Journal of Southeast University (English Edition) Vol. 25, No. 3, pp. 316 – 319 Sept. 2009 ISSN 1003—7985

Modular solution of dynamic multiple-phased systems

Mo Yuchang[1,2] Yang Quansheng[1]

([1]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

([2]College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China)

Abstract: A new modular solution to the state explosion problem caused by the Markov-based modular solution of dynamic multiple-phased systems is proposed. First, the solution makes full use of the static parts of dynamic multiple-phased systems and constructs cross-phase dynamic modules by combining the dynamic modules of phase fault trees. Secondly, the system binary decision diagram (BDD) from a modularized multiple-phased system (MPS) is generated by using variable ordering and BDD operations. The computational formulations of the BDD node event probability are derived for various node links and the system reliability results are figured out. Finally, a hypothetical multiple-phased system is given to demonstrate the advantages of the dynamic modular solution when the Markov state space and the size of the system BDD are reduced.

Key words: binary decision diagram (BDD); dynamic fault tree; Markov chain; modular solution

The mission of a dynamic multiple-phased system (MPS) is characterized by several phases in time. The system structure, failure and recovery processes, and success criteria may change from phase to phase. During the dynamic fault tree analysis process[1], the fault tree model of the system structure of a MPS is internally and automatically converted to a Markov model, which is added to the dynamic behavior information. However, the size of the Markov model expands exponentially with the increase in the size of the system. Therefore, it can be computationally intensive to solve the model[2].

The modularization of a large fault tree before analysis may achieve high computational efficiency[3]. In 2004, Ou and Dugan[4] presented an approach to the modularization of a MPS. This modular approach modularizes the whole MPS into its static and dynamic modules. During the application of the "total" modular solution to a large real MPS instance, the advantage of the BDD technique cannot be exploited fully due to the commonly encountered inconsistent module joints[5-6]. For example, if there is only one module joint $\overline{M}_1\overline{M}_2$, the BDD solution can be used. But if there are two module joints $\overline{M}_1\overline{M}_2$ and $\overline{M}_1\overline{M}_2M_3$, the BDD solution is not enough to fulfill the task of the joint probability computation and the Markov solution must be used. In order to further alleviate the state explosion problem caused by Markov models, a dynamic modular solution is presented to fully exploit the power of the BDD technique, which only modularizes the dynamic part of the MPS (i. e., only dynamic phase modules are used).

Received 2009-01-05.

Biography: Mo Yuchang(1980—), male, doctor, lecturer, myc@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 60903011), the Natural Science Foundation of Jiangsu Province (No. BK2009267).

Citation: Mo Yuchang, Yang Quansheng. Modular solution of dynamic multiple-phased systems[J]. Journal of Southeast University (English Edition), 2009, 25(3): 316 – 319.

1 HEMPS

We use a reference MPS, which is similar to the hypothetical MPS (HEMPS) presented in Ref. [4], to demonstrate the solutions.

The fault tree of the HEMPS is shown in Fig. 1. It consists of three consecutive phases. Each phase has a distinct reliability requirement, and thus produces a different fault tree. PFT1, PFT2 and PFT3 denote fault trees for phases 1, 2 and 3 of the HEMPS, respectively.

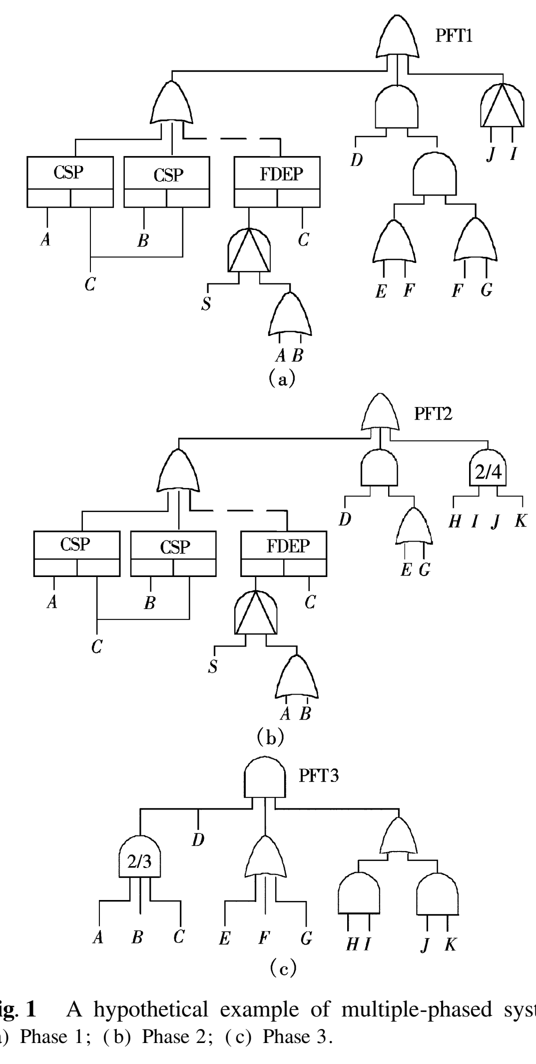

Fig. 1 A hypothetical example of multiple-phased system. (a) Phase 1; (b) Phase 2; (c) Phase 3.

2 Total Modular Solution

2.1 Generation of modules

Based on the most efficient algorithm proposed by Dutuit and Rauzy[3], the HEMPS is modularized to three modules

M_1, M_2 and M_3. The basic events in each module are expressed as

$$M_1 = \{A, B, C, S\},\ M_2 = \{D, E, F, G\}$$
$$M_3 = \{H, K, I, J\}$$

The modularized HEMPS is shown in Fig. 2.

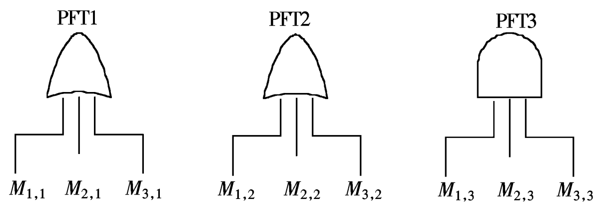

Fig. 2 Modularized HEMPS

2.2 Generation of system BDD

The system BDD for the modularized HEMPS is generated by a variable ordering in the widely used backward PDO form[7-8], i. e., $M_{1,3} < M_{1,2} < M_{1,1} < M_{2,3} < M_{2,2} < M_{2,1} < M_{3,3} < M_{3,2} < M_{3,1}$, as shown in Tab. 1.

The reliability of HEMPS R_{sys} is the sum of the probabilities of the disjoint paths from the root to the terminal 0 vertex through the BDD, i. e.

$$\begin{aligned} R_{\text{sys}} = P\{\overline{M}_{1,1}\overline{M}_{1,2}\overline{M}_{1,3}\}P\{\overline{M}_{2,1}\overline{M}_{2,2}\}P\{\overline{M}_{3,1}\overline{M}_{3,2}\} + \\ P\{\overline{M}_{1,1}\overline{M}_{1,2}M_{1,3}\}P\{\overline{M}_{2,1}\overline{M}_{2,2}\overline{M}_{2,3}\}P\{\overline{M}_{3,1}\overline{M}_{3,2}\} + \\ P\{\overline{M}_{1,1}\overline{M}_{1,2}M_{1,3}\}P\{\overline{M}_{2,1}\overline{M}_{2,2}M_{2,3}\}P\{\overline{M}_{3,1}\overline{M}_{3,2}\overline{M}_{3,3}\} \end{aligned} \tag{1}$$

Tab. 1 System BDD for modularized HEMPS in Fig. 2

Node. index	Node. v	Node. then	Node. else
1	$M_{1,3}$	2	12
2	$M_{1,2}$	1	3
3	$M_{1,1}$	1	4
4	$M_{2,3}$	5	10
5	$M_{2,2}$	1	6
6	$M_{2,1}$	1	7
7	$M_{3,3}$	1	8
8	$M_{3,2}$	1	9
9	$M_{3,1}$	1	0
10	$M_{2,2}$	1	11
11	$M_{2,1}$	1	8
12	$M_{1,2}$	1	13
13	$M_{1,1}$	1	10

2.3 Computation of joint probability for static module

Module M_2 is static through all phases. The BDD-based solution technique in Ref. [5] can be directly applied to compute the consistent joint probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}\overline{M}_{2,3}\}$ or $P\{\overline{M}_{2,1}\overline{M}_{2,2}\}$ without any modification.

Due to $\overline{M}_{2,1}\overline{M}_{2,2}\overline{M}_{2,3} = \overline{M_{2,1} + M_{2,2} + M_{2,3}}$, the probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}\overline{M}_{2,3}\}$ is the sum of the probabilities of the disjoint paths from the root to the terminal 0 vertex through the BDD encoding $M_{2,1} + M_{2,2} + M_{2,3}$, and the probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}\}$ is that through the BDD encoding $M_{2,1} + M_{2,2}$, as shown in Tab. 2.

Tab. 2 Joint BDDs encoding $M_{2,1} + M_{2,2} + M_{2,3}$ and $M_{2,1} + M_{2,2}$

Node. index	$M_{2,1} + M_{2,2} + M_{2,3}$			$M_{2,1} + M_{2,2}$		
	Node. v	Node. then	Node. else	Node. v	Node. then	Node. else
1	D_3	2	0	D_2	2	0
2	E_3	1	3	D_1	3	6
3	G_3	1	4	E_2	1	4
4	F_3	1	0	G_2	1	5
5				F_1	1	0
6				E_2	1	7
7				G_2	1	0

Although module M_2 is static through all phases, the BDD-based technique cannot be directly applied to compute the inconsistent joint probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}M_{2,3}\}$, because it can not be represented as the sum of the probabilities of the disjoint paths from the root to the terminal 0 or 1 vertex through the BDD encoding $M_{2,1} + M_{2,2} + M_{2,3}$. To calculate the joint probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}M_{2,3}\}$, we refer to the Markov chain-based calculation procedure.

First, we work out the state space for M_2. The underlying stochastic process representing the state evolution of M_2 during each phase is a CTMC. The generator matrix is shown in Tab. 3. Secondly, the system initial state "*DEFG*" is set and the Markov chain of the phase at time T_1 is solved. The probabilities at time T_1 of the failing states of $M_{2,1}$ are set to zero. Subsequently, the Markov chain at time $T_2 + T_1$ is solved. And the probabilities at time $T_2 + T_1$ of the failing states of $M_{2,2}$ are set to zero. Finally, the Markov chain at time $T_3 + T_2 + T_1$ is solved to obtain the joint probability $P\{\overline{M}_{2,1}\overline{M}_{2,2}M_{2,3}\}$ by summing up all the probabilities of the failing states of $M_{2,3}$.

3 Dynamic Modular Solution

Several key steps for the dynamic modular analysis of a MPS are sketched out as follows: 1) Dynamic phase modules are found; 2) The system BDD which encodes the modularized MPS is generated; 3) The system BDD is evaluated to produce system reliability results.

3.1 Generation of dynamic modules

The phase modules of a MPS are formed by using the same method as the total modular solution mentioned above. But here we only retain the dynamic modules.

For our HEMPS, module M_2 is static, so it is dropped off and only two dynamic modules M_1 and M_3 are used. The new modularized HEMPS is shown in Fig. 3.

Tab. 3 Generator matrix for CTMC chains of module M_2

State	*DEF*	*DEG*	*DFG*	*EFG*	*DE*	*DG*	*DF*	*EG*	*EF*	*GF*	*F*	*D*	*X*
DEFG	λ_G	λ_F	λ_E	λ_D									
DEF					λ_F		λ_E		λ_D				
DEG					λ_G	λ_E		λ_D					
DFG						λ_F	λ_G			λ_D			
EFG								λ_F	λ_G	λ_E			
DE												λ_E	λ_D
DG												λ_G	λ_D
DF												λ_F	λ_D
EG													
EF											λ_E		λ_F
GF											λ_G		λ_F
F													λ_F
D													λ_D

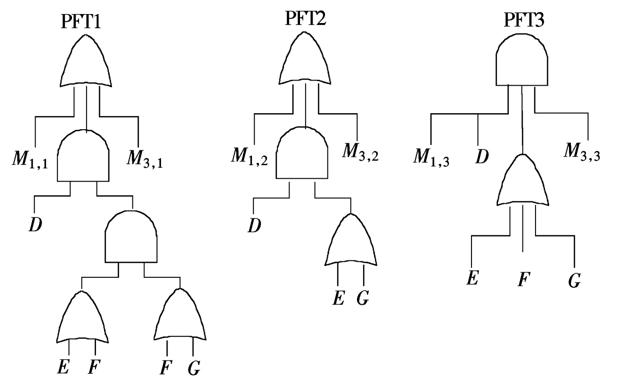

Fig. 3 New modularized HEMPS

3.2 Generation of system BDD

The system BDD for the modularized HEMPS is generated by a variable ordering of $M_{1,1} < M_{1,2} < M_{1,3} < M_{3,1} < M_{3,2} < M_{3,3} < E_3 < E_2 < E_1 < G_3 < G_2 < G_1 < F_3 < F_1 < D_3 < D_2 < D_1$, as shown in Tab. 4. Here, module events are ordered in forward PDO form to make the subsequent joint probability computation smooth. Basic events are ordered in backward PDO form to make the system BDD small.

Tab. 4 System BDD for modularized HEMPS in Fig. 3

Node. index	Node. v	Node. then	Node. else
1	$M_{1,1}$	1	2
2	$M_{1,2}$	1	3
3	$M_{1,3}$	4	16
4	$M_{3,1}$	1	5
5	$M_{3,2}$	1	6
6	$M_{3,3}$	7	11
7	E_3	8	9
8	D_3	1	0
9	G_3	8	10
10	F_3	8	0
11	E_2	12	13
12	D_2	1	0
13	G_2	12	14
14	F_1	15	0
15	D_1	1	0
16	$M_{3,1}$	1	17
17	$M_{3,2}$	1	11

3.3 Evaluation of system BDD

For the root event of a system BDD G, there are two cases.

Case 1 If the root event of G is a basic event A_i, then

$$P\{G=1\} = E\{G\} = E\{A_iG_1 + \overline{A}_iG_0\} = E\{A_iG_1\} + E\{\overline{A}_iG_0\} \tag{2}$$

According to the phase algebra $\overline{A}_i\overline{A}_j = \overline{A}_i(j < i)$, the 0-edge of G always links the variables of two different components. Then we can obtain

$$E\{\overline{A}_iG_0\} = E\{\overline{A}_i\}E\{G_0\} \tag{3}$$

For the 1-edge of G, we have the following two situations: If the 1-edge of G links two basic events of different components, then

$$E\{A_iG_1\} = E\{A_i\}E\{G_1\} \tag{4}$$

If the 1-edge of G links two basic events of the same components, then

$$E\{A_iG_1\} = E\{A_i(A_jH_1 + \overline{A}_jH_0)\} = E\{A_iA_jH_1 + A_i\overline{A}_jH_0\} \tag{5}$$

According to $A_iA_j = A_j$ and $\overline{A}_i\overline{A}_j = \overline{A}_i(j<i)$, then

$$E\{A_iG_1\} = E\{A_jH_1 + \overline{A}_jH_0 - \overline{A}_iH_0\} = E\{G_1\} - E\{\overline{A}_i\}E\{H_0\} \tag{6}$$

Case 2 If the root event of G is a module event M, then

$$P\{G=1\} = E\{G\} = E\{M_iG_1 + \overline{M}_iG_0\} = E\{M_iG_1\} + E\{\overline{M}_iG_0\} \tag{7}$$

1) For the 0-edge of G, we have the following two situations: If the 0-edge of G links two module events of different modules, then

$$E\{\overline{M}_iG_0\} = E\{\overline{M}_i\}E\{G_0\} \tag{8}$$

If the 0-edge of G links two module events of the same modules, then

$$E\{\overline{M}_iG_0\} = E\{\overline{M}_i(M_jH_1 + \overline{M}_jH_0)\} = E\{\overline{M}_iM_jH_1\} + E\{\overline{M}_i\overline{M}_jH_0\} \tag{9}$$

According to the above-mentioned method, the extension of subBDD can be performed recursively until the 0-edge of subBDD links two module events of different modules.

2) Supposing that the 0-edge of H_0 links two module events of different modules, we can obtain

$$E\{\overline{M}_i\overline{M}_jH_0\} = E\{\overline{M}_i\overline{M}_j\}E\{H_0\} \tag{10}$$

For the 1-edge of G, we have the following two situations: If the 1-edge of G links two module events of different modules, then

$$E\{M_i G_1\} = E\{M_i\} E\{G_1\} \tag{11}$$

If the 1-edge of G links two module events of the same modules, then

$$E\{M_i G_1\} = E\{M_i(M_j H_1 + \overline{M}_j H_0)\} = E\{M_i M_j H_1\} + E\{M_i \overline{M}_j H_0\} \tag{12}$$

The extension of subBDD can be performed recursively until the 1-edge of subBDD links two module events of different modules.

3) Supposing that the 1-edge of H_1 links two module events of different modules, we can obtain

$$E\{M_i M_j H_1\} = E\{M_i M_j\} E\{H_1\} \tag{13}$$

The computation method of module joint probabilities, such as $E\{\overline{M}_i \overline{M}_j\}$ and $E\{M_i M_j\}$, is the same as that in the total modular solution. The computation method of basic event probabilities, such as $E\{\overline{A}_i\}$ and $E\{A_i\}$, is given in Ref. [5].

3.4 Advantage of dynamic modularization

To compute the module joint probabilities of static module M_2 by using the total modular solution, two BDDs with sizes 4 and 7 shown in Tab. 2 and one Markov model with 14 states shown in Tab. 3 are generated.

By using the dynamic modular solution, the size of the generated system BDD shown in Tab. 4 and that of the system BDD for the total modular solution shown in Tab. 1 are calculated to be 17 and 13, respectively. Therefore, there is a moderate increase by 17 − 13 = 4 in the size of the system BDD.

As a synthesis of the above comparison, it can be concluded that by using the dynamic modular solution, large Markov models with 14 states for inconsistent joints of static modules and BDDs with sizes 4 and 7 for consistent joints for the same static modules can be dropped off at the cost of a moderate increase in the size of the system BDD (size 4).

4 Conclusion

Due to the commonly encountered inconsistent module joints, the advantage of the BDD technique cannot be exploited fully by the total modular solution. Our dynamic modular solution only retains the dynamic modules, and thus can alleviate the state explosion problem caused by Markov models and exploit the BDD solution fully.

The phase modularization can be a recursive process as dynamic modules might have independent sub-modules inside. To produce system reliability results, we need to compute the joint probability of each dynamic module at the bottom-level of the system. Then the joint probability of an upper dynamic module is directly derived from the joint probabilities of its child module(s). Recursive joint probability computation is performed until the top-level node is reached.

References

[1] Dugan J B, Bavuso S J, Boyd M A. Dynamic fault tree models for fault-tolerant computer system [J]. *IEEE Transactions on Reliability*, 1992, **41**(3): 363 − 377.

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

[3] Dutuit Y, Rauzy A. A linear time algorithm to find modules of fault trees [J]. *IEEE Transactions on Reliability*, 1996, **45**(3): 422 − 425.

[4] Ou Y, Dugan J B. Modular solution of dynamic multi-phase systems [J]. *IEEE Transactions on Reliability*, 2004, **53**(4): 499 − 508.

[5] Rauzy A. New algorithms for fault tree analysis [J]. *Reliability Engineering & System Safety*, 1993, **40**(3): 203 − 211.

[6] Bryant E R. Graph-based algorithms for Boolean function manipulation [J]. *IEEE Transactions on Computers*, 1986, **35**(8): 677 − 691.

[7] Zang X, Sun H, Trivedi K S. A BDD-based algorithm for reliability evaluation of phased mission systems [J]. *IEEE Transactions on Reliability*, 1999, **48**(1): 50 − 60.

[8] Mo Yuchang. Variable ordering to improve BDD analysis of phased-mission systems with multimode failures [J]. *IEEE Transactions on Reliability*, 2009, **58**(1): 53 − 57.

动态多阶段系统模块化分析

莫毓昌[1,2] 杨全胜[1]

([1] 东南大学计算机科学与工程学院, 南京 211189)

([2] 浙江师范大学数理信息学院, 金华 321004)

摘要: 针对 Markov 方法在动态多阶段系统模块化分析中的状态爆炸问题,给出了一种新的动态模块化方法. 该方法首先从充分利用动态多阶段系统所包含的静态特性的角度出发,利用集合并操作将阶段故障树的动态模块构造成跨阶段的动态模块. 然后,利用变量排序和 BDD 操作生成模块化 MPS 对应的系统 BDD. 针对系统 BDD 中各种节点连结关系,推导出 BDD 节点事件概率计算公式,从而计算出动态多阶段系统的可靠度. 最后,通过实例说明了动态模块化方法在 Markov 状态空间和系统 BDD 尺度均减少时的优越性.

关键词: 二进制决策图; 动态故障树; 马尔科夫链; 模块化分析

中图分类号: TP320