤 3　对 CSS 的初级形式应用 SFS 推演规则，得到最终的故障树最小割序集(Minimal CSS，MCSS)。

SFS 推演规则用于将 CSS 的初级形式整理为 MCSS，它涉及到对 SFS、SFE 和与/或逻辑符号的基本操作。例如，对于如图 1 所示的 DFT，由步骤 2 得到的 CSS 初级形式为 CSS $=\{A\to(B\to C)\}$，但该割序集并未确切阐明导致顶事件发生的基本事件序列。事实上该 DFT 的 MCSS 为$\{(A\to B\to C)\cup(B\to A\to C)\}$，即导致系统失效的可能情况分别是$\{A\to B\to C\}$和$\{B\to A\to C\}$。由此可以看出利用 SFS 推演规则将 CSS 转换为 MCSS 的必要性。

文献[8]中仅给出了 SFS 推演规则的初步定义，本文在文献[8]的基础上，进一步深入研究了 CSS 模型中的 SFS 推演规则，分别列举出 CSS 模型中与 SFS 相关的逻辑或时序操作，并给出其证明，用以解决将 CSS 的初级形式转换为 MCSS 的问题。

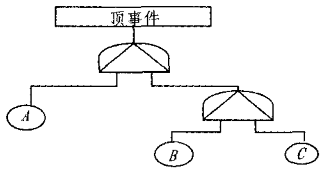

图 1　DFT 示例

2　相关概念

定义 1　$A\gg B$

符号“$\gg$”表示事件发生的先后关系，即 A 在 B 之前发生。

$\gg$与$\to$不同。对于 $A\gg B$，A 和 B 之间可能存在其它事件；而 $A\to B$ 则表示 B 紧接在 A 之后失效，两者之间没有其它事件。

定义 2　${}^{0}_{A}B$

${}^{0}_{A}B$ 带有左上标 0 和左下标 A，表示 B 在 A 失效后才开始工作，而在 A 失效之前，B 的失效率为 0。

由于冷储备部件在进入工作状态之前的失效率为 0，因此可以用 $A\to{}^{0}_{A}B$ 表示 B 作为 A 的冷储备，并且其失效顺序为先 A 后 B。如果 B 为 A 的冷储备，则必有 $A\gg{}^{0}_{A}B$。

定义 3　${}^{\alpha}_{A}B(0<\alpha<1)$

${}^{\alpha}_{A}B$ 带有左上标 α 和左下标 A，表示 B 在 A 失效后开始进入正常工作状态，而在 A 失效之前，B 处于温储备状态，其睡眠因子为 α。

定义 4　${}^{\alpha}B(0<\alpha<1)$

${}^{\alpha}B$ 只带有左上标 α，表示 B 的失效不与其它事件相关，并且 B 在失效之前的睡眠因子为 α。称${}^{\alpha}B$ 为 B 发生了“独立失效”。

如果 WSP 门的初始输入为 E，而 $x_1,x_2,\cdots,x_n$ 是 E 的替补输入，则 WSP 门产生输出的方式可以有多种，因为 $x_i(1\leqslant i<n)$可以在温储备状态下独立失效，也可以因其它事件转为工作状态后再失效。例如，在 WSP 门中，对于按照先 E、再 x_1、再 x_2 的顺序失效的 3 个事件$\{E,x_1,x_2\}$，其所有可能的顺序失效组合为：

$$E\to{}^{\alpha_{x_1}}_{E}x_1\to{}^{\alpha_{x_2}}_{x_1}x_2 \tag{1}$$

$$E\to{}^{\alpha_{x_1}}x_1\to{}^{\alpha_{x_2}}_{E}x_2 \tag{2}$$

在对静态门和动态门进行 SFS 转换之后，故障树顶事件的发生将表示为 SFE 的多种逻辑组合，例如 CSS$=\{(A\to(B\cup C))\cup(C\to(A\cup B))\}$。这种不规则的表达式仍旧不能直接体现出顶事件和基本事件的关系，称之为 CSS 的初级形式。为了统一 CSS 的表示形式，进一步提出 CSS 的一些相关概念。

定义 5　标准顺序失效表达式(Standard SFE，SSFE)

SSFE 是指只含 SFS 和基本事件的 SFE。

定义 6　标准割序(Standard CS，SCS)

SCS 是指由 SSFE 表示的故障树割序。

定义 7　标准割序集(Standard CSS，SCSS)

SCSS 是由多个 SCS 的逻辑 OR 表示的割序集。

定义 8　最小割序(Minimal Cut Sequence，MCS)

MCS 是用 SFE 表示的动态系统的一个割序，当去掉该 SFE 中的任何一个基本事件后，该 SFE 将不再成为割序。MCS 一般表示成 SCS 的形式。

定义 9　最小割序集(MCSS)

MCSS 是指动态系统中所有 MCS 的集合，一般表示成 SCSS 的形式。

CSS 的初级形式是多个 SFE 的各种逻辑组合。为了由割序集的初级形式推导出 SCSS，还需要对初级形式的 CSS 进行整理，这种“整理”操作应当依照正确的规则进行，称之为 SFS 的推演规则。

在 CSS 模型中，生成故障树 CSS 的流程如图 2 所示。

图 2　生成故障树 CSS 的流程

3　SFS 的推演规则

3.1　结合律

结合律用于分解由多个 SFS 连接的 SFE，其表示如下：

$$(x_1\to x_2\to\cdots\to x_m)\to(y_1\to y_2\to\cdots\to y_n)\Leftrightarrow$$
$$(x_1\to\cdots\to x_m\to y_1\to\cdots\to y_n)\cup$$
$$(x_1\to\cdots\to x_{m-1}\to y_1\to x_m\to y_2\to\cdots\to y_n)\cup\cdots\cup$$
$$(y_1\to\cdots\to y_n\to x_1\to\cdots\to x_m\to y_n)$$

其中，(1)若 $m\leqslant n$，则共$\sum_{p=1}^{m}C_{m-1}^{p-1}C_n^p$ 项，则(2)若 $m>n$，则共$\sum_{p=1}^{n}C_{n-1}^{p-1}C_m^{p-1}$ 项。

结合律的证明：$(x_1\to x_2\to\cdots\to x_m)\to(y_1\to y_2\to\cdots\to y_n)$意指$(x_1\to x_2\to\cdots\to x_m)$在$(y_1\to y_2\to\cdots\to y_n)$之前发生，即仅需 $x_m\gg y_n$，而不对 x_i 与 y_j 之间的发生顺序作出限制。因此有 $E=E_1\cup E_2\cup\cdots\cup E_m$，其中 $E_p(1\leqslant p\leqslant m)$根据下述方法构造。

(1)若 $m\leqslant n$

步骤 1　在保持所有事件$\{x_1,x_2,\cdots,x_m\}$先后顺序的前提下，将$\{x_1,x_2,\cdots,x_m\}$分为 $p(1\leqslant p\leqslant m)$组，分组结果为$\{X_1',X_2',\cdots,X_p'\}$；

步骤 2　对于每种分组，将 $X_k'(1\leqslant k\leqslant p)$按照下标由小到大的顺序依次插入事件$\{y_1,y_2,\cdots,y_n\}$的 y_h 和 $y_{h+1}(1\leqslant h\leqslant n-1)$之间，或者$y_1$ 左侧，但不能插入到 y_n 的右侧(否则将有 $y_n\gg x_m$)，并要求 X_k'与 $X_{k+1}'(1\leqslant k\leqslant p-1)$不相邻；

步骤 3　用顺序失效符连接插入操作生成的所有事件，

以此作为 E_p 的一项。将所有可能的项用“∪”连接，其结果即为 E_p。

由于将 $\{x_1, x_2, \cdots, x_m\}$ 分为 p 组共有 C_{m-1}^{p-1} 种分法，而根据步骤 2 的规则将 $\{X_1', X_2', \cdots, X_p'\}$ 插入 $\{y_1, y_2, \cdots, y_n\}$ 共有 C_n^p 种方法，因此 E_p 共有 $C_{m-1}^{p-1}C_n^p$ 项。从而，E 共有 $\sum_{p=1}^{m}C_{m-1}^{p-1}C_n^p$ 项。

(2)若 $m>n$

步骤 1　在保持所有事件 $\{y_1, y_2, \cdots, y_n\}$ 先后顺序的前提下，将 $\{y_1, y_2, \cdots, y_n\}$ 分为 $p(1\leqslant p\leqslant n)$ 组，分组结果为 $\{Y_1', Y_2', \cdots, Y_p'\}$；

步骤 2　对于每种分组，将 Y_p' 插入到 x_m 的右侧，将 $Y_k'(1\leqslant k\leqslant p-1)$ 按照下标由小到大的顺序依次插入事件 $\{x_1, x_2, \cdots, x_m\}$ 的 x_h 和 $x_{h+1}(1\leqslant h\leqslant m-1)$ 之间，或者 x_1 左侧，并要求 Y_k' 与 $Y_{k+1}'(1\leqslant k\leqslant p-2)$ 不相邻；

步骤 3　用顺序失效符连接插入操作生成的所有事件，以此作为 E_p 的一项。将所有可能的项用“∪”连接，其结果即为 E_p。

由于将 $\{y_1, y_2, \cdots, y_n\}$ 分为 p 组共有 C_{n-1}^{p-1} 种分法，而根据步骤 2 的规则将 $\{Y_1', Y_2', \cdots, Y_p'\}$ 插入 $\{x_1, x_2, \cdots, x_m\}$ 共有 C_m^{p-1} 种方法，因此 E_p 共有 $C_{n-1}^{p-1}C_m^{p-1}$ 项。从而，E 共有 $\sum_{p=1}^{n}C_{n-1}^{p-1}C_m^{p-1}$ 项。

证毕。

3.2　或分配律

或分配律用于分解含有 OR 逻辑关系的 SFE，以消除非标准割序中的“∪”，其表示如下：

$$
\begin{aligned}
&(x_1\cup x_2\cup\cdots\cup x_m)\rightarrow(y_1\cup y_2\cup\cdots\cup y_n)\Leftrightarrow\\
&(x_1\rightarrow y_1)\cup(x_1\rightarrow y_2)\cup\cdots\cup(x_1\cup y_n)\cup\cdots\cup\\
&(x_m\rightarrow y_1)\cup(x_m\rightarrow y_2)\cup\cdots\cup(x_m\cup y_n)
\end{aligned}
$$

共 $m\times n$ 项。

或分配律的证明：$(x_1\cup x_2\cup\cdots\cup x_m)\rightarrow(y_1\cup y_2\cup\cdots\cup y_n)$ 表示 $\{x_1, x_2, \cdots, x_m\}$ 中的任意一个事件 $x_i(1\leqslant i\leqslant m)$ 在 $\{y_1, y_2, \cdots, y_n\}$ 中任意一个事件 $y_j(1\leqslant j\leqslant n)$ 之前发生，因此可以将 $(x_1\cup x_2\cup\cdots\cup x_m)\rightarrow(y_1\cup y_2\cup\cdots\cup y_n)$ 的结果表示为 $x_i\rightarrow y_j$，共 $m\times n$ 项。证毕。

3.3　与分配律

与分配律用于将“∩”分解，从而生成 SSFE，其表示如下 $(m\leqslant n)$：

$$
\begin{aligned}
&(x_1\rightarrow x_2\rightarrow\cdots\rightarrow x_m)\cap(y_1\rightarrow y_2\rightarrow\cdots\rightarrow y_n)\Leftrightarrow\\
&(x_1\rightarrow\cdots\rightarrow x_m\rightarrow y_1\rightarrow\cdots\rightarrow y_n)\cup\\
&(x_1\rightarrow\cdots\rightarrow x_{m-1}\rightarrow y_1\rightarrow x_m\rightarrow y_2\rightarrow\cdots\rightarrow y_n)\cup\cdots\cup\\
&(y_1\rightarrow\cdots\rightarrow y_n\rightarrow x_1\rightarrow\cdots\rightarrow x_m)
\end{aligned}
$$

共 $\sum_{p=1}^{m}C_{m-1}^{p-1}C_{n+1}^p$ 项。

与分配律的证明：设 $E=(x_1\rightarrow x_2\rightarrow\cdots\rightarrow x_m)\cap(y_1\rightarrow y_2\rightarrow\cdots\rightarrow y_n)$，则 E 表示 x_i 在 $x_j(1\leqslant i<j\leqslant m)$ 之前发生，y_i 在 $y_j(1\leqslant i<j\leqslant n)$ 之前发生。这里并没有对 x_i 和 y_j 之间有顺序失效的限制，因此有 $E=E_1\cup E_2\cup\cdots\cup E_m$，其中 $E_p(1\leqslant p\leqslant m)$ 根据下述方法构造：

步骤 1　在保持所有事件 $\{x_1, x_2, \cdots, x_m\}$ 先后顺序的前提下，将 $\{x_1, x_2, \cdots, x_m\}$ 分为 $p(1\leqslant p\leqslant m)$ 组，分组结果为 $\{X_1', X_2', \cdots, X_p'\}$；

步骤 2　对于每种分组，将 $X_k'(1\leqslant k\leqslant p)$ 按照下标由小到大的顺序依次插入事件 $\{y_1, y_2, \cdots, y_n\}$ 的 y_h 和 $y_{h+1}(1\leqslant h\leqslant n-1)$ 之间，或者 y_1 左侧和 y_n 的右侧，并要求 X_k' 与 $X_{k+1}'(1\leqslant k\leqslant p-1)$ 不相邻；

步骤 3　用顺序失效符连接插入操作生成的所有事件，以此作为 E_p 的一项。将所有可能的项用“∪”连接，其结果即为 E_p。

由于将 $\{x_1, x_2, \cdots, x_m\}$ 分为 p 组共有 C_{m-1}^{p-1} 种分法，而将 $\{X_1', X_2', \cdots, X_p'\}$ 插入 $\{y_1, y_2, \cdots, y_n\}$ 共有 C_{n+1}^p 种方法，因此 E_p 共有 $C_{m-1}^{p-1}C_{n+1}^p$ 项。从而，E 共有 $\sum_{p=1}^{m}C_{m-1}^{p-1}C_{n+1}^p$ 项。证毕。

3.4　吸收律

吸收律是 SFS 较为独特的规则，用于化简含有同一基本事件的 SFE。吸收律包含如下 2 条规则：

吸收律 1：$(\mathrm{SFE}_1\rightarrow x\rightarrow\mathrm{SFE}_2)\cup x\Leftrightarrow x$

其中，SFE_1 或 SFE_2 可以为空。

吸收律 1 的证明：在 $\mathrm{SFE}_1\rightarrow x\rightarrow\mathrm{SFE}_2$ 发生时，基本事件 x 已经发生，即 x 本身已经包含了 $\mathrm{SFE}_1\rightarrow x\rightarrow\mathrm{SFE}_2$ 的情况。证毕。

吸收律 2：$x\rightarrow x\Leftrightarrow x$

吸收律 2 的证明：首先，$x\cap x\Leftrightarrow x$。而根据 SFS 的与分配律，有 $x\cap x\Leftrightarrow(x\rightarrow x)\cup(x\rightarrow x)\Leftrightarrow x\rightarrow x$，因此得到 $x\rightarrow x\Leftrightarrow x$。证毕。

吸收律 2 实际上是由与分配律导出的，由于实际使用中，经常会遇到此类情况，因此本文将其作为一条基本规则，以便在化简 SFE 的表达式时可以直接使用。下文将会介绍此规则的具体应用。

3.5　CSP 规则

CSP 规则用于化简由 CSP 门生成的 SFE，其表示如下：

$(x\gg_x^0 y)\cup y$ 或 $(x\gg_x^0 y)\cap y\Leftrightarrow x\gg_x^0 y$

证明：由于 y 是冷储备部件，因此 y 不能独立失效。如果 y 失效，则 x 已经在 y 之前失效，因此 $x\gg_x^0 y$ 始终成立。证毕。

事实上，对于冷储备部件和温储备部件，其在系统中的工作过程应当满足一定的约束条件。例如，冷储备部件不能独立失效，温储备部件不能再作为冷储备部件。相应地，在 CSS 模型中的 SFE 表示也应当满足一定的约束，否则用于表示导致系统失效的基本事件序列将与现实世界发生的情况相违背。

3.6　WSP 规则

WSP 规则用于化简由 WSP 门生成的 SFE，其表示如下：

WSP 规则 1：$(x\gg_x^\alpha y)\cup y\Leftrightarrow(x\gg_x^\alpha y)\cup^\alpha y$

WSP 规则 1 的证明：y 作为温储备部件，可能独立失效，也可能在 x 之后失效。因此，如果 y 独立失效，则 $(x\gg_x^\alpha y)\cup y\Leftrightarrow(x\gg_x^\alpha y)\cup^\alpha y$；如果 y 在 x 之后失效，则 $(x\gg_x^\alpha y)\cup y\Leftrightarrow x\gg_x^\alpha y)$。综合上述两种情况，有 $(x\gg_x^\alpha y)\cup y\Leftrightarrow(x\gg_x^\alpha y)\cup^\alpha y$。证毕。

WSP 规则 2：$(x\gg_x^\alpha y)\rightarrow y\Leftrightarrow(x\gg_x^\alpha y)$

WSP 规则 2 的证明：由于 y 在 x 之后失效，因此根据吸收律，有 $(x\gg_x^\alpha y)\rightarrow y\Leftrightarrow x\gg_x^\alpha y\rightarrow_x^\alpha y\Leftrightarrow x\gg_x^\alpha y$。证毕。

WSP 规则 3：$(x\gg_x^\alpha y)\cap y\Leftrightarrow(x\gg_x^\alpha y)$

WSP 规则 3 的证明：根据与分配律，有 $(x\gg_x^\alpha y)\cap y\Leftrightarrow(x\gg_x^\alpha y)\rightarrow y)\cup(y\rightarrow(x\gg_x^\alpha y))$，而 $y\rightarrow(x\gg_x^\alpha y)$ 不符合现实情况（下文将单独讨论），因此 $(x\gg_x^\alpha y)\cap y\Leftrightarrow((x\gg_x^\alpha y)\rightarrow y)$。根据

WSP 规则 2,知$(x\gg{}^{\alpha}_{x}y)\cap y\Leftrightarrow((x\gg{}^{\alpha}_{x}y))$。证毕。

4 不存在的割序

在 CSS 模型中,类似 $x\to y\to x$ 的情况不应当出现,因为该表达式违背了事件在时间上发生的先后关系,因此在 CSS 中,应当删除具有这些表达式的 SFE。本文给出如表 1 所列的在现实中不存在的割序。

表 1 现实中不存在的 SFE

(1)	$x\gg SFE\gg x$;
(2)	${}^{0}_{x}y\gg SFE$
(3)	${}^{\alpha}_{x}y\gg SFE$
(4)	$x\gg y\gg{}^{0}_{y}z$(z 同时是 x 和 y 的冷储备部件) $x\gg y\gg{}^{\alpha}_{y}z$(z 同时是 x 和 y 的温储备部件)

(1)对于 $x\gg SFE\gg x$,在 x 已经失效的情况下,不应再次出现 x 失效这一事件。需要注意的是,该情况应当区别于吸收律 $x\to x\Leftrightarrow x$,在吸收律中,两个 x 之间没有任何其他基本事件。

(2)对于${}^{0}_{x}y\gg SFE$,y 作为 x 的冷储备部件,不能在 x 之前失效。

(3)对于${}^{\alpha}_{x}y\gg SFE$,y 作为 x 的温储备部件,如果以正常失效率发生失效,则该事件不能在 x 失效之前发生。

(4)对于 $x\gg y\gg{}^{0}_{y}z$,由于 z 也是 x 的冷储备部件,因此如果 x 在 y 之前失效,则 z 的发生应当是${}^{0}_{x}z$,而不是${}^{0}_{y}z$。$x\gg y\gg{}^{\alpha}_{y}z$ 的情况与此类似。

需要注意的是,上述不存在的割序仅列举出了常见的情况,有关完备的不存在的割序需要进一步的研究。

5 导出规则

由前文的 SFS 推演规则还可以推导出其它类似的表达式规则。例如$(x\to y)\cap x$ 或$(x\to y)\cap y\Leftrightarrow x\to y$;由与分配律可以得到$(SFE_1\to x\to SFE_2)\cap(SFE_3\to x\to SFE_4)\Leftrightarrow(SFE_1\cap SFE_3)\to x\to(SFE_2\cap SFE_4)$,等等。

本小节将以定理的形式给出两个具有重要应用的 SFE 规则,在 SFE 的简化过程中将经常使用这些规则。

定理 1 设 x_i 和 y_j 表示任意基本事件。对于任意两个顺序失效表达式 $SFE_1=\{x_1\to x_2\to\cdots\to x_m\}$和 $SFE_2=\{y_1\to y_2\to\cdots\to y_n\}$,如果存在两个基本事件 x_a 和 x_b,使得 $SFE_1=\{SFE_a\to x_a\gg x_b\to SFE_b\}$且 $SFE_2=\{SFE_c\to x_b\gg x_a\to SFE_d\}$,则有 $SFE_1\cap SFE_2=\varnothing$。

证明:根据与分配律,有 $SFE_1\cap SFE_2=(SFE_1\to SFE_2)\cup(SFE_2\to SFE_1)$。对于 $SFE_1\to SFE_2$,其结果中必然存在有 $x_a\gg x_b\gg x_a$,在现实中不存在此种情况,因此$\{SFE_1\to SFE_2\}=\varnothing$。同理可得$\{SFE_2\to SFE_1\}=\varnothing$。因此有 $SFE_1\cap SFE_2=\varnothing$。证毕。

定理 2 设 $SFE_1=\{x_1\to x_2\to\cdots\to x_m\}$,若 $SFE_2=\{SFE_a\to x_1\gg x_2\gg\cdots\gg x_m\to SFE_b\}$,其中 SFE_a 和 SFE_b 为任意 SFE,则有 $SFE_1\cap SFE_2=SFE_2$。

证明:根据与分配律,可以直接推出上述结论。

6 用例

例 1 $(x\to y)\to(x\to z)$整理为 SSFE 的过程如下:

$(x\to y)\to(x\to z)$

$=(x\to y\to x\to z)\cup(x\to x\to y\to z)\cup(x\to x\to y\to z)$(结合律)

$=x\to x\to y\to z$ (删除不存在的 SFE)

$=x\to y\to z$ (吸收律)

例 2 对割序集 $CSS=\{(A\to(B\cup C))\cup(C\cap(A\to B))\}$进行整理,以便得到标准割序集的过程如下:

$CSS=(A\to B)\cup(A\to C)\cup(C\cap(A\to B))$(或分配律)

$=(A\to B)\cup(A\to C)\cup(C\to(A\to B))\cup((A\to B)\to C)$(与分配律)

$=(A\to B)\cup(A\to C)\cup(C\to A\to B)\cup(A\to C\to B)\cup(A\to B\to C)$(结合律)

通过上述 SFS 转换得到的 CSS 是 SCSS,但并不一定是 MCSS。设系统的割序集 $CSS=\{SFE_1,SFE_2,\cdots,SFE_n\}$,本文给出一种由该 CSS 生成 MCSS 的算法,其伪代码描述如下:

```
MCSS(CSS){//CSS 含有 n 个割序
  for(i=1;i<=n-1;i++)
    for(j=i+1;j<=n;j++)
      if(SFEi∩SFEj==SFEi)
        从 CSS 中删除 SFEj;
  return CSS;
}
```

该算法逐个搜索 CSS 中的割序,利用定理 2 判断每个割序是否是 MCS,并将不是 MCS 的 SFE 从 CSS 中删除,从而保证 CSS 中保留的割序均为 MCS。算法的最终输出即为 MCSS。

例 3 由例 2 得到的割序集为 $CSS=\{(A\to B)\cup(A\to C)\cup(C\to A\to B)\cup(A\to C\to B)\cup(A\to B\to C)\}$。根据求解 MCSS 的算法,由定理 2 知$(A\to B)\cap(C\to A\to B)=(C\to A\to B)$,$(A\to B)\cap(A\to C\to B)=(A\to C\to B)$,$(A\to B)\cap(A\to B\to C)=(A\to B\to C)$,因此割序$(C\to A\to B)$、$(A\to C\to B)$、$(A\to B\to C)$应当从 MCSS 中删除,即该 DFT 的最小割序集为 $MCSS=\{A\to B,A\to C\}$。

例 4 对于如图 3 所示的 DFT,CSS 的初级形式为$\{(A\to(B\cap{}^{0}_{A}C))\cup(B\to(A\cap{}^{0}_{B}C))\}$,将其转换为 MCSS 的过程如下:

$CSS=(A\to(B\to{}^{0}_{A}C))\cup(A\to({}^{0}_{A}C\to B))\cup(B\to(A\to{}^{0}_{B}C))\cup(B\to({}^{0}_{B}C\to A))$

$=(A\to B\to{}^{0}_{A}C)\cup(B\to A\to{}^{0}_{A}C)\cup(A\to{}^{0}_{A}C\to B)\cup({}^{0}_{A}C\to A\to B)\cup(B\to A\to{}^{0}_{B}C)\cup(A\to B\to{}^{0}_{B}C)\cup(B\to{}^{0}_{B}C\to A)\cup({}^{0}_{B}C\to B\to A)$

$=(A\to B\to{}^{0}_{A}C)\cup(A\to{}^{0}_{A}C\to B)\cup(B\to A\to{}^{0}_{B}C)\cup(B\to{}^{0}_{B}C\to A)$

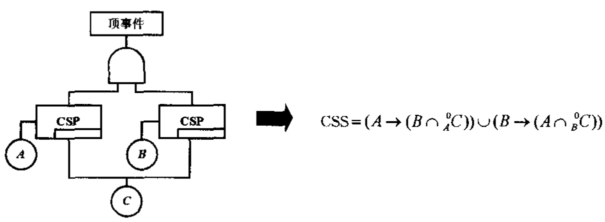

图 3 含 CSP 门的 DFT 示例

结束语 提出了多个 SFS 的推演规则,上述规则仍未涵盖 SFE 的所有可能的各种组合情况。在整理由 DFT 转换得

到的CSS的过程中,仍需要根据实际情况进行具体分析。然而,提出完备的SFS形式化推演规则是基于CSS的DFT定性分析方法的最终目标,因为基于该推演规则,可以利用计算机辅助方法自动完成CSS的定性分析,进而快速给出动态系统中导致顶事件失效的基本事件动态失效模式。

参考文献

[1] Dugan J B, Bavuso S, Boyd M. Dynamic fault tree models for fault tolerant computer systems[J]. IEEE transaction on reliability, 1992, 41(3): 363-377

[2] 李堂经,王新阁,杨哲. 动态故障树的综合分析方法[J]. 装备制造技术,2009(8):22-23,49

[3] Zhang X, Miao Q, Fan X, et al. Dynamic fault tree analysis based on Petri nets[C]// Proceedings of 8th international conference on reliability, maintainability and safety. Chengdu, China, 2009: 138-142

[4] Marquez D, Neil M, Fenton N. Solving dynamic fault trees using a new hybrid bayesian network inference algorithm[C]// Proceedings of 16th Mediterranean Conference on Control and Automation. Ajaccio, France, 2008: 609-614

[5] Boudali H, Crouzen P, Stoelinga M. Dynamic fault tree analysis using input/output interactive markov chains[C]// Proceedings of 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07). Edinburgh, UK, 2007: 708-717

[6] Sullivan K J, Dugan J B, Coppit D. The Galileo fault tree analysis tool[C]// Proceedings of the 29th Annual International Symposium on Fault-Tolerant Computing. Madison, Wisconsin, 1999: 232-235

[7] Relex[EB/OL]. http://www.ptc.com/ products/relex/, 2010-12-16

[8] Liu D, Xing W, Zhang C, et al. Cut sequence set generation for fault tree analysis[C]// Proceedings of 2007 International Conference on Embedded Software and Systems (ICESS 2007). Daegu, South Korea, 2007: 58-69

[9] 陈越洲,谭琳,邢维艳,等. 一种新的故障树定性分析方法[J]. 计算机工程,2008,34(13):67-69

[10] Liu D, Zhang C, Xing W, et al. Quantification of cut sequence set for fault tree analysis[C]// Perrott R, et al., eds. Proceedings of the 3rd international conference on high performance computing and communication (HPCC 2007). 2007: 755-765

(上接第204页)

结束语 受2DPCA-L1的启发,本文提出了2DLPLP-L1。2DLPP-L1使用的是L1范数而非L2范数,这样2DLPP-L1就具有了更强的抗噪声能力。另外,2DLPP-L1使用迭代方法来寻找最优的投影向量,避免了复杂的特征值分解,从而不会出现奇异值问题。与传统的2DLPP-L2相比,所提出的2DLPP-L1抗噪声能力更强。与2DPCA-L2和2DPCA-L1相比,2DLPP-L1也表现出了更优的识别性能。

为了使所提方法更具吸引力,在以后的工作中,我们会从以下两个方面对2DLPP-L1加以研究和探讨。首先,2DLPP-L1需要将原始的数据集转化成由差异向量构成的矩阵。而基于差异向量的迭代过程非常耗时,下一步我们会减小2DLPP-L1的时间复杂度并将之应用于大规模数据集;第二,2DLPP-L1没有考虑类别信息。在以后的工作中,我们将对利用判别信息的监督2DLPP-L1加以研究。

参考文献

[1] Jolliffe I T. Principal Component Analysis (2nd Edition) [M]. New York: Springer, 2002

[2] Duda R O, Hart P E, Stork D H. Pattern Classification (2nd Edition) [M]. Wiley Interscience, 2000

[3] Martínez A M, Kak A C. PCA Versue LDA [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2): 228-233

[4] He X, Niyogi P. Locality Preserving Projections[C]// Advance in Neural Information Processing System. 2003, 16: 152-160

[5] Yang J, Zhang D, Frangi A, et al. Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(1): 131-137

[6] Meng J, Zhang W. Volume Measure in 2DPCA-based Face Recognition [J]. Pattern Recognition Letters, 2007, 28(10): 1203-1208

[7] Ye J, Janardan R, Li Q. Two-dimensional Linear Discriminant Analysis [C]// Advance in Neural Information Processing Systems. 2004, 2: 1569-1576

[8] Liang Z, Li Y, Shi P. A Note on Two-dimensional Linear Discriminant Analysis [J]. Pattern Recognition Letters, 2008, 29: 2122-2128

[9] Chen S, Zhao H, Kong M, et al. A Two-Dimensional Extension of Locality Preserving Projections [J]. Neurocomputing, 2007, 70: 912-921

[10] Hu D, Feng G, Zhou Z. Two-dimensional Locality Preserving Projections (2DLPP) with Its Application to Palmprint Recognition [J]. Pattern Recognition, 2007, 40: 339-342

[11] Yu W. Two-dimensional Discriminant Locality Preserving Projections for Face Recognition [J]. Pattern Recognition Letters, 2009, 30(15): 1378-1383

[12] Kwak N. Principal Component Analysis Based on L1-norm Maximization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(9): 1672-1680

[13] Ding C H Q, Zhou D, He X, et al. R1-PCA: Rotational Invariant L1-norm Principal Component Analysis for Robust Subspace Factorization [C]// Proceedings of the 23rd International Conference on Machine Learning. 2006: 281-288

[14] Pang Y, Li X, Yuan Y. Robust Tensor Analysis with L1-norm [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(2): 172-178

[15] Pang Y, Yuan Y. Outlier-resisting Graph Embedding [J]. Neurocomputing, 2010, 73: 968-974

[16] Li X, Pang Y, Yuan Y. L1-norm-based DPCA [J]. IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics, 2009, 40(4): 1170-1175

[17] Li X, Hu W, Wang H, et al. Linear Discriminant Analysis Using Rotational Invariant L1-norm [J]. Neurocomputing, 2010, 73: 2571-2579