第35卷 第15期 计 算 机 工 程 2009年8月
Vol.35 No.15 Computer Engineering August 2009
·安全技术· 文章编号：1000—3428(2009)15—0117—04 文献标识码：A 中图分类号：TP309

基于改进DFTA的安全苛求系统可靠性分析

吴 剑[1]，徐中伟[1]，喻 钢[1,2]，李弋强[1]
(1. 同济大学电子与信息工程学院，上海 201804；2. 上海大学悉尼工商学院，上海 201800)

摘 要：针对传统可靠性分析方法容易忽视冗余多态性、可修复性等安全苛求系统特性问题，在形式化的系统可靠性建模中引入可修复因子，提出一种模块化动态故障树分析方法。在动态和静态相结合的模块化定量分析过程中，利用马尔可夫模型和顶事件发生概率逼近算法，有效避免动态故障树分析过程中的状态组合爆炸问题，提高安全苛求系统可靠性分析的可行性和实践效率。

关键词：动态故障树；安全苛求系统；可靠性分析；马尔科夫模型；模块化

Reliability Analysis of Safety-Critical System Based on Improved Dynamic Fault Tree Analysis Method

WU Jian[1], XU Zhong-wei[1], YU Gang[1,2], LI Yi-qiang[1]
(1. School of Electronics and Information Engineering, Tongji University, Shanghai 201804;
2. Sydney Institute of Language and Commerce, Shanghai University, Shanghai 201800)

【Abstract】Accounting for neglecting repairable attributions or system redundancy in traditional reliability analysis, a new modular dynamic fault tree analysis method is proposed by bringing repairable factors into formalized system reliability modeling. In the process of modular analysis both dynamic and static, the state combination explosive problem is avoided by adopting the approximation algorithm based on the Markov model and the probability of the top events, which improves the accuracy of Safety-Critical System(SCS) estimation results.

【Key words】dynamic fault tree; Safety-Critical System(SCS); reliability analysis; Markov model; modulation

随着计算机技术的飞速发展与应用普及，诸如航空航天、国防、交通运输、核电能源和医疗卫生等安全苛求系统(Safety-Critical System, SCS)除了满足业务功能性需求之外，还必须满足特定的冗余性、容错性等安全性需求，否则一旦系统失效就可能导致人员伤亡等不可预期的灾难性后果。因此，如何对安全苛求系统进行高效准确的可靠性分析是可靠性工程领域研究的焦点[1]。

1 安全苛求系统形式化建模

1.1 安全苛求系统特性

安全苛求系统具有如下特性：

(1)系统多态冗余性：系统失效行为多样化，系统部件采用热备、冷备、优先级等方式工作，且相互间通过约束、与或非等形式逻辑关联。

(2)系统可修复性：系统部件在 t 时刻故障之后单位时间内具有 $\mu(t)$ 的修复概率。

(3)系统部件独立失效性：系统部件状态相互独立，且分别为齐次马尔可夫过程(某时刻后的状态只与该时刻有关，与该时刻之前的状态无关)，若 t 时刻尚未失效，之后单位时间内具有 $\lambda(t)$ 的失效概率。

1.2 系统部件的可修复模型

假定系统部件的故障率和修复率分别为常数 λ_i 和 μ_i，用0来表示系统的正常状态，非0表示故障状态，则系统部件的状态转移情况如图1所示。

在初始状态正常的条件下，该系统部件单位时间内对应的状态转移概率为

$$P_{00}(\Delta t)=\begin{pmatrix}1-\sum_i \lambda_i \Delta t & \sum_i \lambda_i \Delta t \\ \sum_i \mu_i \Delta t & 1-\sum_i \mu_i \Delta t\end{pmatrix}$$

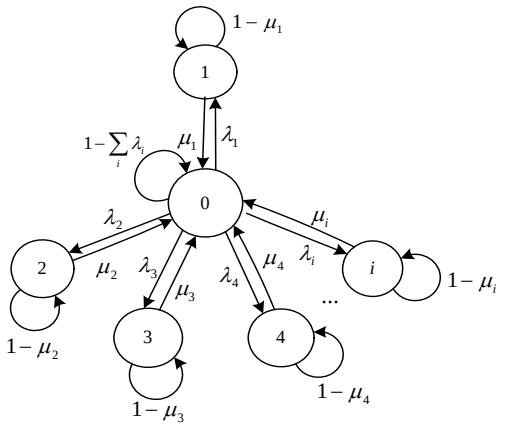

图1 系统部件的状态转移

根据贝叶斯全概率公式和安全苛求系统的可靠性指标，建立部件状态的马尔可夫模型如下：

基金项目：国家自然科学基金资助项目(60674004)；铁道部科技研究开发计划基金资助项目(2007X003)
作者简介：吴 剑(1983－)，男，硕士研究生，主研方向：安全软件形式化，建模、测试和仿真；徐中伟，教授、博士生导师；喻 钢，讲师、博士研究生；李弋强，硕士研究生
收稿日期：2009-02-27 **E-mail：**1212ziegler@163.com

$$P_0(t+\Delta t)=P_0(t)P_{00}(\Delta t)+\sum_i P_i(t)P_{i0}(\Delta t)=$$
$$(1-\sum_i \lambda_i \Delta t)P_0(t)+\sum_i \mu_i \Delta t P_i(t) \quad (1)$$

$$P_i(t+\Delta t)=P_0(t)P_{0i}(\Delta t)+P_i(t)P_{ii}(\Delta t)=$$
$$\lambda_i \Delta t P_0(t)+(1-\mu_i \Delta t)P_i(t) \quad (2)$$

由于 $\lim\limits_{\Delta t\to 0}\dfrac{P_i(t+\Delta t)-P_i(t)}{\Delta t}=P_i'(t)$，因此对式(1)和式(2)作齐次处理($\mu=\sum\mu_i,\lambda=\sum\lambda_i$)得:

$$\left(P_0'(t)\quad P_1'(t)\right)=\left(P_0(t)\quad P_1(t)\right)\begin{pmatrix}-\lambda & \lambda\\ \mu & -\mu\end{pmatrix}$$

考虑到部件状态的二元性和初始完备性(即 $P_0(t)+P_1(t)=1$，$P_0(0)=1$，$P_1(0)=1$)，可得到马尔可夫模型的故障状态概率函数:

$$P_1(t)=\begin{cases}\dfrac{\lambda}{\lambda+\mu}-\dfrac{\lambda}{\lambda+\mu}e^{-(\lambda+\mu)t} & \text{部件初始时正常}\\ \dfrac{\lambda}{\lambda+\mu}+\dfrac{\lambda}{\lambda+\mu}e^{-(\lambda+\mu)t} & \text{部件初始时故障}\end{cases}$$

1.3 安全苛求系统形式化建模

考虑到安全苛求系统特性和建立的系统部件马尔科夫模型，引入改进的形式化动态故障树(Dynamic Fault Tree Analysis, DFTA)方法分析安全苛求系统，可以较好地完成安全苛求系统的可靠性评估，充分考虑了系统的失效状态分布和安全苛求特性[2]。

形式化的安全苛求系统可靠性建模包括:

(1)建立动态故障树(DFT)：通过分析系统失效的运行剖面，以形式逻辑门的形式将系统部件失效联系起来，形成待分析系统的故障树。针对多态冗余等安全苛求特性，须引入动态逻辑门，如图 2 所示。

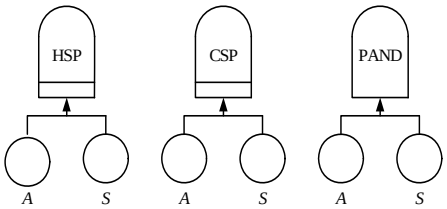

图 2 动态逻辑门

1)热备件门(HSP):*A* 部件运行失效时自动切换到 *S* 备件，未切换时 *A*、*S* 部件均在运行;

2)冷备件门(CSP)：*A* 部件运行失效时中自动启动并切换到 *S* 备件，未切换时 *S* 备件不运行;

3)优先与门(PAND)：按照 *A* 优先于 *S* 运行的规则工作。

(2)动态故障树的模块化[3]：采用深度优先算法(DFLM)，将动态故障树模块化，分解为最小静态子树和最小动态子树。深度优先算法实现步骤为:对动态故障树进行深度优先搜索，对搜索到的故障树中的底事件和中间事件分别设置 3 个标记，第 1 个标记表示第 1 次搜索到该事件所用的步数，第 2 个标记表示第 2 次搜索到该事件所用的步数，第 3 个标记表示最后一次搜索到该事件所用的步数。若对于中间节点 *N* 有：1)与其相连接的所有下层事件中，标记 1 的最小值比节点 *N* 的标记 1 小；2)与其相连接的所有下层事件标记 3 的最大值比节点 *N* 的标记 2 小，则节点 *N* 对应的子树为独立子树，进一步根据该子树是否含有动态逻辑故障门来确定该子树是动态子树还是静态子树。

(3)改进的动态故障树模块化分析方法：结合系统部件的马尔科夫模型来处理安全苛求系统的可修复性，引入下文中改进的模块分析方法对故障树模块的可靠性进行可靠性分析。

2 改进的模块化动态故障树可靠性分析方法

本文对于系统故障树中的静态模块子树分析和动态模块子树分析均做出了改进：(1)引入可修复因子，在可靠性分析中考虑到了安全苛求系统的可修复特性，采用引入的系统部件马尔可夫模型取代不可修复系统部件模型；(2)考虑到安全苛求系统的复杂性和高可信特点，通过引入顶事件概率逼近算法和多态冗余逻辑特性取代传统故障树分析中的马尔可夫状态组合方式，避免了多模块系统中马尔可夫模型组合爆炸的问题[4]。

2.1 静态子树模块可靠性分析

结合安全苛求系统部件的形式化组合逻辑和部件可维修特性，引入静态子树的顶事件逼近算法(下列逻辑门子事件状态为系统部件马尔科夫模型结果):

(1)逻辑与门:

$$P(t)=P(T_1\leqslant t,T_2\leqslant t,\cdots,T_i\leqslant t)=\prod_1^i P_j(t)$$

(2)逻辑或门:

$$P(t)=P\{\min(T_1,T_2,\cdots,T_i)\leqslant t\}=P_1(t)+\sum_2^i\left(P_j(t)\prod_1^{j-1}(1-P_m(t))\right)$$

静态子树如图 3 所示。

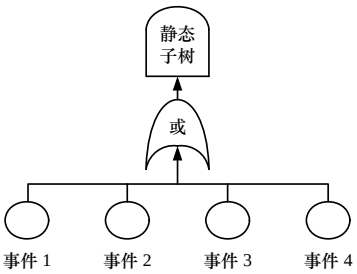

图 3 静态子树

静态子树由 4 个部件通过与门关联，根据前面建立的部件可修复模型，依次可得部件的故障状态函数 $P_i(t)$，由逻辑与门的顶事件逼近算法得子树的故障概率:

$$P(t)=P\{\min(T_1,T_2,T_3,T_4)\leqslant t\}=$$
$$P_1(t)+P_2(t)(1-P_1(t))+$$
$$P_3(t)(1-P_1(t))(1-P_2(t))+$$
$$P_4(t)(1-P_1(t))(1-P_2(t))(1-P_3(t))$$

子树的平均无故障时间为 $MTBF=\int_0^\infty P(t)\mathrm{d}t$。

静态子树的逼近算法与基于 BDD(二元决策树[5])的分析方法相比，避免了故障树转化为 BDD 的过程，优化了所有故障模式和传播途径的实现步骤，逼近误差保持在底事件参数估计范围内(与部件系统的元素个数和单一部件的故障状态有关)。对于部件指标集为 0 和 1 的故障树，逼近算法的分析结果与基于 BDD 方法的分析结果完全一致，静态逼近算法的复杂度为 $\sum\limits_{i=1}^{i=n} i$，优于基于 BDD 方法中的 2^n 种故障模式分析；对于具有多重指标的部件系统，逼近算法归一化处理，严格避免了安全分析中的“漏警”情况。

2.2 动态子树模块可靠性分析

结合动态逻辑门的形式化定义和安全苛求系统的可维修特性，引入动态子树的顶事件失效逼近算法来分析子树的可

靠性指标，下列逻辑门中子事件状态为系统部件马尔科夫模型结果。

(1)优先与门：顶事件的故障状态分布为

$$P(t)=P(T_1\leqslant T_2\leqslant t)=\int_{t_1=0}^{t}\int_{t_2=t_1}^{t}\mathrm{d}P_2(t_2)\mathrm{d}P_1(t_1)=$$

$$\int_{t_1=0}^{t}(P_2(t)-P_1(t_1))\mathrm{d}P_1(t_1)$$

(2)冷备件门：顶事件的故障状态分布为

$$P(t)=P(T_1+T_2\leqslant t)=\int_{t_2=0}^{t}\int_{t_1=0}^{t_2}\mathrm{d}P_1(t_1)\mathrm{d}P_2(t_2-t_1)$$

(3)热备件门：有顶事件的故障状态分布为

$$P(t)=P\{\max(T_1,T_2)\leqslant t\}=P_1(t)P_2(t)$$

动态子树如图 4 所示，其中，[A]表示事件在同一分析系统中。

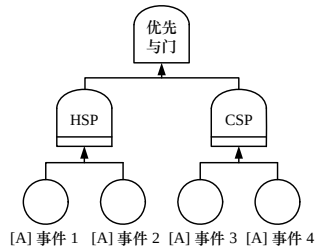

图 4　动态子树

可以得到动态子树的热备件门和冷备件门的故障概率分别为

$$P_H(t)=P\{\max(T_1,T_2)\leqslant t\}=P_1(t)P_2(t)$$

$$P_C(t)=P(T_1+T_2\leqslant t)=\int_{t_2=0}^{t}\int_{t_1=0}^{t_2}\mathrm{d}P_1(t_1)\mathrm{d}P_2(t_2-t_1)$$

结合优先与门运算，得到该动态子树的故障概率为

$$P_D(t)=P(T_1\leqslant T_2\leqslant t)=\int_{t_1=0}^{t}\int_{t_2=t_1}^{t}\mathrm{d}P_C(t_2)\mathrm{d}P_H(t_1)=$$

$$\int_{t_1=0}^{t}(P_C(t)-P_H(t_1))\mathrm{d}P_H(t_1)$$

子树的平均无故障时间：$MTBF=\int_0^\infty P_D(t)\mathrm{d}t$，其中，积分式的实现采用微分逼近算法($k$ 为逼近算法中的阶数)：

$$\int_{x_1}^{x_2}f(x)\mathrm{d}x\approx\sum_{n=0}^{k}\{f(x_1+n(x_2-x_1)/k)(x_2-x_1)/k\}$$

2.3　系统故障树可靠性分析

系统模块化故障树如图 5 所示。

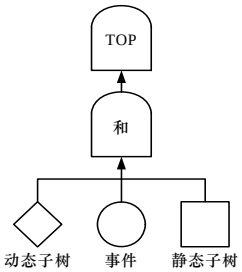

图 5　系统模块化故障树

整个系统故障树由动态子树、单独底事件和静态子树通过逻辑与门输出，通过引进的静态树顶事件逼近算法，可得整个系统的可靠性指标。整个系统的故障概率为

$$P_T(t)=P\{T_1\leqslant t,T_2\leqslant t,T_3\leqslant t,T_4\leqslant t\}=P_D(t)P_f(t)P_S(t)$$

整个系统的平均寿命为 $MTBF=\int_0^\infty P_T(t)\mathrm{d}t$

对于模块化后的故障子树，须考查其在系统中的概率重要度：

$$I_i^T=\sum_p P(S_p),(S_p\overset{i}{\rightarrow}S_q,S_p\in W,S_q\in F,\forall p\in N)$$

其中，系统正常状态集和故障状态集分别为 W,F，系统模块 i 为系统状态 S_p 转移到系统状态 S_q 的条件。因此，图 5 中模块概率重要度为

$$I_m^T=\sum_p P(S_p)=(1-p_m(t))\sum_{n\neq m}P_n(t)$$

3　高速铁路列车运行控制系统可靠性分析

高速铁路列车运行控制系统中，联锁子系统(IL)、列车调度子系统(CTC)、报文接口子系统、微机监测(MC)窗口以及维护窗口等决定了列车运行控制系统的输入失效空间。其中，联锁、列车调度子系统采用了双机热备结构；报文接口子系统采用了优先级控制。根据列车运行控制系统失效机制，建立该系统的动态故障树，并通过 DFLM 算法模块化获得该故障树的约简模型，系统动态故障树如图 6 所示。

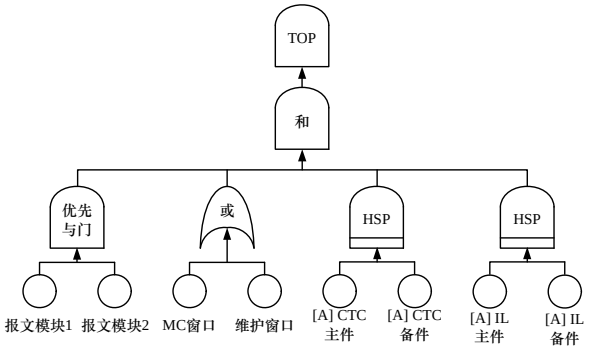

图 6　系统动态故障树

该故障树可由 IL 动态故障子树模块、CTC 动态故障子树模块、窗口服务静态故障子树模块和报文接口动态故障子树模块通过逻辑与门构成。

考虑系统可修复特性：根据引入的系统部件马尔可夫模型，系统部件故障概率分布为

$$P_i(t)=\frac{\lambda_i}{\lambda_i+\mu_i}-\frac{\lambda_i}{\lambda_i+\mu_i}\mathrm{e}^{-(\lambda_i+\mu_i)t}$$

定义系统故障树中第 i 个模块子树下的第 j 个底事件故障概率为 P_{ij}。

分析窗口静态模块子树：根据静态模块子树顶事件分析方法，得到窗口服务模块的可靠性指标。

子树故障概率为

$$P_2(t)=P_2\{\min(T_1,T_2)\leqslant t\}=P_{21}(t)+P_{22}(t)-P_{21}(t)\times P_{22}(t)$$

子树平均寿命为 $MTBF_2=\int_0^\infty P_2(t)\mathrm{d}t$。

分析动态模块子树：根据动态顶事件逼近算法分别可得当前可修复系统中各动态模块子树的可靠性指标。

接口模块子树故障概率为

$$P_1(t)=P(T_1\leqslant T_2\leqslant t)=\int_{t_1=0}^{t}\int_{t_2=t_1}^{t}\mathrm{d}P_{12}(t_2)\mathrm{d}P_{11}(t_1)=$$

$$\int_{t_1=0}^{t}(P_{12}(t)-P_{11}(t_1))\mathrm{d}P_{11}(t_1)$$

接口模块子树平均寿命为 $MTBF_1=\int_0^\infty P_1(t)\mathrm{d}t$。

IL 模块子树故障概率为

$$P_3(t)=P\{\max(T_1,T_2)\leqslant t\}=P_{31}(t)P_{32}(t)$$

IL 模块子树平均寿命为 $MTBF_3=\int_0^\infty P_3(t)\mathrm{d}t$。

CTC 模块子树故障概率为

$$P_4(t)=P\{\max(T_1,T_2)\leqslant t\}=P_{41}(t)P_{42}(t)$$

CTC 模块子树平均寿命为 $MTBF_4=\int_0^\infty P_4(t)\mathrm{d}t$。

分析整个系统可靠性指标:由各模块静态逻辑组合方式

和顶事件概率算法，可得系统故障概率为

$$P(t)=P(T_1\leqslant t,T_2\leqslant t,\cdots,T_4\leqslant t)=\prod_1^4 P_j(t)$$

系统平均寿命为 $MTBF_T=\int_0^\infty P_T(t)\mathrm{d}t$ 。

系统模块的概率重要度为

$$I_m^T=\sum_p P(S_p)=p_m(t)\sum_{n\neq m}(1-P_n(t))$$

根据列车运行控制系统的技术大纲和安全级别，取各部件的失效率和修复率为：IL 系统(2×10^{-9}, 4×10^{-3})，CTC 系统(2×10^{-9}, 2×10^{-3})，报文接口系统(4×10^{-9}, 3×10^{-3})，窗口系统(2×10^{-9}, 1×10^{-3})。通过计算机辅助计算，当逼近阶数分别为 100, 500 时，上述改进的 DFTA 方法实现了高速铁路控制系统可靠性定性、定量的分析，运算复杂度与各子树模块相关，系统的组合状态数与故障树的部件数以及单一部件的多态性相关，分析结果如表 1 所示，其中，*FT* 为故障树名；*Step* 为逼近阶数；*EMTBF* 为平均寿命；*EMI* 为模块重要度；*ECR* 为逼近算法复杂度；*CSC* 为组合状态数。

表 1　系统可靠性分析结果

FT	*Step*	*EMTBF*	*EMI*	*ECR*	*CSC*
接口	100	10.743 07	0.276 71	3	4
接口	500	11.802 46	0.294 13	3	4
窗口	100	11.850 07	0.250 86	3	4
窗口	500	13.873 59	0.250 21	3	4
CTC	100	12.750 09	0.233 15	3	4
CTC	500	14.714 08	0.239 52	3	4
IL	100	12.424 08	0.239 27	3	4
IL	500	15.799 03	0.219 72	3	4
系统	100	24.320 77	0.998 99	22	256
系统	500	29.440 26	0.999 98	22	256

可以看出，采用逼近算法分析安全苛求系统的可靠性，其复杂度是各子模块和系统模块复杂度的累积，而传统故障树分析中的状态组合为各底事件的状态乘积，改进的模块化处理方法能在安全苛求系统的可靠性分析中高效的实现，且在充分考虑苛求系统的多态性和可修复性时，改进的 DFTA 方法通过逼近算法中复杂度累加，避免了多部件多状态系统中指数级状态组合爆炸问题。

不可靠度随系统运行时间的变化曲线如图 7 所示。结合图 7 和各模块的特性参数可以预知，逼近算法中逼近阶数的增长使得分析结果趋近于理想值，系统模块重要度由其本身的故障概率、平均寿命等决定，且系统整体的可靠性指标与系统部件的形式化逻辑组织形式相关联，改进的 DFTA 方法对于多态失效模式能给出可行的可靠性分析方案。

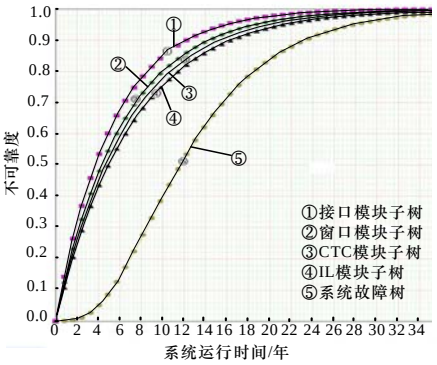

图 7　不可靠度随系统运行时间的变化曲线

4　结束语

本文通过在系统可靠性分析中引入系统部件的多态冗余性、可修复性等安全苛求特性，建立改进的基于可修复因子的动态故障树分析方法，引入顶事件逼近方法，能高效地实践安全苛求系统可靠性指标分析过程。下一步的研究方向是分析可修复性参数分布对于系统可靠性[5-6]的影响。

参考文献

[1] Dugan J B, Coppit D. Developing a Low-cost High-quality Software Tool for Dynamic Fault-tree Analysis[J]. IEEE Transactions on Reliability, 2000, 49(1): 49-59.

[2] Dugan J B, Bavuso S, Boyd M. Dynamic Fault Tree Models for Fault Tolerant Computer Systems[J]. IEEE Transactions on Reliability, 1992, 41(3): 263-377.

[3] Dugan J B, Bavuso S J. Fault Trees and Markov Models for Reliability Analysis of Fault-tolerant Digital System[J]. Reliability Engineering and System Safety, 1993, 39(1): 291-307.

[4] Amari S, Dill G, Howald E. A New Approach to Solve Dynamic Fault Tree[C]//Proc. of Annual Reliability and Maintainability Symposium. Tampa, Florida, USA: IEEE Press, 2003.

[5] Marko C, Mavko B. A Dynamic Fault Tree[J]. Reliability Engineering and System Safety, 2002, 75(1): 83-91.

[6] Blanks H S. Quality and Reliability Research into the Next Century[J]. Quality and Reliability Engineering International, 1994, 10(3):179-184.

编辑　顾姣健

(上接第 99 页)

5　结束语

本文提出 TT-Apriori 算法，先从最有可能包含最大频繁项集的所有事务中进行搜索，再从各事务的子集中进行搜索，避免遗漏。且当最小支持度与事务数据库更新时无须重新扫描数据库，根据算法 1 中已得的各项集支持度 *s*[*i*]值即可容易地挖掘到新的频繁项集；当事务数据库添加新的事务时，只需增加的事务树的节点即可，具有一定的实用意义。

参考文献

[1] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases[C]//Proc. of ACM-SIGMOD'93. Washington D. C., USA: ACM Press, 1993.

[2] Agrawal R, Aggrawal C, Pasad A V V V. Tree Projection Algorithm for Generation of Frequent Itermsets[J]. Journal of Parallel and Distributed Computing, 2001, 61(3): 350-371.

[3] Burdick D, Calimlim M, Mafia J G. A Maximal Frequent Itemset Algorithm for Transactional Database[C]//Proc. of the 17th Int'l Conf. on Data Engineering. Heidelberg, Germany: [s. n.], 2001.

[4] 胡　斌, 蒋外文. 基于位阵的更新最大频繁项集算法[J]. 计算机工程, 2007, 33(2): 59-61.

[5] 冯　洁, 陶宏才. 一种频繁项集的快速挖掘算法[J]. 微计算机信息, 2007, 23(6): 164-166.

编辑　金胡考