追踪复杂事件的形成世系

熊招招 王永利
(南京理工大学计算机科学与技术学院 南京 210094)

摘 要 复杂网络安全事件、物联网世系追溯等新型应用为复杂事件的世系研究提出许多挑战。由于模糊时间以及状态不确定性转移等因素的存在,追溯复杂事件的世系时往往出现时间推导不精确以及无法有效逆向推导等问题,因此无法高效地追踪及查询复杂事件的形成世系。针对此类问题,结合起源语义提出了一种基于扩展的模糊时间 Petri 网的逆向推理模型(BREFTN),并根据时间自动机理论利用此模型设计了逆向推理算法。在给定目标库所以及相关条件的情况下,它不仅可以得到所有演变路径信息并分析其可能性分布,还可以对复杂事件的各个状态及变迁的模糊时间函数值进行有效地推算分析。最后分析了 BREFTN 模型的完备性及演变路径的性质,并通过实验检测了算法的性能。

关键词 复杂事件,世系,模糊时间 Petri 网,时间自动机
中图法分类号 TP311 **文献标识码** A

Tracing the Evolution Lineage of Complex Event

XIONG Zhao-zhao WANG Yong-li
(Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract Some novel applications such as network security event and tracing lineage of IOT, etc. presented many challenges for the lineage studying of complex event. Because of the presence of factors including fuzzy time and uncertain transfer of state, tracing the lineage of complex event often encounter inaccurate derived time and hard reverse derivation problem, therefore the evolution lineage of complex event can not be traced or queried effectively. A reverse derivation model with the provenance semantic, called BREFTN (backward reasoning extended fuzzy time petri net), was proposed for such problems, and based on this model, a backward reasoning algorithm according to time automation theory was designed. Given goal place(s) and other conditions, it can not only get all information evolution path and analyze the possibility distribution of path, but also efficiently compute the fuzzy time function value of the state and transition of complex events. Finally, the completeness of BREFTN model and the properties of evolution path were analyzed, and the performance of algorithm was verified by experiments.

Keywords Complex event, Lineage, Fuzzy time Petri net, Time automation

1 引言

有关起源的信息可以验证科学应用所产生结果的正确性,还可确定科学结果的可信度(包括内容的数量和质量)[1]。数据的产生及随时间推移而演化的过程信息称为数据世系或起源(data lineage)。数据世系包含静态的源数据信息和动态的数据演化过程[2,3],主要有 when 源、where 源、how 源、why 源以及 what 源等语义。

复杂事件处理(CEP)是对基本事件的与、或、非及时间和顺序等方面的多次复合,能够表达上层应用丰富的逻辑语义需求,属于智能信息处理和信息融合的范畴,也是物联网的关键技术之一[4]。目前,CEP 研究主要集中于网络复杂安全事件、传感器网络(RFID 复杂事件)等应用;而复杂事件的起源更关注事件的状态变迁过程以及相关变迁模糊时间等问题,主要解决 when 源、how 源问题。计算机取证研究是其典型应用,但在完善的形式化建模分析方法出来之前,想要分析被攻击系统内部事件状态变迁仍是件很困难的事情;利用 BREFTN(Backward Reasoning Extended Fuzzy Time Petri Net)模型,能为此类事件状态变迁提供有效的模糊性分析。

通过这种模型,可以更好地理解复杂事件状态信息,包括起源信息等,其贡献如下:

1)对复杂事件演变过程中各状态变迁的信息及追溯操作进行形式化描述,明确了各状态之间关联(如模糊时间戳)的关系;

到稿日期:2011-08-11 返修日期:2011-11-23 本文受国家自然科学基金(60803001),中国博士后科学基金特别资助项目(200902517),江苏省博士后基金(0801043B),江苏省高校 2010 年"青蓝工程"优秀青年骨干教师项目,南京市科技计划重点项目(020142010)以及南京理工大学 2009 年紫金之星资助。

熊招招(1987—),男,硕士生,主要研究领域为复杂事件处理、数据挖掘等,E-mail:shelleyhacker@21cn.com;王永利(1974—),男,博士,副教授,CCF 会员,主要研究领域为数据库技术、情境感知、物联网数据处理、模式识别等。

2)可以对复杂事件演变过程做更深入的分析，包括提出了具有模糊时间属性的演变路径追溯算法，并支持对各路径进行可能性分布度量等工作。

本文第2节介绍BREFTN模型、起源信息以及算法相关定义；第3节介绍BREFTN模型相关算法的具体实现；第4节针对一个具体的案例应用，利用BREFTN模型算法来解决相关问题；第5节介绍相关工作；最后提出相关不足之处并介绍未来工作。

2 BREFTN模型

2.1 BREFTN定义

BREFTN是在EFTN[5,6]基础上改进而来的；对每个托肯和变迁均增加有效的时序约束描述，同时增加了库所与变迁的转移函数等内容。

定义1 逆向模糊时间高级Petri网是一个九元组，即BREFTN=$(P,T,I,O,FT,TOT,CT,AD,M_0)$。各元组定义如下：

1)P由一个库所的有限集合构成，$P=\{p_1,p_2,\cdots,p_n\}$；

2)T是有限的变迁集合构成，$T=\{t_1,t_2,\cdots,t_n\}$，$P\cup T\neq\Phi$且$P\cap T=\Phi$；

3)I为输入函数，即库所集到变迁集的一个映射，$I(t)$表示从P到T的库所集合；

4)O为输出函数，即变迁集到库所集的映射，$O(t)$表示T到P的库所集合；

5)TOT是托肯的有效模糊时间戳，$h[a,b,c,d]$，$0\leqslant a\leqslant b\leqslant c\leqslant d$，$0\leqslant h\leqslant 1$，其中$a,b,c,d$均为时间点，托肯到达时间处于$b$和$c$之间的概率值为$h$，之外的概率值则小于$h$；

6)FT是模糊时间戳集合，一个模糊时间戳是一个从时间刻度T到实数区间$[0,1]$的模糊时间函数，一个模糊时间戳与库所中的托肯相关；

7)CT是从变迁集T到模糊触发区间的映射函数，$CT:T\rightarrow[a,b,c,d]$，$0\leqslant a\leqslant b\leqslant c\leqslant d$；

8)AD是从变迁输出弧到模糊延迟的映射函数，$AD:T\times P\rightarrow[a,b,c,d]$，$0\leqslant a\leqslant b\leqslant c\leqslant d$；

9) M_0是初始标识函数，$M_0:P\rightarrow FT$是对系统动态行为的描述，用托肯的集合表示，即$\{(p,\pi(\tau))\mid p\in P,\pi(\tau)\in FT\}$，其中变迁的模糊触发区间$[a,b,c,d]$表示变迁触发的时间限制。

2.2 BREFTN模型中模糊时间相关计算方法

为了处理模糊时间信息，参照Murata[7,8]提出的模糊时间Petri网模型，模糊时间函数主要包括库所p_i的模糊时间戳$\pi_{p_i}(\tau)(i=0,\cdots,n)$、变迁$t$模糊使能时间$e_t(\tau)$、模糊触发时间$o_t(\tau)$以及模糊延迟$d_{tp}(\tau)$。下面仅列出部分模糊时间函数的公式，具体的latest和earliest[9]模糊时间函数计算方法参见相关文献。

(1)关于模糊触发时间，在变迁t没有变迁冲突的情况下，其模糊触发有效区间为$CT(t)=h(\pi_1,\pi_2,\pi_3,\pi_4)$，则变迁$t$的模糊触发时间为：

$$o(\tau)=e(\tau)\oplus h(\pi_1,\pi_2,\pi_3,\pi_4) \tag{1}$$

式中，变迁不存在冲突的情况，默认$h=1$，$\oplus$是文献[9]提及的扩展加法。如果存在m个变迁，$t_i,i=1,2,\cdots,m$，其各自的模糊使能时间为$e_i(\tau)$，相应模糊触发有效区间为$CT(t_i)=h_i[\pi_{i1},\pi_{i2},\pi_{i3},\pi_{i4}]$，则$t_i$的模糊触发时间为

$$o_i(\tau)=\mathrm{Min}\{e_i(\tau)\oplus h_i[\pi_{i1},\pi_{i2},\pi_{i3},\pi_{i4}],earliest\{e_i(\tau)\oplus h_i[\pi_{i1},\pi_{i2},\pi_{i3},\pi_{i4}]\}\},i=1,2,\cdots,i,\cdots,m \tag{2}$$

(2)关于模糊延迟时间$d_{tp}(\tau)$，用于求库所的模糊时间戳$\pi_{tp}(\tau)$，指某库所到达变迁t的输出库所p的时间可能性分布值，需要进行$\oplus$操作，即

$$\pi_{tp}(\tau)=o_t(\tau)\oplus d_{tp}(\tau)=(o_1,o_2,o_3,o_4)\oplus(d_1,d_2,d_3,d_4) \tag{3}$$

式中，$d_{tp}(\tau)$也可用$AD(tp)$来表示。

(3)关于可能性分布的概率计算见文献[7,8]。假定e和f分别代表可能性分布的隶属函数π_e和π_f，若$e\leqslant f$，则意味着e库所发生在f之前。在此引进ψ来代表$e\leqslant f$、$e\geqslant f$、$e=f$的置信度(可能性分布的概率)，公式为：

$$\psi(e\leqslant f)=\frac{Area([E,F]\cap\pi_e)}{Area(\pi_e)}$$
$$\psi(e\geqslant f)=\frac{Area([F,E]\cap\pi_e)}{Area(\pi_e)} \tag{4}$$
$$\psi(e=f)=\frac{Area(\pi_f\cap\pi_e)}{Area(\pi_e)}$$

2.3 BREFTN模型相关定义

基于BREFTN模型设计的逆向推理算法，可以更直观地描述复杂事件演变过程并度量各状态(包括库所和变迁)之间的关联。为了方便描述算法，需要定义如下概念。

定义2 前关联库所集PRPS(Previous Relation Places Set)，所有满足$p_i=I(t_i)$，即变迁t_i的输入库所的集合，可用$PRPS(t_i)$表示；后关联库所集NRPS(Next Relation Places Set)，所有满足$p_i=O(t_i)$，即变迁t_i的输出库所的集合，用$NRPS(t_i)$表示；$|PRPS(t_i)|$和$|NRPS(t_i)|$为集合元素个数。

定义3 $ITS(p_i)$(Input transition set)，表示对于一个p_i来说，所有满足$p_i\in PRPS(t_i)$的变迁t_i的集合；$OTS(p_i)$(Output transition set)，表示对于一个p_i来说，所有满足$p_i\in NRPS(t_i)$的变迁t_i的集合。

定义4 (库所节点的建立)$n_i(p_i,\pi_{p_i}(\tau),ITS(p_i),OTS(p_i))$，类似于文献[10]，一个库所节点为四元组，其中$\pi_{p_i}$表示库所托肯到达的模糊时间戳。在此$ITS(p_i)$是库所$p_i$的输出变迁的集合，$OTS(p_i)$则是输入变迁的集合；$|ITS(p_i)|$和$|OTS(p_i)|$分别表示集合元素的数目。

定义5 未处理节点集合NPNS(Not Processed Nodes Set)，即模糊时间戳未知的那些节点的集合；相对应的是已处理节点集合HPNS(Have Processed Nodes Set)；种子节点集INS(Initialized Nodes Set)，即初始阶段已知模糊时间戳的节点集合。

定义6 变迁路径集合TPS(Transition Paths Set)，其中$TPS_i=\{(t_j,h_j,\cdot t_j)\},i,j=1,2,\cdots,n,h_j$为层数，$\cdot t_j$表示变迁$t_j$相应的直接前继变迁集合；演变路径集合EPS(Evolution Paths Set)，$Epath_i=\{\{P_{start}\}(t_s,h_s,\Phi)\{P_s\},\cdots,(t_j,h_j,\cdot t_j)\{P_j\},\cdots,(t_n,h_n,\cdot t_n)\{P_{target}\}\},i=1,2,\cdots,n$，其中路径$Epath_i\in EPS$。$\{P_j\}$表示当前变迁的后继库所集合，$\{P_{start}\}$和$\{P_{target}\}$分别为初始库所和目标库所集合；变迁路径集合$TPS$需要最终转化为$EPS$。

定义7 IOP(Information of Path)，表示当前库所(除了初始库所)和变迁路径信息，$IOP(p)$或$IOP(t)=\{TPS_i\}$，同

时，$|IOP|$表示路径个数；路径合并符号"∪"：$IOP' \cup IOP'' = IOP'''$以及$TPS' \cup TPS'' = TPS'''$，后者表示两个路径中元素融合到新的路径中去。

定义 8 演变路径的层次 h，从逆向模糊时间 Petri 网的推理过程来看，引入对应的层次概念[11]，能更好地描述演变路径中各变迁之间的关联状态，其反映了演变状态的深度。如图 1 所示。

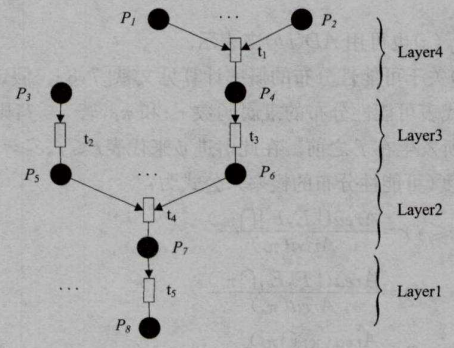

图 1 逆向演变路径层次示意图

通常，为了方便算法中逆向推理的实现，将目标库所以及相应的输入变迁定为第 1 层。

2.4 BREFTN 模型中的数据起源

数据世系（起源）不仅与数据本身信息有关，也与如何创造数据的过程有关[2]。起源的 7W 模型[12]指出了数据起源信息应该包括 who 源、when 源、where 源、how 源、which 源、what 源、why 源这 7 部分，其核心是 what 源，它主要记录数据生命周期内的各种事件来描述数据的各种信息。用树状图表示起源元素关联[2]，如图 2 所示。

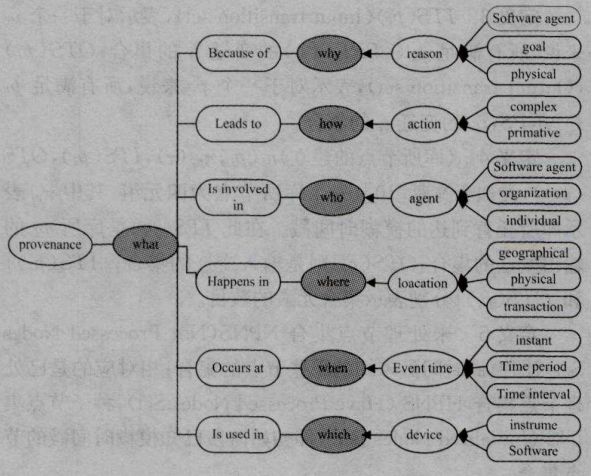

图 2 数据起源信息的内容

在 BREFTN 模型中，复杂事件内部各状态及其变迁中的模糊时间和事件演变路径的产生及描述等问题，主要对应于 7W 模型中的 how 源和 when 源问题；Petri 网中库所是托肯的状态集合，变迁则描述状态的转移动作，而在 BREFTN 模型中的 how 源主要体现于库所与变迁之间及各自的关联。因为在给定目标库所的情况下，需要模型能够产生并描述所有到达既定状态的形式化的解，这里用定义 6 的演变路径集合 EPS 表示；在模糊时间 Petri 网中，托肯的模糊时间戳以及变迁的模糊时间函数等，则需要模型提供有效的形式化描述，并根据已知条件进行推理计算，即需解决 BREFTN 模型的 when 源问题，如表 1 对应的关系。

表 1 BREFTN 模型 how 源及 when 源对应关系

起源语义	BREFTN 模型	推理结果
how 源	EPS(evolution path set)	复杂事件形成途径
when 源	$\pi_p(\tau), e_t(\tau), o_t(\tau), d_{tp}(\tau)$ 等	复杂事件各环节时间

2.5 BREFTN 模型相关性质与定理

在 BREFTN 模型中，若将 Petri 网中的库所和变迁处理为顶点，则可将其转化为有向无环图（DAG 图）。此模型具有如下性质。

性质 1 BREFTN 模型完备性

当有向无环从目标库所开始，反复推理计算每个前继关联变迁及库所，构造逆向演变路径集合 EPS。采用类似白色路径的着色方法，为库所和变迁着色，表示推理的状态。开始时，每个顶点 v（库所和变迁）的颜色为白色，搜索中被发现时置为灰色，记录下时间戳 $d[v]$；当 v 检查结束时，置为黑色，记录下第二个时间戳 $f[v]$。采用这种时间戳的方式，可以保证每个节点均被访问过，故当搜索结束时，所有路径的解都将被穷尽，从而保证了 BREFTN 模型的完备性。

引理 1 白色路径定理

在一个 BREFTN 有向无环图 G 的深度优先森林中，顶点 v（库所和变迁）是顶点 u 的后裔，当且仅当在搜索过程中于时刻 $d[u]$ 发现 u 时，可以从顶点 u 出发，经过一条完全由白色顶点组成的路径到达 v。

证明（充分性） 假设 v 是 u 的后裔，设 w 是深度优先树中 u 和 v 之间通路上的任意顶点，这样 w 就是 u 的后裔，可知 $d[u]<d[w]$，因此在 $d[u]$ 时刻由于 w 还未被访问，故 w 被标记为白色。

（必要性） 假设在 $d[u]$ 时刻，顶点 v 沿着一条仅由白色顶点组成的通路可达 u。但在深度优先树中，v 没有成为 u 的后裔。不失一般性，假定该通路上的其他每一个顶点都成为了 u 的后裔，否则可以设 v 是该通路中最接近 u 的，且不为 u 的后裔的顶点。设 w 为该通路上 v 的前驱，这样 w 就是 u 的后裔。根据推论，$f[w] \leqslant f[u]$，v 必须在 u 被发现之后、在 w 完成之前被发现，于是有 $d[u]<d[w]<f[w] \leqslant f[u]$。证毕。

定理 1 基于 BREFTN 模型得到的演变路径不存在回路。

证明：当前仅当该 BREFTN 的深度优先搜索没有产生反向边（深度优先树中，连接顶点 u 到它的某一祖先顶点 v 的那些边）。假设 BREFTN 模型得到的演变路径有一个反向边 (u,v)，则在深度优先森林中，顶点 v 为 u 的祖先，则必然存在一条路径（通路），此路径与反向边 (u,v)，形成一个回路。证毕。

3 BREFTN 逆向推理模型算法

3.1 BREFTN 模型推理算法描述

此模型由两个子算法组成。其中，BREFTN 逆向推理算法（算法 1）的基本思想是，步骤 1 为初始化算法中各数据集；步骤 2 从目标库所开始，通过遍历算法（类似 DFS）反复推理计算每个前继关联变迁及库所，最终得出每条演变路径（此路径是变迁三元组的集合），其中函数 DFS_Visit_P(p) 和 DFS_Visit_T(t,h_t) 分别为访问库所和变迁节点的过程，而函数

Path_Handle($IOP(t)$,$IOP(p_x)$,h_c)则是处理子路径的过程；步骤 2 得到的中间路径，由步骤 3 处理为完整演变路径集 TPS；最后 TPS 中的各路径通过算法 2 分别计算各库所和变迁模糊时间函数值以及各路径对应的可能性分布，并进行有效性分析。

算法 1 BREFTN 模型逆向推理算法

输入：目标库所(集合)，变迁有效时间间隔(CT(t)) 以及模糊时间延迟(AD(tp))，已知节点集合(INS)；

输出：每个 Epath 中库所和变迁的模糊函数值，以及该路径的可能性分布值；

符号定义："←"为赋值操作，"⇐"为插入到集合操作；

步骤 1 初始化 PRPS(t) 和 NRPS(t)，建立相应的节点 $n_i(p_i,\pi_{p_i}(\tau),ITS(p_i),OTS(p_i))$；

for each p_i,NPNS ⇐n_g；

Set EPS=HPNS=TPS=IOP(t)=IOP(P)=Φ。

步骤 2 Set h_c=PNum=0；

Start do-while loop if HPNS≠Φ and NPNS no longer increases {

For eachp∈P,t∈T {

color[p]←white;color[t]←white;}

For each $n_i\in$ NPNS and $p_i\in P$ {

if color[p_i]=white then DFS_Visit_P(p_i);PNum ←|IOP(p_i)|;}

函数 DFS_Visit_P(p) {

Set h_t←0;color[p]←gray;

d[p]←time←time+1;h_t←h_c←h_c+1;

For each $t_n\in$OTS(p) {

If color [t_n]=white then DFS_Visit_T(t_n,h_t);

IOP(P)⇐IOP(P)⋃IOP(t);

}

color[p]←black,f[p]←time←time+1;

}

函数 DFS_Visit_T(t,h_t) {

h_c←h_t;color[t]←gray;d[t]←time←time+1;

for each $p_x\in$PRPS(t) {

if $p_x\notin$INS and $n_x\notin$NPNS{NPNS⇐n_x;

if color[p_x]=white then DFS_Visit_P(p_x);

Path_Handle(IOP(t),IOP(p_x),h_c);}

else if $p_x\in$INS and $n_x\notin$NPNS {HPNS⇐n_x;

if IOP(t)=Φ and |PRPS(t)|=1 then

IOP(t)⇐(t,h_c,Φ);} }

color[t]←black;f[t]←time←time+1;

h_c←h_c−1

}

函数 Path_Handle(IOP(t),IOP(p_x),h_c){

Set n_1=|IOP(t)|;n_2=|IOP(p_x)|;

If n_1=0 then IOP(t)⇐IOP(p_x);

Else then {

for each i∈n_1,$TPS_i\in$IOP(t) and j∈n_2,$TPS_j\in$IOP(p_x) {

if TPS_i exits then delete it;

IOP(t)⇐((TPS$_i$⋃TPS$_j$)⇐(t,h_c,$\cdot$t));} }

}

步骤 3 for each $TPS_r\in$TPS{为每个 TPS_r 建立对应 $NPNS_r$(⊂NPNS),$HPNS_r$(⊂HPNS);

For each $(t_j,h_j,\cdot t_j)\in TPS_r$ {

{p_j}←NRPS($\cdot t_j$)∩PRPS(t_j);

If {n_j}∈HPNS then $HPNS_r$⇐{n_j};

else $NPNS_r$⇐{n_j} }

$Epath_r$←

{{P_{start}}(t_s,h_s,Φ){P_s},…,(t_j,h_j,$\cdot t_j$){P_j},…,(t_n,h_n,t_n){P_{target}}}

由步骤 3 得到的 EPS 演化路径集合将作为算法 2 的输入数据做下一步处理；

步骤 4 经算法 2 计算，输出 EPS 以及每个演化路径中的库所和变迁的模糊时间函数值，以及该路径的可能性分布值。

算法 2 演化路径集合(EPS)处理算法

输入：EPS

输出：每个演化路径中的库所和变迁的模糊时间函数值，以及该路径的可能性分布值

Start do-while loop if $NPNS_r=\Phi$ {

For each $n_x\in NPNS_r$ $\exists n_{jk}\in HPNS_r$,

$\forall t_{aa}\in OTS(p_x)$ and $t_{aa}\in ITS(p_{jk})$ {

$e_{t_{aa}}$←latest{$\pi_{p_{jk}}(\tau)$,j=1,2,…,m};

If t_{aa} exits structural conflict then

$o_{t_{aa}}$←$e_{t_{aa}}\oplus CT(t_{aa})$ (式(1))；

Else$\forall t_{aa}\in\{t_{bi}\}$,(i=1,…,n)and $t_{bi}\in ITS(p_{jk})$then

$o_{t_{aa}}$←Min{$e_{t_{aa}}\oplus CT(t_{aa})$,earlist{$e_{t_{bi}}\oplus CT(t_{ai})$,…}},i=1,2,…,n;(式(2))

}

$\pi_{p_x}(\tau)$←$o_{t_{aa}}\oplus AD(t_{aa}p_x)$ (式(3))；

update n_x and $HPNS_r$⇐n_x and delete from $NPNS_r$；

}

最后根据式(4)求出各路径可能性分布值。

3.2 BREFTN 模型算法 1 的时间复杂度分析

针对算法 1 的主要部分(步骤 2)，假设 BREFTN 模型中 Petri 网的节点(库所和变迁)数为 n，边数为 m；算法类似于 DFS 深度搜索过程，初始每个节点需要预设为白色，第一次访问被置为灰色，执行占用时间为 $\Theta(n)$；对节点的深度搜索主要是通过函数 DFS_Visit_P(p)(访问库所节点)和函数 DFS_Visit_T(t,h_t)(访问变迁节点)两者相互调用进行。前者最多$|OTS(p)|$次，后者则最多$|PRPS(t)|$次，理论上执行时间为$\sum_{p\in P}|OTS(p)|+\sum_{t\in T}|PRPS(t)|\geqslant\Theta(m)$。由于只访问白色节点，因此实际执行时间为 $\Theta(m)$，即函数调用次数与边数有关；模型中算法从访问库所过程结束返回到访问变迁过程中，即从 DFS_Visit_P(p)返回到 DFS_Visit_T(t,h_t)。当变迁输入库所大于 1 个，即$|PRPS(t)|>1$时，需要调用路径来处理函数 Path_Handle。由于需要同时访问集合 $IOP(t)$以及 $IOP(p_x)$内部循环，假设$|IOP(t)|=n_1$，$|IOP(p_x)|=n_2$，且 $n_1<n$,$n_2<n$，则聚集运算的复杂度为 $\Omega(n_1*n_2*|T|)$，上限为 $O(n^3)$，其中$|T|(<n/2)$为变迁节点数量；算法在最好的情况下，满足$|OTS(p)|=|PRPS(t)|=1$，则此算法总执行时间为 $\Theta(n+m)$；最坏情况下所有变迁节点均需要调用路径处理过程，执行时间为 $\Theta(n+m)+O(n^3)$ 。

3.3 BREFTN 模型算法 1 与广度优先搜索算法(BFS)比较

BREFTN 模型的算法 1 类似于 DAG 图的遍历过程，算法本身就是对深度优先搜索 DFS 的改进。在以往的 Petri 网模型的遍历方法中，比较常用的是广度优先搜索(BFS)。根据 3.2 节对算法 1 的复杂度分析，再假定 BREFTN 模型中节点排列类似于满二叉树的情形下(目标节点只有 1 个)，比较两种算法在不同节点数的情况下的算法响应时间，其仿真结果如图 3 所示。

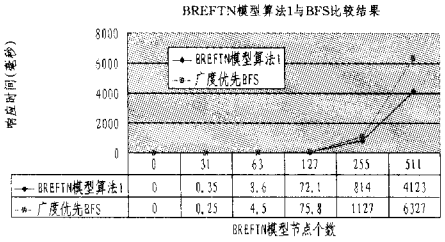

图 3 BREFTN 算法 1 与 BFS 比较

由图 3 可得，当节点数较少时(127 左右)，BFS 的性能略好于算法 1；但当节点数较多时，算法 1 的性能明显好于前者。

4 蠕虫病毒入侵实验具体案例

此处将介绍一种蠕虫病毒的入侵过程。Linux. Adore. Worm 是一种利用 Linux 系统漏洞并能自我恶意传播的蠕虫[13]。为了建立入侵模型，需要必要的信息数据以及这些数据之间的关联信息。一般来说，入侵过程中关键数据体现在系统内存的进程和网络信息中、Rootkit 信息以及完整性检查方面[14]。蠕虫入侵行为语义信息详见文献[13]。

先对利用本文模型中的算法对具体案例进行仿真计算并分析结果，最后针对仿真结果分析比较算法的相关性能。仿真程序在 Matlab 环境下完成。

实验环境：OS：Fedora core Linux machine (2.6.23-14 SMP)、Windows2007，CPU：Intel 酷睿 2Q8300 2.5GHz，主存：8G。实验默认设置：变迁冲突概率 h 误差控制在 0.3 以内，蠕虫传播时耗误差不超过 20s 等。

4.1 蠕虫入侵取证起源语义

计算机取证中的推理自动化，可以通过快速提供形式化结果来优化分析过程，同样推理过程往往是精简过的[15]。类似于文献[14]，可以通过 6W1H 来表述这些抽象过程的信息。表 2 给出了案例中取证信息与起源语义的对应情况。

表 2 蠕虫入侵取证信息与起源语义对应关系

	证据		推理结果
	起源语义	系统信息	
who	入侵者信息(蠕虫特征等)	入侵者(蠕虫)行为日志	蠕虫特征信息
when	入侵过程各环节发生时间	入侵时间记录	蠕虫入侵时间
where	网络发起点、IP以及端口等	系统连接日志	蠕虫入侵连接信息
how	入侵过程各环节(事件行为)	所有入侵行为(日志文件)	蠕虫入侵途径
what	入侵过程发生事件	相关文件创建操作记录	系统破坏或文件创建
why	入侵事件前提条件	网络可连接，端口开放等	入侵有利条件(系统漏洞等)
which	入侵工具或软件应用	蠕虫后门或系统调用程序	入侵辅助条件

4.2 推理知识库的建立

要进行下一步的模型推理过程，需要将蠕虫入侵信息转换成基本的 BREFTN 形式。实际上，蠕虫可实施的入侵方式不止一种，即抽象出来的 Petri 网模型中将会有多种目标状态库所等。为了方便本文 BREFTN 模型演算，图 4 给出蠕虫入侵 how 语义推理示意情况。

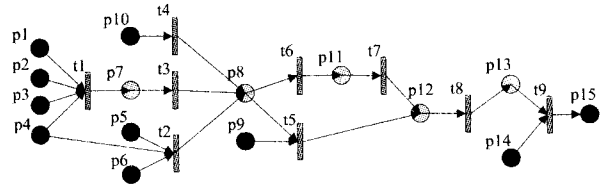

图 4 Linux. Adore. Worm 入侵的 how 语义推理形式

根据文献[13]，对应于图 4 中库所和变迁的语义信息如表 3 和表 4 所列。

表 3 库所的状态语义

库所	该库所的状态语义
p_1	主机存在 statd，lprng，wuftp 以及 bind 漏洞
p_2	启动补丁管理
p_3	主机网络可连接
p_4	go.163.com 站点正常
p_5	无可杀此蠕虫软件
p_6	打开防火墙屏蔽重要端口
p_7	蠕虫具有远程访问权限
p_8	具有远程权限并运行主程序
p_9	无进程异常监控
p_{10}	用户误操作
p_{11}	主机处于重启状态
p_{12}	漏洞已修补
p_{13}	蠕虫程序可访问 root shell
p_{14}	完成感染并远程传播

表 4 变迁的行为语义

变迁	该变迁的行为语义
t_1	利用主机漏洞进行攻击
t_2	下载压缩包到/usr/lib/lib 下
t_3	复制并安装压缩包到指定主机
t_4	下载蠕虫程序并运行
t_5	目标进程被替换为新的且具有木马特征
t_6	处理日常 script 脚本文件
t_7	主机启动脚本
t_8	替换系统 klogd
t_9	指定邮件服务器发送邮件，生成随机 B 类 IP 地址，并继续扫描漏洞

图 4 中实心圆表示初始库所和目标库所，其中 p_{15} 为目标库所(本文只考虑一个目标库所情况)，其他均为中间库所。各库所的托肯有效时间间隔如表 5 所列(库所 p_{15} 的托肯不考虑)。

表 5 托肯有效时间间隔情况(单位：s)

各库所对应托肯	token1，token2，token3，token4，token5，token6，token7，token8，token9，token10，token11，token12，token13，token14
有效时间间隔(TOT(p_i))	[0,0,0,0]，[0,0,0,0]，[0,0,0,0]，[0,0,0,0]，[0,0,0,0]，[0,0,0,0]，[0,2,4,6]，[3,5,7,8]，[0,0,0,0]，[0,0,0,0]，0.5[2,5,6,8]，0.5[3,6,7,10]，[2,3,5,7]，[0,0,0,0]

初始库所(种子库所)包括 p_1，p_2，p_3，p_4，p_5，p_6，p_9，p_{10}，p_{14}，其模糊时间戳分别为 $\pi_{p_1}(\tau)$，…，$\pi_{p_{14}}(\tau)=[0,0,0,0]$；变迁 t_1，…，t_9 的模糊触发时间间隔 $CT(t_i)$ 分别为[2,4,5,9]，[3,5,8,10]，[1,3,4,5]，[5,6,8,9]，0.6[4,6,7,9]，0.4[1,3,4,5]，[2,3,4,6]，[3,5,6,8]，[2,4,5,8]；变迁触发到库所模糊时间延迟包括 t_1p_7，t_2p_8，t_3p_8，t_4p_8，t_5p_{12}，t_6p_{11}，t_7p_{12}，t_8p_{13}，t_9p_{15}，其值分别为[12,15,17,20]，[19,24,27,29]，[4,9,11,13]，[21,23,32,35]，[23,25,26,28]，[14,16,19,21]，[9,11,12,15]，[17,21,23,26]，[12,15,17,20] (以上单位均为 s)。

该蠕虫完成传播并感染计算机，平均耗时为 145s 左右。

下面利用 BREFTN 模型求出蠕虫可能的入侵路径，并评估蠕虫在 145s 内完成入侵操作最有可能的路径情况。

4.3 实验步骤

4.3.1 仿真算法求出 Linux. Adore. Worm 入侵路径

由算法 1 的步骤 2 得出的路径集合 TPS 如表 6 所列。

表 6 由各路径计算结果

TPS	路径元素，按层次大小排列
TPS_1	$(t_9,1,\Phi),(t_8,2,t_9),(t_7,3,t_8),(t_6,4,t_7),(t_3,5,t_6),(t_1,6,t_3)$
TPS_2	$(t_9,1,\Phi),(t_8,2,t_9),(t_5,3,t_8),(t_3,4,t_5),(t_1,5,t_3)$
TPS_3	$(t_9,1,\Phi),(t_8,2,t_9),(t_7,3,t_8),(t_6,4,t_7),(t_4,5,t_6)$
TPS_4	$(t_9,1,\Phi),(t_8,2,t_9),(t_7,3,t_8),(t_6,4,t_7),(t_2,5,t_6)$
TPS_5	$(t_9,1,\Phi),(t_8,2,t_9),(t_5,3,t_8),(t_4,4,t_5)$
TPS_6	$(t_9,1,\Phi),(t_8,2,t_9),(t_5,3,t_8),(t_2,4,t_5)$
其余	略

由表 6 可知，Linux. Adore. Worm 的入侵路径有 6 条。分别处理每条路径，可得到相应的完全路径 $Epath$。以 TPS_1 为例：

$$Epath_1=\{\{p_1,p_2,p_3,p_4\}(t_1,6,t_3)\{p_7\},(t_3,5,t_6)\{p_8\},(t_6,4,t_7)\{p_{11}\},(t_7,3,t_8)\{p_{12}\},(t_8,2,t_9)\{p_{13},p_{14}\},(t_9,1,\Phi)\{p_{15}\}\}$$

4.3.2 求出每个路径的库所和变迁相应模糊时间函数值

按照算法 1 的步骤 3、步骤 4 以及算法 2，分别处理每条演变路径。以 $Epath_1$ 为例，如表 7 所列。

表 7 $Epath_1$ 中间库所以及目标库所情况

中间以及目标库所	模糊时间函数值
p_7	[14,19,22,29]
p_8	[19,33,41,53]
p_{11}	0.4[37,57,71,87]
p_{12}	0.4[50,76,93,116]
p_{13}	0.4[73,108,129,160]
p_{15}(目标库所)	0.4[89,130,156,195]

对其它路径进行类似的计算，获取各路径的目标库所情况，如表 8 所列。

表 8 各路径对应的目标库所计算情况

中间以及目标库所	模糊时间函数值
$Epath_1$	0.4[89,130,156,195]
$Epath_2$	0.5[88,123,141,173]
$Epath_3$	0.4[96,126,155,186]
$Epath_4$	0.4[92,126,150,181]
$Epath_5$	0.5[95,109,140,164]
$Epath_6$	0.5[91,118,135,159]

4.3.3 路径可能性分布

分别计算各路径的可能性分布，如图 5 所示，其中，以 $Epath_1$ 为例，根据式(4)可以得到：

$$\psi(\pi_{p_{15}}(\tau)\leqslant 145)=\frac{Area(ABEF)}{Area(ABCD)}=0.5379$$

而 $Possibility(Epath_1)=0.4\times 0.5379=0.2152$

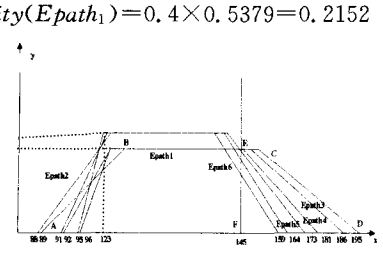

图 5 各路径可能性分布梯形图

同理可得其它路径情况，如表 9 所列。

表 9 各路径的可能性分布值

路径	各路径的可能性分布值
$Epath_1$	0.2152
$Epath_2$	0.3811
$Epath_3$	0.2286
$Epath_4$	0.2547
$Epath_5$	0.3496
$Epath_6$	0.4039

从结果中可以看出，蠕虫在 145s 内完成入侵最可能的路径是 $Epath_6$。

4.4 实验仿真

本文在给定条件包括蠕虫完成攻击事件为 145s 等情况下，通过修改相应模糊时间函数值，设定 200 次仿真计算，每隔 50 次对各路径的可能性分布函数值进行比较分析，得出图 6。其中横坐标为仿真计算次数，纵坐标为可能性分布函数值，其介于 0,1 之间。

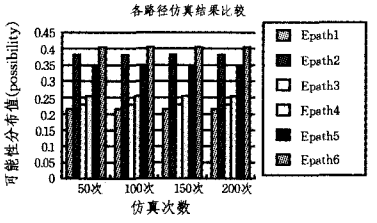

图 6 仿真各路径可能性分布比较示意图

多次仿真结果表明，其中路径 6 是蠕虫最有可能采取的入侵路径。

5 相关工作

基本 Petri 网中没有时间的概念，为了能建模和分析各种实时系统，各种类型经过扩展的时间 Petri 网均曾被提及过，包括文献[5-9]等。但很多实时系统具有时间的不确定性，例如类似于计算机取证分析系统中，病毒或犯罪宿主往往具有行为或时序的不确定性或随机性。

为了处理实时系统中时序的不确定性，Murata 提出了高级模糊时间 Petri 网(FTHNs)[8]，采取了模糊集理论来表达时间信息的不确定性或主观性。在 FTHNs 中，模糊时间是通过 4 种模糊时间函数或者可能性分布来表示的，包括模糊时间戳、模糊使能时间、模糊触发事件、模糊延迟。关于 FTHNs 的形式化定义以及计算和更新模糊使能时间以及模糊触发时间，在文献[8]中已经提及，但该模型缺乏对时序约束的有效描述，且在冲突的不确定性方面也缺乏定量分析。文献[5,6]提出了扩展的模糊时间 Petri 网(EFTN)，加入了变迁的时序约束的有效分析，包括增加变迁触发默认延迟为 $d_{tp}(\tau)=(0,0,0,0)$等内容。但 EFTN 模型缺乏支持逆向推理的方法，不能生成由源节点到结果节点的推理路径，并且还缺乏对库所的时序约束方法，无法适用于起源追溯等新型应用。

目前所提出的时间 Petri 网(FTPN)模型均无法解决以上提出的起源信息相关的问题，包括解决已知事件的历史推演过程，具体如状态变迁的模糊时序定量方法、事件推演路径可能性分布、冲突的不确定性度量等问题，即起源信息中的 when 源、how 源等问题。故在 EFTN 的基础上建立了逆向

(下转第 169 页)

5.4 集合特征码运算的综合实验

根据二维表(表2)以及这个二维表对应的全集、子集、运算要求,通过集合特征码的运算,得到符合要求的集合 X:

$$X=(\sim A)\cap(\sim P)\cap(\sim K)$$

$$C(X)=C(\sim A)\cap' C(\sim P)\cap' C(\sim K)$$

$$=\sim' C(A)\cap'\sim' C(P)\cap'\sim' C(K)$$

$$=01011110111\cap'00111001110\cap'11010111011$$

$$=00010000010$$

将该代码与全集人员姓名序列表进行对照,可以知道 $X=\{$杜小刚,王海$\}$。

实验分析:此例说明,集合运算可以在计算机中通过0、1运算实现。通过集合运算,可以实现数据库中的查询操作。

结束语 本文创新性地将二进制引进到集合运算过程中,提出集合特征码的概念,并定义了一系列的集合特征码运算规则,从而形成了一个较为完备的、形式化的集合特征码运算体系。在集合特征码的运算体系下,提出一系列的算法来实现这个运算体系。实验结果表明,通过集合特征码,集合运算可以在计算机中通过0、1运算实现,可以成功地实现数据库中的查询操作。下一步的工作就是将这个集合特征码运算体系加以完善,并深化对集合特征码的应用研究。

参 考 文 献

[1] 汪洋,张冠宇,史开泉. P-集合与F-记忆信息特性-应用[J]. 计算机科学,2011,38(2):246-249,266

[2] 于秀清. $P(\rho,\sigma)$-集合与它的随机特性[J]. 计算机科学,2010,37(9):218-221

[3] 宋笑雪,张文修. 基于集值决策属性的集值信息系统[J]. 计算机工程与应用,2007,43(17):8-10

[4] 史开泉. 函数s-粗集,函数粗集与信息系统规律拆分-合成[J]. 计算机科学,2010,37(10):1-10

[5] 刘伟斌,李天瑞,邹维丽,等. 特性关系粗糙集下属性值粗化细化时近似集增量更新方法研究[J]. 计算机科学,2010,37(6):248-251

[6] 宋笑雪,李鸿儒,张文修. 集值信息系统的知识约简与属性特征[J]. 计算机工程,2006,32(22):26-27

[7] 史开泉. P-集合与它的应用特征[J]. 计算机科学,2010,37(8):1-8

[8] 亓正坤,王廷明. 关于主范式的下标集合及其应用[J]. 青岛理工大学学报,2009,30(4):205-208

[9] 崔鹏,钱丽艳. 集合多覆盖问题的乘性权重更新分析[J]. 计算机科学,2007,34(10):219-220

[10] 崔鹏,刘红静. 测试集问题的集合覆盖贪心算法的深入近似[J]. 软件学报,2006,17(7):1494-1500

[11] 权光日,洪炳熔,叶风,等. 集合覆盖问题的启发函数算法[J]. 软件学报,1998,9(2):156-160

[12] 陈韬,吴志勇,孙乐昌,等. 集合覆盖问题的数据约简研究[J]. 计算机应用研究,2010,27(9):3307-3311

[13] 朱明,王俊普. 一种最优特征集的选择算法[J]. 计算机研究与发展,1998,35(9):803-805

(上接第153页)

推理的扩展模糊时间Petri网(Backward Reasoning Extended Fuzzy Time Petri Net,BREFTN)模型,有效解决了上述种种问题。

结束语 本文在原有研究的基础上,结合数据起源的相关研究,针对EFTN特性进行了改进,提出了模糊时间Petri网逆向推理模型(BREFTN)。利用模型的算法,有效地解决了此模型的逆向推理以及模糊时间计算问题。但是本文模型在描述有环的Petri网存在薄弱环节,相关的模型以及算法尚待改进,这是未来工作的重点。

参 考 文 献

[1] Tan W C. Provenance in databases: Past, current, and future[J]. IEEE Data Eng. Bulletin, 2007, 30(4): 3-12

[2] Simmhan Y L, Plale B, Gannon D. A Survey of Data Provenance Techniques[R]. Bloomington, Bloomington IN 47405: Computer Science Department, Indiana University, 2005

[3] 高明,金澈清,王晓玲,等. 数据世系管理技术研究综述[J]. 计算机学报,2010,33(3):373-389

[4] 刘强,崔莉,陈海明. 物联网关键技术与应用[J]. 计算机科学,2010(6):1-4,10

[5] Zhou Y, Murata T. Modeling and Performance Using Extended Fuzzy-timing Petri Nets for Networked Virtual Environment [J]. IEEE Transactions on System, Man, and Cybernetics-part B: Cybernetics, 2000, 30(5): 737-755

[6] Zhou Y, Murata T. Modeling and Analysis of Distributed Multimedia Synchronization by Extended Fuzzy-timing Petri Nets[J]. Transaction of the SDPS, 2001, 15(4): 130-141

[7] Murata T. Temporal Uncertainty and Fuzzy-timing High-level Petri Nets[C]// Proc. of the 17th International Conference on Application and Theory of Petri Nets. New York, USA: Springer, 1996: 11-28

[8] Murata T, Suzuki T, Shatz S. Fuzzy-timing High-level Petri Nets (FTHNs) for Time-critical Systems[M]. Fuzziness in Petri Nets, Studies in Fuzziness and Soft Computing, Physica-Verlag, 1999: 88-114

[9] Zhou Y, Murata T. Petri Net Model with Fuzzy-timing and Fuzzy-metric Temporal Logic[J]. International Journal of Intelligent Systems, 1999, 14(8): 719-7467

[10] 张稳,张桂成. 改进的基于规则的逆向模糊推理算法[J]. 通信学报,2008,29(2):101-105

[11] Yang R, Heng P A, Leung K S. Backward Reasoning on Rule-based System Modeled by Fuzzy Petri Nets Through Backward Tree[J]. FSKD, 2002: 18-22

[12] Ram S, Liu J. A new perspective on semantics of data provenace [C]// Proc. First International Workshop on the Role of Semantic Web in Provenance Management. Washington D. C., 2009

[13] Chien E. Linux. Adore. Worm [EB/OL]. http://securityresponse. symantec. com/avcenter/venc/data/linux. adore. worm. html, 2008-08

[14] Hwang H U, Kim M S, Noh B N. Expert System Using Fuzzy Petri Nets in Computer Forensics[J]. Lecture Notes in Computer Science, 2007, 4413(2007): 312-322

[15] Russell S J, Norvig P. Artificial Intelligence: a modern approach (2nd edition)[M]. Prentice-Hall, 2003