一种状态事件故障树的定量分析方法

徐丙凤[1],黄志球[1],胡 军[1,2],魏 欧[1],肖芳雄[1,3]
(1.南京航空航天大学计算机科学与技术学院,江苏南京 210016;
2.南京大学计算机软件新技术国家重点实验室,江苏南京 210093;
3.广西财经学院信息与统计学院,广西南宁 530003)

摘 要: 状态事件故障树是一种适合于描述复杂系统中失效因果链的建模技术,对系统失效结果的概率特性进行定量分析是获得系统安全性参数的一种重要途径.由于状态事件故障树是半形式化模型,需先精确描述其语义才能进行定量分析.为此,本文提出一种基于交互马尔可夫链的状态事件故障树定量分析方法.首先,通过将交互马尔可夫链的交互动作精化为输入和输出动作,提出接口交互马尔可夫链模型用于状态事件故障树的形式语义描述.然后,在此形式语义的基础上设计了一种状态事件故障树定量分析方法.最后给出了一个飞机起落架收放系统的状态事件故障树建模及概率特性定量分析的实例研究.

关键词: 安全性分析;状态事件故障树;交互马尔可夫链;定量分析;形式化方法
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2013) 08-1480-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.08.005

A Method for Quantitative Analysis of State/Event Fault Tree

XU Bing-feng[1], HUANG Zhi-qiu[1], HU Jun[1,2], WEI Ou[1], XIAO Fang-xiong[1,3]
(1. *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu* 210016, *China*;
2. *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*;
3. *School of Information and Statistics, Guangxi University of Finance and Economics, Nanning, Guangxi* 530003, *China*)

Abstract: State/Event Fault Tree (SEFT) is a modeling technique for describing the causal chains which lead to failure in complex systems. One important way for capturing the safety parameters of systems is quantitatively analyzing the probabilistic characteristic of system failures. As lack of precise semantics, SEFT can only be quantitatively analyzed after its semantics being precisely described. In this paper, we present a quantitative analysis method of SEFT based on Interactive Markov Chain (IMC). Firstly, Interface Interactive Markov Chain (Interface-IMC) is proposed based on refining the interactive action of IMC into input and output actions. Secondly, the precise semantics of SEFT is described based on Interface-IMC. Thirdly, a quantitative analysis method is presented based on formal semantic model of SEFT. Finally, the method in this paper is illustrated by modeling and quantitatively analyzing SEFT of aircraft landing gear system.

Key words: safety analysis; state/event fault tree; interactive Markov chain; quantitative analysis; formal method

1 引言

嵌入式系统在航空、汽车以及工业控制等安全关键领域的广泛应用使得对其进行安全性分析成为系统开发过程中的重要部分[1].随着嵌入式软件系统功能和复杂度的增加,目前嵌入式系统多为组件化分布式架构[2].为对组件化分布式架构的嵌入式系统进行安全性分析,研究人员提出了一系列相关的安全性建模分析方法[3~5].其中,状态事件故障树(State/Event Fault Tree, SEFT)[5]是一种表达系统失效因果链的建模方法,顶层事件表示失效的发生,通过逻辑门和基本构件表达失效发生的因果链.SEFT 中包含了软件系统的行为模型,以及系统中构件的失效与系统失效结果之间的因果关系链,适合于对组件化嵌入式系统进行安全性分析.

对 SEFT 顶层事件发生的概率特性进行定量分析是获得系统安全性参数的一种有效途径.但目前 SEFT 模型尚缺乏精确的语义描述,使得难以直接对其进行定量分析.由于 SEFT 既具有构件化特征又包含了系统的动

收稿日期:2012-11-19;修回日期:2013-01-17;责任编辑:蓝红杰
基金项目:江苏省研究生培养创新工程(No.CXZZ11_0218);中央高校基本科研业务费专项资金(No.CXZZ11_0218,No.NS2012129);国家自然科学基金(No.61272083,No.61170043,No.61100034,No.61262002);回国留学人员科研启动基金(2012)

态行为语义,也难以使用诸如二叉决策图(Binary Decision Diagrams,BDD)[6]等之类的方法对其精确描述和分析.已有分析方法采用 Petri Net 对 SEFT 进行精确的语义描述[5],但 Petri Net 缺少精确描述 SEFT 构件之间消息交互的语义,使得在对 SEFT 中的模型元素进行语义描述之后仍需要专业人员手动修改才能够形成分析模型,难以快速有效地得到整个 SEFT 的形式语义模型.为此,本文工作提出了一种基于交互马尔可夫链(Interactive Markov Chain,IMC)[7]的 SEFT 分析方法,能够给出符合 SEFT 模型特征的形式语义模型并进行 SEFT 的自动定量分析.

2　状态事件故障树

本节介绍状态事件故障树的基本建模元素,并给出一个状态事件故障树的应用实例.

状态事件故障树(SEFT)模型结合了基于状态的模型元素和故障树元素,在基于状态转换的基础上表达失效事件发生的因果链.其基本建模元素如图 1 所示,包括构件、状态、事件等,各建模元素的具体含义见文献[5].

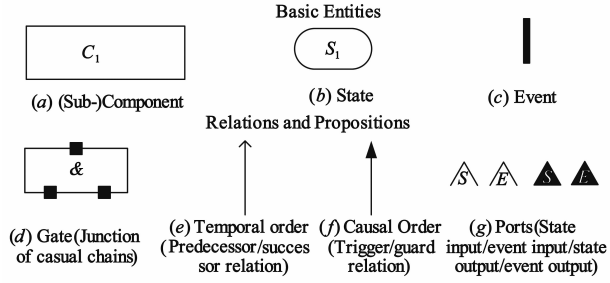

图1　SEFT的基本建模元素

利用 SEFT 可以对组件化嵌入式系统失效的因果关系链进行描述.图 2 给出飞机起落架收放系统中"起落架放不下来"这个失效结果的状态事件故障树.系统包含三个构件,分别是起落架操作手柄(Landing gear control handle)、起落架控制器(Landing gear controller)和起落架信号灯(Landing gear semaphore).起落架操作手柄由飞行员操作,用于控制发送起落架收起或者放下的信号;起落架控制器在接收到收起或者放下信号之后对起落架进行控制;起落架信号灯用于监控起落架控制器,在出现故障的时候发出预警.图 2 中 SEFT 顶层事件是"起落架放不下来"失效的发生,通过两个 AND 逻辑门连接三个构件自底向上描述了该失效发生的因果链.即飞机的起落架在收起的情况下,需要放下起落架,但起落架控制器出现故障并且信号灯没有正常预警从而导致起落架放不下来的失效发生.此外,各构件的内部行为通过有限状态机进行描述,该有限状态机同时描述构件的动作与概率行为.

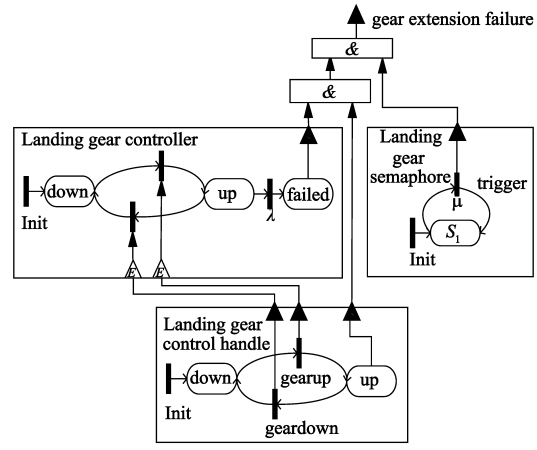

图2　飞机起落架收放系统的SEFT示例

3　接口交互马尔可夫链

本节中将 IMC 的交互动作精化为输入、输出动作,提出接口交互马尔可夫链模型,并定义其并行组合操作用于描述构件系统中的组合行为语义.

3.1　Interface-IMC 模型

接口交互马尔可夫链(Interface-IMC)对交互马尔可夫链(IMC)[7]的交互动作语义进行了精化,将动作分为输入、输出和内部动作且取消了输入使能(input-enabled)特性,明确地表示出在当前状态上哪些输入动作是不可接受的.Interface-IMC 形式定义如下:

定义 1　一个接口交互马尔可夫链 P 是一个五元组$(S, s^0, A, \rightarrow, \rightarrow^M)$,其中:(1)$S$ 是状态集合;(2)s^0 是初始状态;(3)A 是动作集,其中 $A=(A^I, A^o, A^{\text{int}})$,$A^I$ 为输入动作集,A^O 为输出动作集,A^{int}为内部动作集;(4)$\rightarrow$是交互转换集合.通常将$(s, a, s')\in\rightarrow$记作 $s\xrightarrow{a}s'$,Interface-IMC 为非输入使能(noninput-enabled);(5)$\rightarrow^M\subseteq S\times\mathbb{R}_{>0}\times S$ 是马尔可夫转换集合.通常将$(s,\lambda,s')\in\rightarrow^M$ 记作 $s\xrightarrow{\lambda}{}^{M}s'$.

根据以上定义,图 3(a)中 Interface-IMC P 状态集为 $S=\{S_1,S_2,S_3,S_4\}$;初始状态为 S_1;动作集 A 中 $A^I=\{a\}$,$A^O=\{b\}$,$A^{\text{int}}=\phi$;交互转换集$\rightarrow=\{(S_2,a?,S_3),(S_3,b!,S_4)\}$;马尔可夫转换集$\rightarrow^M=\{(S_1,\lambda,S_2)\}$.该 Interface-IMC 的直观语义描述为系统在初始状态 S_1 经过一个停留时间服从参数为 λ 的指数分布的随机转换后转换到 S_2 状态,然后接收动作 a 转换到 S_3 状态,最后通过输出动作 b 转换到 S_4 状态.

3.2　Interface-IMC 的并行组合

构件化系统往往由多个子构件组成,并行组合操作使得可以使用子构件的本地行为模型构建整个系统的全局行为模型[8].下面给出Interface-IMC的并行组合

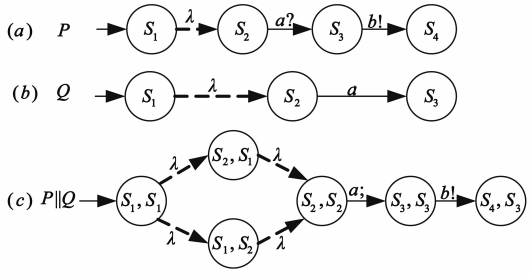

图3 Interface-IMC示例

定义.

定义 2 可组合. 当两个接口交互马尔可夫链 P 和 Q 满足 $A_P^O \cap A_Q^O = A_P^I \cap A_Q^I = A_P^{\text{int}} \cap A_Q = A_P \cap A_Q^{\text{int}} = \phi$ 时，则 P 和 Q 可组合. 同时记 P 和 Q 的共享动作集 $\text{shared}(P,Q) = (A_P^O \cap A_Q^I) \cup (A_P^I \cap A_Q^O)$.

定义 3 Interface-IMC 的并行组合. P 和 Q 为两个 Interface-IMC，如果 P 和 Q 可以组合，则组合的结果 $P \parallel Q$ 为 $(S_P \times S_Q, (S_P^0, S_Q^0), ((P \parallel Q)^I, (P \parallel Q)^O, (P \parallel Q)^{\text{int}}), \rightarrow_{P\parallel Q}, \rightarrow_{P\parallel Q}^M)$ 其中：

- $(P \parallel Q)^I = (A_P^I \cup A_Q^I) \setminus (A_P^O \cup A_Q^O)$；
- $(P \parallel Q)^O = (A_P^O \cup A_Q^O) \setminus (A_P^I \cup A_Q^I)$；
- $(P \parallel Q)^{\text{int}} = A_P^{\text{int}} \cup A_Q^{\text{int}} \cup ((A_P^I \cup A_Q^I) \cap (A_P^O \cup A_Q^O))$；
- $\rightarrow_{P\parallel Q} = \{(s,t) \xrightarrow{a}_{P\parallel Q} (s',t) \mid s \xrightarrow{a}_P s' \wedge Za \in A_P \setminus A_Q\} \cup \{(s,t) \xrightarrow{a}_{P\parallel Q} (s,t') \mid t \xrightarrow{a}_Q t' \wedge a \in A_Q \setminus A_P\} \cup \{(s,t) \xrightarrow{a}_{P\parallel Q} (s',t') \mid s \xrightarrow{a}_P s' \wedge t \xrightarrow{a}_Q t' \wedge a \in A_Q \cap A_P\}$；
- $\rightarrow_{P\parallel Q}^M = \{(s,t) \xrightarrow{\lambda}^M (s',t) \mid s \xrightarrow{\lambda}_P^M s'\} \cup \{(s,t) \xrightarrow{\lambda}^M (s,t') \mid t \xrightarrow{\lambda}_Q^M t'\}$.

根据以上并行组合规则，图 3 中的 P 和 Q 并行组合结果为 $P \parallel Q$，如图 3(c)所示.

在 Interface-IMC 并行组合的过程中可能会出现这样一种情况：在某一个组合状态上的一个 Interface-IMC 相对于另一个 Interface-IMC 有输出动作时，后者并没有相应的输入动作作为接收[8]，这类组合状态称之为非法状态. 在组合的过程中，需要去掉组合状态集中的非法状态. 此外，在 Interface-IMC 并行组合的过程中可能会出现状态空间爆炸的情况. 此时，可以采用状态约简技术化简所生成的状态空间.

4 状态事件故障树(SEFT)的语义描述

对 SEFT 进行形式语义描述[9]是对其进行分析的基础. 本节使用上节中所建立的接口交互马尔可夫链(Interface-IMC)对 SEFT 的构件与逻辑门语义进行精确描述.

4.1 SEFT 构件的语义

由于 SEFT 构件的内部行为使用有限状态机进行描述且同时描述功能和概率特性，使用 Interface-IMC 可对其进行精确语义描述. SEFT 中含有两种转换边：一种是时序转换边；一种是因果转换边. 对于时序转换边，其描述的是构件内部状态转换关系，可以使用 Interface-IMC 中的交互转换对其进行表示. 对于概率延时的时序转换边，由于表达系统经过一个指数时间的概率延迟转换到下一状态，可使用 Interface-IMC 中的马尔可夫转换进行表示. 对于表示构件之间事件触发关系的因果转换边，存在触发事件与被触发事件之间的关系. 而在 Interface-IMC 中存在输入/输出动作的同步关系，所以能够使用输入/输出动作的同步对因果转换边进行描述.

综上，采用 Interface-IMC 来描述 SEFT 构件语义的原则如下：

(1)使用 Interface-IMC 状态描述 SEFT 构件的状态.

(2)使用 Interface-IMC 动作描述 SEFT 构件的事件.

(3)使用 Interface-IMC 中的交互转换对 SEFT 构件内部时序边语义进行描述.

(4)使用 Interface-IMC 之间输出与输入动作发送与接收的同步对因果边语义进行描述.

(5)使用 Interface-IMC 中消息的输入输出类型对 SEFT 状态和事件端口的语义进行描述.

4.2 SEFT 逻辑门的语义

下面使用 Interface-IMC 对 SEFT 逻辑门进行精确的语义描述. 其中，每一种逻辑门的语义(记为 $[\cdots]_{ELT}$)是一个函数，以若干动作作为输入，输出为相应的 Interface-IMC 模型.

图 4(a)给出了两输入 AND 门(AND,2)的语义，即函数 $[(AND,2)]_{ELT}: A^3 \rightarrow Interface - IMC$，以 AND 门的输出和两个输入信号作为参数. 当 AND 门接收到两个输入信号时才会被触发.

图 4(b)给出了两输入 OR 门(OR,2)的语义，即函

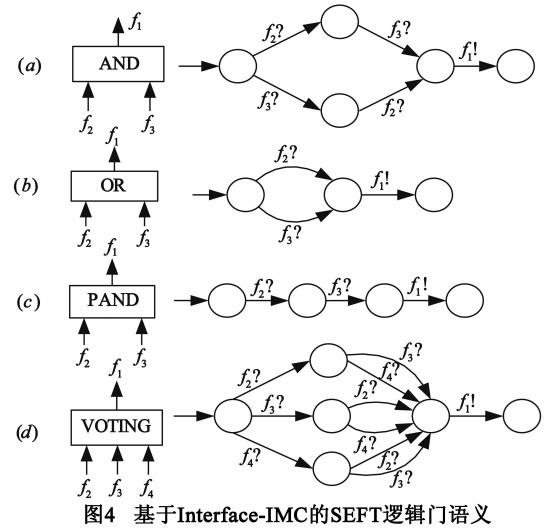

图4 基于Interface-IMC的SEFT逻辑门语义

数$[(OR,2)]_{ELT}: A^3 \rightarrow Interface - IMC$,以 OR 门的输出和两个输入信号为参数.当 OR 门接收到其中一个输入信号时 OR 门被触发.

图 4(*c*)中给出了两输入 PAND 门(PAND,2)的语义,即函数$[(PAND,2)]_{ELT}: A^3 \rightarrow Interface - IMC$,以 PAND 门的输出和两个输入作为参数.当 PAND 门接收到以从左到右的顺序发生的两个输入信号时被触发.

图 4(*d*)给出了 2/3 VOTING 门(*VOTING*,3,2)的语义,即函数$[(VOTING,3,2)]_{ELT}: A^4 \rightarrow Interface - IMC$,以 VOTING 门的输出和三个输入信号为参数.当 VOTING 门接收到三个输入信号中的两个时 VOTING 门被触发.

5　SEFT 的定量分析

本节介绍 SEFT 的定量分析方法.这里,SEFT 模型应为"合法模型"(判断 SEFT 是否合法的具体步骤见文献[5]).对 SEFT 的逻辑门与构件都采用 Interface-IMC 进行了精确的语义描述之后,可通过并行组合得到 SEFT 的完整语义模型,再利用概率分析工具对最终得到的模型进行定量分析.以下对 Interface-IMC 并行组合以及 MRMC 定量计算进行详细介绍.

5.1　Interface-IMC 并行组合

如前文所述,在进行 Interface-IMC 的并行组合过程中,可能存在两个构件接口之间交互不同步所形成的非法状态,在进行并行组合的过程中需要去掉这些非法状态.本文并行组合算法思想为:首先组合两个 Interface-IMC 的初始状态,构造组合后 Interface-IMC 的初始状态.然后根据定义 3 的并行组合规则组合生成下一状态,在这里交替处理马尔可夫转换和动作转换,并在组合状态生成之后判断是否属于非法状态,若属于则去掉状态及相应转换,如算法 1 所示,给定两个 Interface-IMC P 和 Q,通过并行组合得到 P 和 Q 的并行组合结果 R.该算法的时间复杂度为 $O(m \times n)$,其中 m 和 n 分别是 P 和 Q 的状态数.

在进行多个 Interface-IMC 的并行组合过程中,可能会出现状态空间爆炸的问题.为解决该问题,可进一步采用弱互模拟技术对组合结果 Interface-IMC 进行等价状态合并,得到一个与原 Interface-IMC 弱互模拟的模型,然后再与其余的 Interface-IMC 进行并行组合.

算法 1　Computing parallel composition of two Interface-IMCs

Input: $P = (S_P, s_P^0, A_P, \rightarrow, \rightarrow^M)$ and $Q = (S_Q, s_Q^0, A_Q, \rightarrow, \rightarrow^M)$.

Output: Parallel composition result $R = P \parallel Q$.

Function: Parallel composing two Interface-IMCs

if $A_P^O \cap A_Q^O = A_P^{int} \cap A_Q = A_P \cap A_Q^{int} = A_P^I \cap A_Q^I = \phi$

then begin

$sharedaction = A_P \cap A_Q$; $A_R^I = (A_P^I \cup A_Q^I) \setminus (A_P^O \cup A_Q^O)$;

$A_R^O = (A_P^O \cup A_Q^O) \setminus (A_P^I \cup A_Q^I)$; $A_R^{int} = A_P^{int} \cup A_Q^{int} \cup sharedaction$;

$s_R^0 = (s_P^0, s_Q^0)$; $S_R \leftarrow S_R \cup S_R^0$;

$unprocessed\_S_R \leftarrow S_R$, save unprocessed element of S_R;

$temp \leftarrow s_R^0$, save a pair of elements, which respectively represents state of Interface-IMC P and Q;

repeat

pick one unprocessed element *temp* from S_R;

s represents the first element of *temp*, t represents the second element of *temp*;

if $s \xrightarrow{\lambda}_P^M s'$ **then** add (s',t) to S_R and $(s,t) \xrightarrow{\lambda}^M (s',t)$ in $\rightarrow^M$ of R if they do not exist

if $t \xrightarrow{\lambda}_Q^M t'$ **then** add (s,t') to S_R and $(s,t) \xrightarrow{\lambda}^M (s,t')$ in $\rightarrow^M$ of R if they do not exist

if *temp* is not illegal state

then begin

if $s \xrightarrow{a}_P s' \wedge a \in A_P \setminus A_Q$ **then** add (s',t) *to* S_R and $(s,t) \xrightarrow{a}_{P \parallel Q} (s',t)$ *to* $\rightarrow$ of R

if $t \xrightarrow{a}_Q t' \wedge a \in A_Q \setminus A_P$ **then** add (s,t') to S_R and $(s,t) \xrightarrow{a}_{P \parallel Q} (s,t')$ to $\rightarrow$ of R

if $s \xrightarrow{a}_P s' \wedge t \xrightarrow{a}_Q t' \wedge a \in A_Q \cap A_P$ **then** add (s',t') *to* S_R and $(s,t) \xrightarrow{a}_{P \parallel Q} (s',t')$ to $\rightarrow$ of R

end

else begin

remove transition to *temp* from $\rightarrow$ of R;

if *temp* does not appear in $\rightarrow^M$ of R **then** remove *temp* from S_R

end

update $unprocessed\_S_R$;

until $unprocessed\_S_R = \phi$

end

return R

else return false

5.2　SEFT 的概率特性计算

对 SEFT 的定量分析即计算顶层事件发生的概率特性.在采用 Interface-IMC 描述 SEFT 的语义模型之后,SEFT 的顶层事件在 Interface-IMC 中表示为一个输出动作,而构件中输出动作的发生通常认为没有时间延迟.因此,计算 SEFT 顶层事件发生的概率就等同于计算在给定时间区间内,从 SEFT 的形式语义模型的初始状态到达该输出动作的出发状态的概率是否在可接受的范围内.

从 SEFT 构件和逻辑门的形式语义模型的并行组合过程中可以看出,除了描述 SEFT 顶层事件的输出动作未形成内部动作,其他所有的输入输出动作都经过同步成为内部动作.因此在最终用于定量计算的 SEFT 语义模型中,只有表示顶层事件语义的一个输出动作.注意到该输出动作的出发状态即为分析的目标状态,

因而在分析模型中并不需要包含表示顶层事件语义的输出动作.此时从初始状态到目标状态之间所有的转换都为马尔可夫转换.因此,最后分析的模型实际上是一个不包含交互动作的连续时间马尔可夫链(Continuous Time Markov Chain,CTMC)模型[10].

本文利用 MRMC[11]对所得到的 CTMC 模型进行特定时间区间内从初始状态到目标状态的概率计算.具体包括:采用组合结果的初始状态作为进行分析的初始状态;根据 SEFT 中顶层事件所对应的输出动作确定在 SEFT 的语义模型中该输出动作的出发状态,作为目标状态;给定需要分析的时间区间以及概率范围;编写符合 MRMC 输入文件规范的 CTMC 模型文件,使用 MRMC 得到概率计算的结果.在 SEFT 中该概率特性表示在给定的时间内,顶层事件发生的概率是否在可接受的范围内.该定量计算结果可以为评估系统安全性提供参考.

综上所述,图 5 中给出了本文所提出的 SEFT 定量分析方法的具体处理流程.首先得到状态事件故障树的 XML 形式的模型文件,通过解析模型文件,提取建立各构件和逻辑门的 Interface-IMC 模型所需要的信息.然后,对各构件的 Interface-IMC 模型进行并行组合以及状态约简,得到最终的 SEFT 语义模型.在得到 SEFT 的 Interface-IMC 语义模型之后,调用已有的概率模型分析工具对其进行计算,得到计算结果,即 SEFT 中顶层事件的概率特性.可以看出,在该处理流程中,得到 SEFT 模型的 XML 文件之后,整个过程可以完全通过程序自动执行,无需人工参与手动修改.

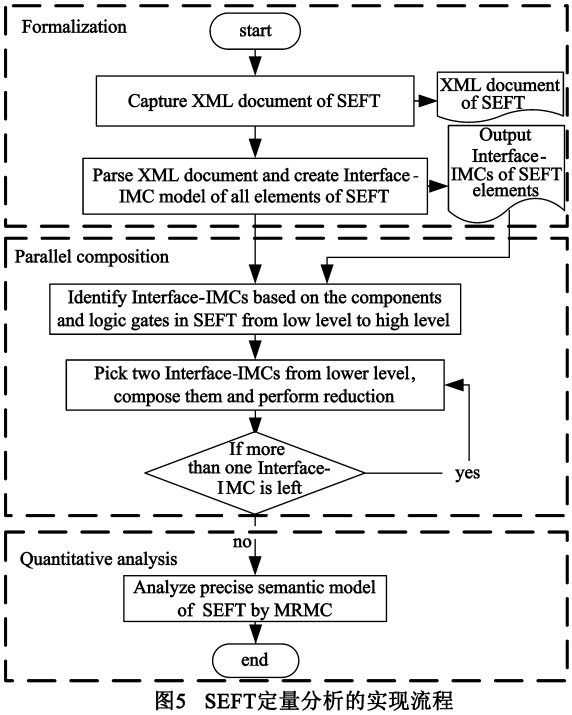

图5 SEFT定量分析的实现流程

6 实例研究

本节针对图 2 中给出的飞机起落架收放系统的 SEFT,利用文章所提出的方法对其进行定量分析.为表述清晰,根据本文方法的处理流程给出完整的执行过程描述.

首先对于图 2 中的 SEFT 采用 Interface-IMC 给出其精确语义.该 SEFT 中一共包含 3 个构件与 2 个逻辑门,利用 Interface-IMC 精确描述 3 个构件与 2 个逻辑门的语义如图 6 所示.论文根据 SEFT 中的构件与逻辑门的层次从下到上对得到的 Interface-IMC 进行编号.可以看出,对该 SEFT 的顶层事件发生概率的计算对应到其形式语义模型中即计算 C_5 中 S_3 状态的概率特性.在得到 SEFT 中各个元素的 Interface-IMC 形式语义模型之后,通过对这些 Interface-IMCs 进行并行组合操作得到 SEFT 的完整语义模型.本实例设定的组合顺序为:$(((C_2 \parallel C_1) \parallel C_4) \parallel C_5) \parallel C_3$,每进行一步组合利用弱互模拟操作对组合之后的 Interface-IMC 进行约简,将约简的结果再进行下一次组合,以此类推,形成最终组合结果.整个并行组合以及约简过程如图 7 所示.由于在该实例中,$f5!$ 是顶层逻辑门的输出,计算 SEFT 的顶层事件概率即计算图 7(8)中 S_3 状态的概率特性.下面采用 MRMC 对图 7(8)进行定量计算.

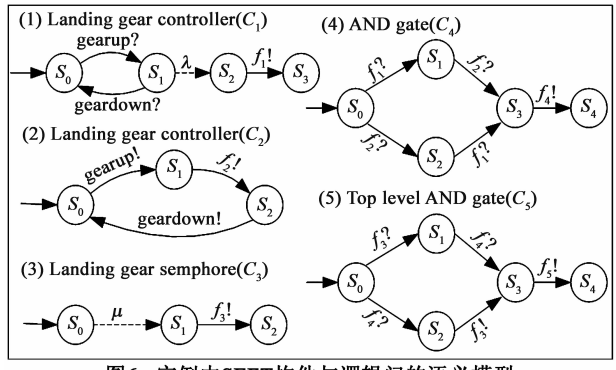

图6 实例中SEFT构件与逻辑门的语义模型

SEFT 中顶层事件发生的概率即 SEFT 的 Interface-IMC 语义模型中目标状态发生的概率,可以得到的概率参数是在给定的时间区间内目标状态发生的概率是否在预期的范围内.例如:图 7(8)中 λ 和 μ 的取值分别为 0.02 和 0.01.分析在[0,10000]时间区间内,从初始状态到达目标状态的概率是否小于 0.0001.利用 MRMC 分析的结果如图 8 所示.由图 8 中的结果可以看出,algs.tra 中的 1 状态为符合要求的状态.即在[0,10000]时间单位内,从初始状态 1 到达目标状态 4 的概率小于 0.0001.因此,对应到 SEFT 中,即在[0,10000]时间区间内,顶层事件发生的概率小于 0.0001.其物理含义为起

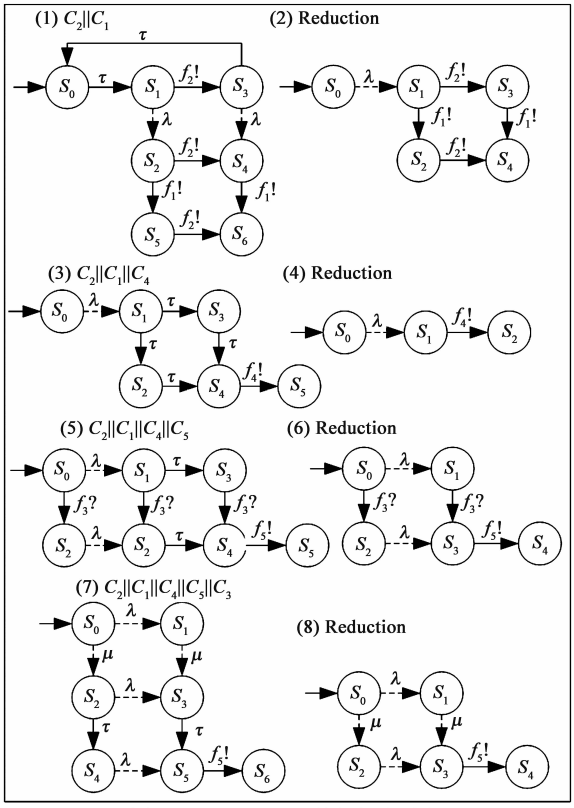

图7　Interface-IMC并行组合过程

落架收放系统不发生“起落架放不下来”失效的概率高于99.99%,该参数可以为评估该系统的安全性提供参考.

```
/cygdrive/e/English/MRMC/mrmc_src_v1.5/bin
 Logic                = CSL
Loading the 'algs.tra' file, please wait.
States=4, Transitions=4
Loading the 'algs.lab' file, please wait.
The Occupied Space is 928 Bytes.
Type 'help' to get help.
>>set simulation on
>>set sim_type one
>>set initial_state 1
>>P{<0.0001}[ initial U[0,10000] end]
$SIMULATED: YES
$INITIAL_STATE: 1
$MAX_NUM_USED_OBSERV: 92800
$CONFIDENCE: 9.500000e-01
$CI_LEFT_RESULT: ( 0.0000000 )
$CI_RIGHT_RESULT: ( 0.0000999 )
$YES_STATE: { 1 }
$NO_STATE: {  }

The Total Elapsed Model-Checking Time is 16 milli sec(s).
```

图8　MRMC的运行结果

7　相关工作

对嵌入式软件进行安全性分析是软件开发过程中的重要步骤.故障树(Fault Tree,FT)[3]是当前工业界安全性分析过程中应用最广泛的技术,但由于故障树在动态系统以及构件化系统方面的表达能力不足,已存在一系列基于故障树的安全性建模分析技术,包括:动态故障树(Dynamic Fault Tree,DFT)[4]、构件故障树(Component Fault Tree,CFT)[5]等.但以上方法仍缺乏描述构件化系统中建模事件顺序以及状态依赖的能力.状态事件故障树(State/Event Fault Tree,SEFT)将传统故障树中的元素与基于状态的模型元素结合起来,同时描述构件化系统的行为模型和失效因果链.

在定量分析方面,由于FT、DFT以及SEFT都缺乏形式语义,难以直接对它们进行定量分析,通常采用具有精确语义的形式化模型[12,13]精确描述其语义再进行分析.如:文献[3]基于BDD分析FT的概率特征;文献[14]基于马尔可夫链分析DFT的概率特征.而对于SEFT的定量分析文献[5]采用Petri Net对其进行定量分析,在对SEFT中的构件和逻辑门进行语义描述之后,仍需对所得到的Petri Net构件进行手动修改并且手动合并生成完整SEFT语义模型,整个过程难以自动有效地进行.与已有的工作相比,本文根据SEFT的特点设计接口交互马尔可夫链对其语义进行描述并设计相应的定量分析方法对其进行定量分析,且整个过程可由程序自动完成,无需人工参与手动修改与合并.

8　总结

本文针对状态事件故障树的定量分析提出了一种新的解决方法.首先通过对交互马尔可夫链的交互动作进行精化,提出了接口交互马尔可夫链模型;其次采用接口交互马尔可夫链对SEFT构件与逻辑门的语义进行精确描述,并通过并行组合得到整个SEFT的语义模型;再利用概率模型分析工具MRMC对SEFT的语义模型进行定量分析,完成对SEFT顶层事件在给定时间内发生的概率进行分析;最终以一个系统实例讨论该方法的可行性.

参考文献

[1] 陈火旺,王戟,董威.高可信软件工程技术[J].电子学报,2003,31(12A):1933-1938.
Chen Huo-wang, Wang Ji, Dong Wei. High confidence software engineering technologies [J]. Acta Electronica Sinica, 2003, 31(12A): 1933 – 1938. (in Chinese)

[2] Mahmood S, Lai R, Soo Kim Y, et al. A survey of component based system quality assurance and assessment [J]. Information and Software Technology, 2005, 47(10): 693 – 707.

[3] Reay KA, Andrews JD. A fault tree analysis strategy using binary decision diagrams [J]. Reliability Engineering & System Safety, 2002, 78(1): 45 – 56.

[4] Čepin M, Mavko B. A dynamic fault tree [J]. Reliability Engineering & System Safety, 2002, 75(1): 83 – 91.

[5] Kaiser B. State Event trees: A safety and reliabiity analysis techniqure for software controlled systems [D]. Kaiserslautern: Universität Kaiserslautern, 2007.

[6] Bryant R E. Graph-based algorithms for Boolean function manipulation [J]. IEEE Transactions on Computers, 1986, 100

(8):677 – 691.
[7] Hersmans H. Interactive Markov Chains[M]. Berlin: Springer-Verlag, 2002.57 – 88.
[8] De Alfaro L, Henzinger T A. Interface automata[A]. Proceedings of the Joint 8th European Software Engineering Conference and 9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (ESEC/FSE 01)[C]. New York: ACM Press, 2001, 109 – 120.
[9] 周颖,郑国梁,李宣东.面向模型检验的 UML 状态机语义[J].电子学报,2003,31(12A):2091 – 2095.
Zhou Ying, Zheng Guo-liang, Li Xuan-dong. An operational semantics for UML state machines in model checking context [J]. Acta Electronica Sinica, 2003, 31(12A): 2091 – 2095. (in Chinese)
[10] Baier C, Haverkort B, Hermanns H, et al. Model-checking algorithms for continuous-time Markov chains [J]. IEEE Transactions on Software Engineering, 2003, 29(6): 524 – 541.
[11] Katoen JP, Khattri M, Zapreevt I. A Markov reward model checker [A]. Proceedings of the Second International Conference on the Quantitative Evaluation of Systems (QEST'05) [C]. Torino: IEEE Computer Society, 2005. 243 – 244.
[12] 张广泉,戎玫,朱雪阳,何亚丽,石慧娟.基于 XYZ/ADL 的 Web 服务组合描述与验证[J].电子学报,2011,39(3A):86 – 93.
Zhang Guang-quan, Rong Mei, Zhu Xue-ya, He Ya-li, Shi Hui-juan. Specification and verification of web service composition based on XYZ/ADL [J]. Acta Electronica Sinica, 2011, 39(3A): 86 – 93. (in Chinese)
[13] 张君华,黄志球,曹子宁.模型检测基于概率时间自动机的反例产生研究[J].计算机研究与发展,2008,45(10):1638 – 1645.
Zhang Jun-hua, Huang Zhi-qiu, Cao Zi-ning. Counterexample generation for probabilistic timed automata model checking [J]. Journal of Computer Research and Development, 2008, 45(10): 1638 – 1645. (in Chinese)
[14] Boudali H, Crouzen P, Stoelinga M. A rigorous, compositional, and extensible framework for dynamic fault tree analysis [J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(2): 128 – 143.

作者简介

徐丙凤 女,1986 年生于安徽安庆市.现为南京航空航天大学计算机科学与技术学院在读博士生.研究方向为软件工程,软件安全性分析与验证.
E-mail: xubingfeng@nuaa.edu.cn

黄志球 男,1965 年生于江苏南京人.在国防科学技术大学获工学学士和工学硕士学位,现为南京航空航天大学教授,博士生导师.主要研究方向为软件工程,服务计算,形式化方法,嵌入式软件分析与验证.

胡 军(通信作者) 男,1973 年生于湖北黄冈.南京大学计算机科学与技术系获计算机软件与理论博士学位,现为南京航空航天大学副教授,硕士生导师.主要研究方向为软件工程.软件分析与验证和嵌入式系统设计与分析.
E-mail: hujun@nuaa.edu.cn