基于形式概念分析的语义角色挖掘算法

周 超[1,2,3,4] 任志宇[1,2,3] 毋文超[1,2,3]
(信息工程大学 郑州 450001)[1] (河南省信息安全重点实验室 郑州 450001)[2]
(数学工程与先进计算国家重点实验室 郑州 450001)[3]
(中国洛阳电子装备试验中心 河南 洛阳 471003)[4]

摘 要 基于角色的访问控制(Role-Based Access Control,RBAC)在管理和安全方面具有优势,经过 20 多年的发展后已被广泛应用于各个领域,如何将数据繁多的非 RBAC 系统迁移成 RBAC 系统已经成为一个意义重大的难题。角色是 RBAC 的基本特征,因此角色挖掘是 RBAC 系统实施的一个重要环节。基于形式概念分析生成用户权限概念格及用户属性概念格,将用户权限概念格翻转后映射为初始候选角色状态,通过约简操作和精简操作来挖掘角色,然后对用户权限概念格及用户属性概念格进行相似性分析,通过定义最近似表达式为角色赋予语义,使得生成的角色具有以下两点优势:1)结构层次,有效地减轻了管理员授权的负担,提高了授权管理的效率;2)语义意义,能够与现实生活中的概念相关联,增强了角色的可解释性。最后,通过实验验证了该算法的正确性和有效性。

关键词 角色挖掘,形式概念分析,概念格,属性,语义
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.12.018

Semantic Roles Mining Algorithms Based on Formal Concept Analysis

ZHOU Chao[1,2,3,4] REN Zhi-yu[1,2,3] WU Wen-chao[1,2,3]
(Information Engineering University,Zhengzhou 450001,China)[1]
(Henan Province Key Laboratory of Information Security,Zhengzhou 450001,China)[2]
(State Key Laboratory of Mathematical Engineering & Advanced Computing,Zhengzhou 450001,China)[3]
(Electronic Equipment Test Cneter,Luoyang,Henan 471003,China)[4]

Abstract Role-based access control (RBAC) with the advantages of management and security has been widely used in various fields after more than 20 years of development. How to migrate a non-RBAC system with a variety of data into an RBAC system has become a significant problem. Role is a basic feature of RBAC, therefore, role mining is an important part of the implementation of RBAC system. In this paper, the user-permission concept lattice and user-attribute concept lattice were generated based on formal concept analysis. After the user-permission concept lattice was reversed, it was mapped to initial candidate role state, and the final role state was mined by reduction and pruning operations. And then, the most approximate expressions were defined to give semantic meanings to roles by analyzing the similarity between user-permission concept lattice and user-attribute concept lattice. The generated roles have two advantages, one is structural hierarchy, which effectively reduces the authorization burden of administrator, and the other one is semantic meanings, which can be associated with the concepts in real life, enhancing the interpretability of role. Finally, the experimental results verify the correctness and effectiveness of the proposed algorithm.

Keywords Role mining, Formal concept analysis, Concept lattice, Attribute, Semantic meanings

1 引言

基于角色的访问控制[1]是当前应用得最为广泛的访问控制模型之一,它将角色作为用户-权限分配的桥梁,极大程度地简化了用户的授权操作,实现了灵活、方便、安全的授权管理。角色是 RBAC 模型的基本特征,也是其实现的基础。RBAC 在系统的实际应用中,首先需要构建一个完整、正确、有效的角色集。1996 年,Coyne 将准确地建立反映机构内的活动、业务和职责的角色集的过程称为角色工程(Role Engineering)[2]。目前,有两大类角色工程方法:1)自顶向下的方法,它需要对机构的业务逻辑以及大量用例进行分析,识别完成具体任务的必要权限,进而定义角色,为角色分配相应的权

到稿日期:2017-11-22 返修日期:2018-03-24 本文受国家自然科学基金(61702550,61502531),国家"八六三"高技术研究发展计划项目基金(SQ2015AA011705)资助。

周 超(1993—),男,硕士生,主要研究方向为信息安全,E-mail:zacharyvic@163.com;任志宇(1974—),女,博士,副教授,主要研究方向为信息安全,E-mail:zhiyu.ren@163.com(通信作者);毋文超(1995—),男,硕士生,主要研究方向为信息安全。

限;2)自底向上的方法,也称为角色挖掘,它通过分析系统中已有的用户-权限指派关系,利用数据挖掘技术或其他方法自动或半自动地生成角色状态。相比于自顶向下的方法,角色挖掘加速了 RBAC 系统的构建,其所独有的自动化、半自动化特点使其在大型信息系统中实施 RBAC 模型具有更大的价值。

Mitra 等在文献[3]中对角色挖掘方法进行了分类总结。现有的角色挖掘算法挖掘出的角色集大部分都是扁平化的结构,在大型系统中不具有实际意义,本文主要关注能够挖掘角色层次的角色挖掘算法。Schlegelmilch 等[4]采用聚类分析的方法从已有的用户权限信息中挖掘角色,并开发相应的角色挖掘工具——ORCA,该方法是最早提出的角色层次挖掘算法,但其得到的最终角色集层次过多,包含了大量的冗余角色,实用性不强。Zhang 等[5]将角色挖掘问题转换为矩阵分解问题,得到了角色层次图,提出了图优化算法,以角色数量、用户角色指派和权限角色指派三者之和最小为目标进行角色挖掘。Guo 等[6]定义了角色层次挖掘问题 RHMP,从最小化角色层次的角度出发,以有向图表示角色层次,角色与角色之间层次性的关系用一条路径表示,优化的角色层次关系既可保持图中的传递关系又可使边数最少。上述方法主要是将权限分组或是将用户分组,但对于角色挖掘来说,需要同时对用户和权限进行分组,因此需要寻找更为有效的方法进行角色挖掘。

形式概念分析是一种强有力的数据分析和规则提取工具,在机器学习、数据挖掘、信息检索、软件工程等领域得到了广泛的应用[7]。形式概念分析不仅能够实现同时对用户和权限进行分组,而且形式概念分析中定义的形式概念及概念格与 RBAC 模型中定义的角色以及角色层次具有十分完美的对应关系,因此使用形式概念分析的方法来挖掘角色具有天然的优势。Molly 等[8]、Sobieski 等[9]、Kumar 等[10-11]、张磊等[12]分别提到使用形式概念分析来挖掘具有层次结构的角色。然而,仅仅有角色层次还不足以将角色与现实生活中的概念结合起来,管理员依然难以理解挖掘出来的角色的意义。

本文提出的算法是在文献[8]的基础上,基于形式概念分析生成用户权限概念格,将用户权限概念格翻转后映射为初始候选角色状态,通过约简、精简操作最终得到带层次结构的角色集以及用户角色与角色权限的分配关系,之后基于形式概念分析生成用户属性概念集,为角色定义最近似表达式,最大程度地为每个角色赋予现实生活中的语义信息,最终得到具有语义意义及层次结构的角色状态,增强了角色的可解释性,减轻了管理员的授权负担。

2 相关理论基础

本节主要介绍角色挖掘及形式概念分析的相关定义。现有文献对角色挖掘问题进行了许多定义,综合各方观点,本文将角色挖掘问题定义如下。

定义 1(角色挖掘) 给定访问控制配置 $\rho=\langle U,P,UP\rangle$,其中,$U$ 是所有用户的集合,P 是所有权限的集合,$UP\subseteq U\times P$ 是用户-权限关系。找出一个与 ρ 一致的 RBAC 状态 $\langle R,UA,PA,RH,UPDA\rangle$,其中,$R$ 是角色集,$UA\subseteq U\times R$ 是用户-角色分配关系,$PA\subseteq P\times R$ 是权限-角色分配关系,$RH\subseteq R\times R$ 是角色层次即角色之间的偏序关系,$DUPA\subseteq U\times P$ 是直接的用户-权限分配关系。在 RBAC 状态中,若 U 中的每一个用户拥有的权限集与 UP 相同,则称 RBAC 状态与 ρ 一致。

1999 年,Ganter 出版的学术著作《Formal Concept Analysis:Mathematical Foundations》对形式概念分析理论的早期成果做了总结[13]。本文在此基础上对形式概念分析及其相关内容做出了以下定义。

定义 2(形式概念分析) 给定一组对象集及每个对象拥有的属性,找出输入数据中所有满足以下要求的对象聚类和属性聚类的集合:1)对象聚类是共享一组属性子集的所有对象的集合;2)属性聚类是一组自然对象聚类共有的所有属性的集合。

定义 3(形式背景) 形式背景 K 是一个三元组 (O,A,I),是形式概念分析的输入。其中,O 是对象集,A 是属性集,$I\subseteq O\times A$ 是 O 和 A 之间的二元关系,对于一个对象 $o\in O$,属性 $a\in A$,那么 oIa 就表示对象 o 具有属性 a,$\forall o\in O,a\in A$,有 $oIa\rightarrow(o,a)\in I$。

设 $X\subseteq O$ 和 $Y\subseteq A$,定义如下两个映射:

1)$f(X)=\{a\in A\mid(\forall o\in X)oIA\}$;

2)$g(Y)=\{o\in O\mid(\forall a\in Y)oIA\}$。

由定义可知,f 映射与 g 映射具有以下性质:

$X\subseteq g(f(X)),Y\subseteq f(g(Y))$

定义 4(形式概念) 形式概念 C 是一个二元组 (X,Y),其中 $X\subseteq O,Y\subseteq A$ 满足以下两点要求:1)$Y=f(X)$;2)$X=g(Y)$。X 为概念的外延,记为 $Ex(C)$,Y 为概念的内涵,记为 $In(C)$。

若 $C_i=(X_i,Y_i),C_j=(X_j,Y_j)$ 为同一形式背景下不同的形式概念,则其关于映射 f 和映射 g 具有以下性质:

1)$X_i=g(f(X_i)),Y_i=f(g(Y_i))$;

2)$X_i\subseteq X_j\Leftrightarrow Y_j\subseteq Y_i$。

定义 5(形式概念的关系) 对于同一形式背景中两个不同的形式概念 (X_1,Y_1) 和 (X_2,Y_2),定义关系 $<$,若 $(X_1,Y_1)<(X_2,Y_2)\Leftrightarrow X_1\subset X_2\Leftrightarrow Y_2\subset Y_1$,则称概念 (X_1,Y_1) 是 (X_2,Y_2) 的亚概念,概念 (X_2,Y_2) 是 (X_1,Y_1) 的超概念。若 $\neg\exists(X_3,Y_3),(X_1,Y_1)<(X_3,Y_3)<(X_2,Y_2)$,则称概念 (X_1,Y_1) 是 (X_2,Y_2) 的子概念,概念 (X_2,Y_2) 是 (X_1,Y_1) 的父概念,并记为 $(X_1,Y_1)\prec(X_2,Y_2)$。

定义 6(概念格) 概念格是形式背景 K 上所有形式概念 C 及其关系 $<$ 构成的偏序集,记为 $L(C,<)$。设 $(X_j,Y_j)(j\in J)$ 是概念格中的一个非空有限子集,其上确界记为 $\bigvee_{j\in J}(X_j,Y_j)$,其下确界记为 $\bigwedge_{j\in J}(X_j,Y_j)$。

$$\bigvee_{j\in J}(X_j,Y_j)=\left(g\left(\bigcap_{j\in J}Y_j\right),\bigcap_{j\in J}Y_j\right),\bigcup_{j\in J}X_j\subseteq g\left(\bigcap_{j\in J}Y_j\right)$$

$$\bigwedge_{j\in J}(X_j,Y_j)=\left(\bigcap_{j\in J}X_j,f\left(\bigcap_{j\in J}X_j\right)\right),\bigcup_{j\in J}Y_j\subseteq f\left(\bigcap_{j\in J}X_j\right)$$

3 语义角色挖掘算法

3.1 基于形式概念分析的角色挖掘

若 $\bigcup_{j\in J}X_j=g\left(\bigcap_{j\in J}Y_j\right)$,则概念 $\bigvee_{j\in J}(X_j,Y_j)$ 可由概念集 $(X_j,Y_j)(j\in J)$ 表示。因此,直接由概念格表示角色状态会出现冗余角色以及冗余的分配关系。本文方法将翻转的用户权限概念格映射为初始候选角色状态(角色继承关系的定义与本文

概念关系的定义是相反的),其中候选角色与概念对应,用户角色分配关系与概念的外延对应,权限角色分配关系与概念的内涵对应,角色层次关系与概念关系对应,在此基础上进行角色挖掘。

首先,消除初始候选角色状态中冗余的分配关系——约简操作。

将概念格中概念 C 去除冗余的对象与属性得到的结果称作约简概念 RC。假设形式概念 $C=(X,Y)$ 有子概念集 $(X_j,Y_j)(j\in J)$ 和父概念集 $(X_k,Y_k)(k\in K)$,约简概念 RC 的计算公式如下:

$$RC=(X-\bigcup_{j\in J}X_j,Y-\bigcup_{k\in K}Y_k)$$

约简概念的外延去除了冗余的用户角色分配关系,其内涵去除了冗余的权限角色分配关系。消除冗余的分配关系后最多存在 4 类候选角色:

1) $\bigcup_{j\in J}X_j=X$, $\bigcup_{k\in K}Y_k=Y$,既没有用户也没有权限;

2) $\bigcup_{j\in J}X_j\subset X$, $\bigcup_{k\in K}Y_k=Y$,有用户没有权限;

3) $\bigcup_{j\in J}X_j=X$, $\bigcup_{k\in K}Y_k\subset Y$,没有用户有权限;

4) $\bigcup_{j\in J}X_j\subset X$, $\bigcup_{k\in K}Y_k\subset Y$,既有用户也有权限。

接下来,消除约简候选角色状态中冗余的候选角色——精简操作。

第 4 类候选角色是必须保留的,但前 3 类候选角色并不是都必须消除,需要考虑消除候选角色之后是否使得 RBAC 状态更优。因此,使用文献[8]中提到的带权结构复杂度作为消除尺度来判断前 3 类候选角色是否需要消除。

定义 7(带权结构复杂度)　给定 $W=\langle w_r,w_u,w_p,w_h,w_d\rangle$,$w_r,w_u,w_p,w_h,w_d\geqslant 0$,一个 RBAC 状态 γ 的带权结构复杂度定义如下:

$$wsc(\gamma,W)=w_r*|R|+w_u*|UA|+w_p*|PA|+w_h*|RH|+w_d*|DUPA|$$

其中,$|X|$ 表示数量,特别注意 $|RH|$ 指的是概念格中由 p 定义的关系数量而不是由 $<$ 定义的关系数量,而且,对于角色 r,我们用 $Sen(r)$ 表示 r 的直接上级角色集,用 $Jun(r)$ 表示 r 的直接下级角色集,用 $Ajs(r)$ 表示删除 r 后需要增加的边,显然,$|Ajs(r)|\leqslant|Sen(r)|*|Jun(r)|$。同时,我们可以设置 $w_*=\infty$ 表示不使用某种条件。带权结构复杂度越小说明 RBAC 状态越好。

这里不考虑用户权限直接分配,我们设置 $w_d=\infty$,使用带权结构复杂度作为消除尺度,为前 3 类候选角色分别制定以下 3 条规则。

规则 1　对于既没有用户也没有权限的候选角色 r,候选角色仅仅作为其他候选角色的一个连接点,移除该候选角色能够减少创建该候选角色以及与该候选角色相关的边带来的花费,但是为了保持继承关系的正确性,需要增加一些边。当满足以下条件时,移除候选角色 r:

$$\begin{cases}|UA|=0\\|PA|=0\\w_h*(|Sen(r)|+|Jun(r)|)+w_r\geqslant w_h*|Ajs(r)|\end{cases}$$

规则 2　对于有用户但没有权限的候选角色 r,如果候选角色 r 被移除,则需要将 r 中的所有用户分配给 r 所有的直接下级候选角色 $Jun(r)$,同时为了保持继承关系的正确性,需要增加一些边。当满足以下条件时,移除候选角色 r:

$$\begin{cases}|UA|=m\\|PA|=0\\w_h*(|Sen(r)|+|Jun(r)|)+w_u*m+w_r\geqslant\\\quad w_h*|Ajs(r)|+w_u*m*|Jun(r)|\end{cases}$$

规则 3　对于有权限但没有用户的候选角色 r,如果候选角色 r 被移除,则需要将 r 中的所有权限分配给 r 所有的直接上级候选角色 $Sen(r)$,同时为了保持继承关系的正确性,需要增加一些边。当满足以下条件时,移除候选角色 r:

$$\begin{cases}|UA|=0\\|PA|=n\\w_h*(|Sen(r)|+|Jun(r)|)+w_p*n+w_r\geqslant\\\quad w_h*|Ajs(r)|+w_p*n*|Sen(r)|\end{cases}$$

执行约简操作消除初始候选角色状态中所有冗余的分配关系,执行精简操作消除约简候选角色状态中所有满足消除规则的候选角色,从而得到最终的角色状态。

注意,该算法是一种贪心算法,消除规则 2 和规则 3 中的候选角色会修改其他候选角色,因此候选角色的消除顺序会影响最终的结果。由于概念格自身的特点,其映射的候选角色状态中满足规则 2 的候选角色往往集中于上层且向下精简,而满足规则 3 的候选角色往往集中于下层且向上精简,因此,采用自上而下的顺序消除满足规则 2 的候选角色,采用自下而上的顺序消除满足规则 3 的候选角色,使得最终得到的角色状态的带权结构复杂度相较而言会更小,即角色状态更优。

3.2　基于形式概念分析的语义赋予

语义可以简单地看作是数据所对应的现实世界中的事物所代表的概念的含义,以及这些含义之间的关系,是数据在某个领域上的解释和逻辑表示。对于角色来说,拥有语义意义的角色应该能够用用户或是客体属性来表示,本文仅以用户属性为例对角色进行语义赋予。

由 3.1 节的描述可知,给定一组访问控制配置 $\rho=\langle U,P,UP\rangle$,可以构造一个用户权限概念格,记为 $L_\rho(C_\rho,<)$,其中,U 是所有用户的集合,P 是所有权限的集合,$UP\subseteq U\times P$ 是用户-权限关系。同理,给定一组属性配置 $\sigma=\langle U,A,UAT\rangle$,也能构造一个用户属性概念格,记为 $L_\sigma(C_\sigma,<)$,其中,U 是所有用户的集合,A 是所有属性的集合,$UAT\subseteq U\times A$ 是用户-属性关系。

用户属性包括工作单位、岗位职责等。为了方便管理,几乎所有的公司都维护着员工的属性信息,甚至有些公司将员工的一部分属性信息公布到访问网站上。属性按照其属性值的类型可分为单值属性、多值属性、区间属性、集合属性等。文献[14]讨论了异构数据集上的偏序形成,提出了面向异构数据分析的广义概念格模型。本文为了简化算法描述,将属性均视为单值属性(多值属性可看作 n_1 个单值属性,n_1 为属性值的个数;区间属性可看作 n_2 个单值属性,n_2 为属性区间的个数;集合属性可看作 2^{n_3} 个单值属性,n_3 为集合中元素的总个数)。

定义 8(属性表达式)　一个属性表达式能够表述成以下两种形式:

1) $e(A)=\Delta,A=\varnothing$:任何用户都满足属性表达式;

2)$e(A)=a_1 \wedge a_2 \wedge \cdots \wedge a_k, A=\{a_1,a_2,\cdots,a_k\}$:一个用户 u 满足配置 σ 下的属性表达式 $e(A)$,当且仅当 $\forall i \in [1,k]$,有 $(u,a_i) \in UAT$。

$U_\sigma[e(A)]$表示所有满足 $e(A)$的用户集合。在概念格 $L_\sigma(C_\sigma,<)$中,若 A 是形式概念 C_σ 的内涵,则 $U_\sigma[e(A)]$是形式概念 C_σ 的外延。

一般而言,用户权限的分配关系是根据用户的属性确定的,因此用户权限概念格与用户属性概念格具有一定程度上的同构性,也就是说,用户属性表达式能够赋予角色语义意义。

定义 9(一致表达式) 给定一组用户属性配置 $\sigma=\langle U,A,UAT\rangle$和与访问控制配置 $\rho=\langle U,P,UP\rangle$一致的 RBAC 状态 γ,当且仅当 $U_\gamma(r)=U_\sigma[e(A)]$时,称属性表达式 $e(A)$是角色 r 的一致表达式,$U_\gamma(r)$表示被分配角色 r 及其父角色的用户集。

角色的一致表达式代表了这个角色真实世界的概念,然而,由于其他因素的影响,不是 RBAC 状态中的每一个角色在用户属性概念格上都有一致表达式。如果一个角色没有一致表达式,那么我们认为在现实世界中不存在一个使用用户属性表达式表示的概念在给定的配置中与之完全关联。在这种情况下,可以考虑这个角色是否能够表达现实世界中概念的一部分。

定义 10(近似表达式) 给定属性配置 $\sigma=\langle U,A,UAT\rangle$和与访问控制配置 $\rho=\langle U,P,UP\rangle$一致的 RBAC 状态 γ,当且仅当 $U_\gamma(r)\subseteq U_\sigma[e(A)]$时,称属性表达式 $e(A)$是角色 r 的近似表达式。

给定一组 RBAC 状态和用户属性关系,角色 r 的所有用户都满足属性表达式 $e(A)$,但满足属性表达式 $e(A)$的所有用户可能有些不是角色 r 所拥有的用户。

定理 1 每个角色至少有一个近似表达式 Δ。

证明:因为 $U_\gamma(r)\subseteq U=U_\sigma[\Delta]$必然成立,所以对于任意角色 r,Δ 是 r 的近似表达式。

定理 2 如果角色 r 拥有一致表达式,那么这个一致表达式一定是其近似表达式。

证明:$e(A)$是角色 r 的一致表达式,$U_\gamma(r)=U_\sigma[e(A)]\subseteq U_\sigma[e(A)]$,因此 $e(A)$是 r 的近似表达式。

角色 r 与其近似表达式描述的现实世界中的概念是相关的。例如,角色 r 拥有近似表达式 $e(A)=$计算机系 $\wedge$ 教师,这就意味着被赋予角色 r 的用户是计算机系全体教师的一部分。

定义 11(最近似表达式) 当且仅当属性表达式 $e(A)$是角色 r 的近似表达式,并且不存在一个 $A'\supset A$,使得 $e(A')$也是角色 r 的近似表达式时,称属性表达式 $e(A)$是角色 r 的最近似表达式。

定理 3 对于给定的 RBAC 状态和用户属性配置,角色 r 有且仅有一个最近似表达式。

证明:角色 r 至少有一个近似表达式 Δ,因此 r 一定存在最近似表达式。设角色 r 有两个不同的最近似表达式 $e(A_1)$和 $e(A_2)$,$a\in A_1$ 且 $a\notin A_2$,若 $A_3=\{a\}\cup A_2$,通过定义 10 可知,$e(A_3)$也是角色 r 的近似表达式,由于 $A_3\supset A_2$,因此$e(A_3)$不是最近似表达式,这与假设矛盾,假设不成立。因此,角色 r 有且仅有一个最近似表达式。

定理 4 如果角色 r 拥有一致表达式,那么这个一致表达式是其唯一的最近似表达式。

证明:若属性表达式 $e(A)$是角色 r 的一致表达式,则由定义 9 有 $U_\gamma(r)=U_\sigma[e(A)]$。假设 $A'\supset A$,由概念格的性质可知,$U_\sigma[e(A')]\subset U_\sigma[e(A)]=U_\gamma(r)$。显然,$e(A')$不会是角色 r 的近似表达式,那么 $e(A)$是角色 r 的最近似表达式,再由定理 3 可知,$e(A)$是 r 唯一的最近似表达式。

推论:对于角色 r 来说,一定存在用户属性概念子集 $Subset(C)=\{(U_i,A_i)\mid U_\gamma(r)\subseteq U_\sigma[e(A_i)]=U_i\}$,则满足$|U_i|$最小的概念所对应的内涵 A_i 的元素的合取即为角色 r 的最近似表达式。也就是说,将用户属性概念格中形式概念按外延数量由小到大的顺序排列,第一个外延包含 $U_\gamma(r)$的概念所对应的内涵元素的合取即为 r 的最近似表达式。

由于最近似表达式具有存在性和唯一性,根据推论可知,在一个有序的用户属性概念集上可以为每个角色搜索其最近似表达式。最近似表达式能够为角色赋予现实生活中的语义信息,同时能够实现用户-角色动态分配,提高授权效率。

3.3 语义角色挖掘算法描述

语义角色挖掘算法分为两部分:基于形式概念分析的角色挖掘算法和基于形式概念分析的语义赋予算法。其主要思想是:1)针对用户-权限分配关系,使用形式概念分析构造用户权限概念格,将翻转后的用户权限概念格映射为初始候选角色状态,根据约简概念计算公式约简初始候选角色状态,使用候选角色消除规则对约简候选角色状态进行精简操作,得到的即为所需的角色状态;2)针对用户-属性分配关系,使用形式概念分析构造有序的用户属性概念集,将所求角色恢复成其对应的概念,在有序的用户属性概念集上搜索其最近似表达式,最终实现为每个角色赋予语义意义。算法 1 是基于形式概念分析的角色挖掘算法,算法 2 是基于形式概念分析的语义赋予算法。

算法 1 角色挖掘算法 MineRole

```
Function MineRole(U,P,UP)
输入:用户 U,权限 P,用户权限分配关系 UP
输出:RBAC 状态〈R,UA,PA,RH〉
BEGIN
1. L=BuildLattice(U,P,UP);//构造概念格
2.〈R,UA,PA,RH〉=Reverse(R,UA,PA,RH);//翻转概念格
3.〈R,UA,PA,RH〉=L;//将翻转后的概念格映射为初始候选角色状态
4.〈R,UA,PA,RH〉=Reduce(R,UA,PA,RH);//使用约简概念计算公式约简初始候选角色状态
5.〈R,UA,PA,RH〉=Prune(R,UA,PA,RH);//根据制定的 3 条消除规则执行精简操作,从而得到最终结果
END
```

算法 2 语义赋予算法 EndowMeaning

```
Function EndowMeaning(r,U,A,UAT)
输入:角色 R,用户 U,属性 A,用户属性分配关系 UAT
输出:角色 R 的最近似属性表达式 e
BEGIN
1. L=BuildLattice(U,A,UAT);//构造概念格
```

2. C_{UAT}=L. C;//得到用户属性概念集

3. C_{UAT}=Sort(C_{UAT});//将用户属性概念集排序

4. For r in R

5.　A_r=Search(r,C_{UAT});//搜索匹配的概念外延

5.　e_r=Conjunction(A_r);//合取式,即为最近似表达式

6. End For

END

假设用户数为 m,权限数为 n,候选角色数为 r,候选角色层次边数为 e,则构造概念格算法的时间复杂度为 $O((m+n)^2+r^2)$,约简初始候选角色算法的时间复杂度为 $O(r)$,精简约简候选角色状态算法的时间复杂度为 $O(re)$,最终算法 1 的时间复杂度为 $O((m+n)^2+r(r+e))$。文献[4]提出的 ORCA 算法的时间复杂度为 $O(mn^2)$,文献[5]提出的 GO 算法的时间复杂度为 $O(m^2n+(k+m)kn)$,其中 k 为迭代步数。虽然本文算法的时间复杂度略大于 ORCA,与 GO 相当,但依然在可接受范围内且得到的角色状态更优。假设用户属性概念集中的概念数为 c,算法 2 的时间复杂度为 $O(rc)$。

4　实例分析

4.1　实例描述

为验证算法的有效性,本文选用文献[12]中提到的电子病历系统作为背景实例,利用基于形式概念分析的语义角色挖掘算法进行角色挖掘和语义赋予,从而产生具有语义意义和层次结构的角色状态。

在该实例中,用户岗位分为普通岗位和管理岗位两大类。普通岗位包括挂号员(1)、外科医生(2)、内科医生(3)、妇科医生(4)、护士(5)和药剂师(6)。管理岗位包括外科主任(7)、内科主任(8)、妇科主任(9)、医务科长(10)、总护士长(11)、药房主任(12)以及院长(13)。根据各个场景信息的读写以及各个职能的授权操作,枚举系统中所使用到的权限如下:读病人基本信息(a)、写病人基本信息(b)、读住院信息(c)、写住院信息(d)、读历史记录(e)、读诊断信息(f)、读药方(g)、读护士报告(h)、写内科历史记录(i)、写外科历史记录(j)、写妇科历史记录(k)、写内科诊断信息(l)、写外科诊断信息(m)、写妇科诊断信息(n)、写内科药方(o)、写外科药方(p)、写妇科药方(q)、写护士报告(r)、内科医生授权(s)、外科医生授权(t)、妇科医生授权(u)、药剂师授权(v)、护士授权(w)。科室与职能信息枚举系统中所使用到的属性如下:内科(A)、外科(B)、妇科(C)、配药(D)、挂号(E)、诊断(F)、护理(G)、主任(H)。最终拥有 13 类用户、23 类权限以及 8 类属性。每类用户与权限的对应关系如表 1 所列,每类用户拥有的属性如表 2 所列。

表 1　电子病历系统的用户权限关系表

Table 1　User-permission relationship in electronic medical record system

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0
3	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
4	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0
5	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
6	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0
8	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0
9	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0
11	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
12	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 2　电子病历系统的用户属性关系表

Table 2　User-attribute relationship in electronic medical record system

	A	B	C	D	E	F	G	H
1	0	0	0	0	1	0	0	0
2	0	1	0	1	0	1	0	0
3	1	0	0	1	0	1	0	0
4	0	0	1	1	0	1	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	1	0	0	0	0
7	0	1	0	1	0	1	0	1
8	1	0	0	1	0	1	0	1
9	0	0	1	1	0	1	0	1
10	1	1	1	1	1	1	0	1
11	0	0	0	0	0	0	1	1
12	0	0	0	1	0	0	0	1
13	1	1	1	1	1	1	1	1

4.2　算法实现

过程 1　角色挖掘

步骤 1　根据表 1 提供的用户权限关系使用 Godin 算法[15]构造用户权限概念格,将其翻转后映射为候选角色状态,使用绘图工具 GraphViz 画出其 Hasse 图,如图 1 所示。

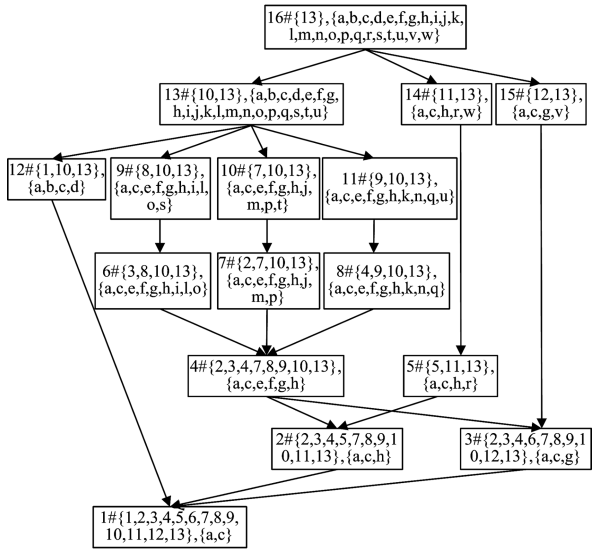

图 1　初始候选角色状态

Fig. 1　Initial candidate role state

步骤 2 查询每一个概念的子概念集和父概念集，根据约简概念计算公式计算其对应的约简概念，建立概念与约简概念的对应关系，从而得到约简的侯选角色状态，如图 2 所示。例：概念 4＃{2，3，4，7，8，9，10，13}，{a，c，e，f，g，h}的子概念集为{6＃，7＃，8＃}，其外延的并集为{2，3，4，7，8，9，10，13}，父概念集为{2＃，3＃}，其内涵的并集为{a，c，g，h}，计算得到概念 4＃对应的约简概念为{}，{e，f}。

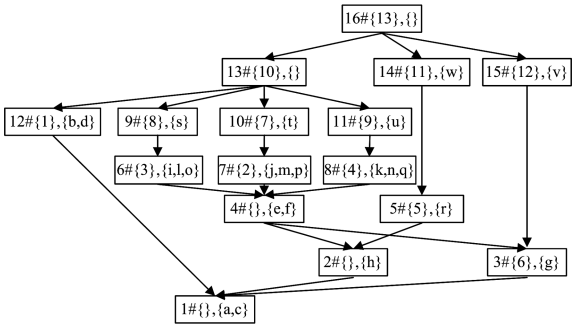

图 2 约简候选角色状态

Fig. 2 Reduced candidate role state

步骤 3 设置 $W=\langle 1,1,1,1,\infty\rangle$，根据消除规则，自上而下地消除满足消除规则 2 的侯选角色{16＃}，自下而上地消除满足消除规则 3 的侯选角色{2＃}，得到用户权限精简格，即为最终的角色状态，如图 3 所示。

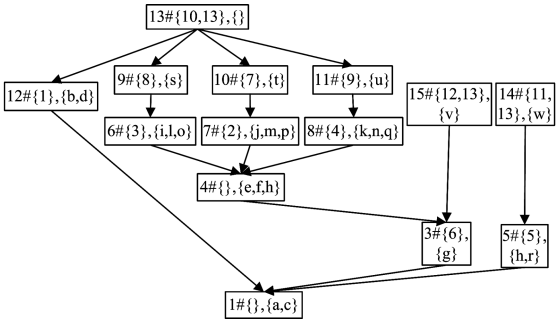

图 3 最终角色状态

Fig. 3 Final role state

过程 2 语义赋予

步骤 4 根据表 2 所提供的用户属性关系，使用 Ganter 提出的 NextClousure 算法[16]生成用户属性概念集，并依据用户数目和权限数目对生成的概念集进行排序，从而得到一个有序的用户属性概念集，如图 4 所示。

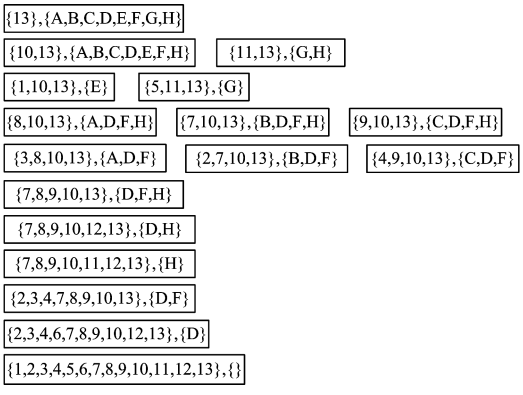

图 4 用户属性概念集

Fig. 4 User-attribute concept sets

步骤 5 使用步骤 2 中建立的约简概念与概念的对应关系，将图 3 中的每个角色恢复成图 1 中的概念。例：角色 4＃{}，{e，f，h}对应的概念为{2，3，4，7，8，9，10，13}，{a，c，e，f，g，h}。

步骤 6 在图 5 所示的概念集中，对于每个角色对应的概念的外延，自上而下按序搜索其最近似表达式，为每个角色赋予语义意义。例：角色 4＃{}，{e，f，h}对应的最近似表达式为 $D\wedge F$，这表示同时具有配药属性和诊断属性的人员能够分配角色 4＃，或者说，角色 4＃代表了同时具有配药属性和诊断属性的人员。

4.3 算法比较

将本文算法挖掘到的角色结构（见图 3）、文献[12]中挖掘到的角色结构与原始角色结构（见图 5）进行对比可以看出，仅仅挖掘最小角色集不足以表示所有的层次结构，虽然最小角色集看起来较简洁，但是扩展起来却比较复杂，例如，需要再添加一个儿科科室时，本文算法挖掘的角色结构扩展起来更加方便，需要添加的分配关系更少。同时，本文算法使用用户属性最近似表达式为角色赋予语义意义，相较于文献[12]中根据角色的权限和用户在系统中的功能和实际岗位为角色赋予语义意义更加精确。

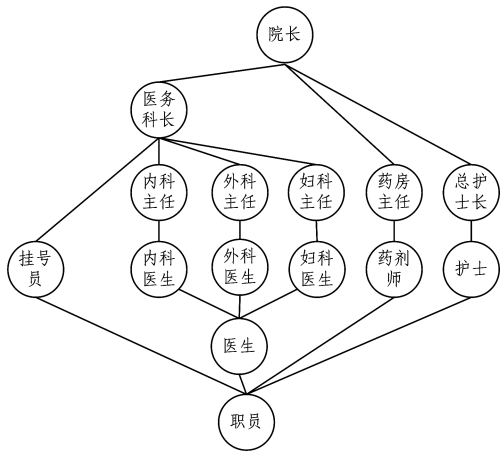

图 5 电子病历系统的原始岗位关系

Fig. 5 Original post relationship in electronic medical record system

结束语 本文研究了基于形式概念分析的语义角色挖掘算法，该算法不仅能得到具有层次结构的角色集，而且能得到用户-角色以及角色-权限的分配关系，同时引入用户属性最近似表达式赋予角色现实生活中的概念作为语义，实现了授权的半自动化，增强了角色的可解释性，能够有效地指导管理员进行授权管理，提高授权效率，减轻授权负担。虽然基于形式概念分析进行语义角色挖掘获得了十分优秀的角色结构，但是概念格构造算法本身的时间复杂度较高，从数据规模非常大的访问控制背景中挖掘角色结构十分费时。然而，随着计算机的发展和大数据时代的到来，概念格的并行构造算法也成为研究重点，基于形式概念分析的语义角色挖掘算法也同样具有很好的应用前景。

引入客体属性、环境属性等约束，使用用户权限直接分配，使得挖掘的角色集中具有一致表达式的角色更多，实现授权管理与访问控制的自动化是本文进一步的研究方向。

（下转第 129 页）

Monitoring[C]// Global Communications Conference. 2017.

[15] YAN Y, ZHANG S, TANG J, et al. Understanding characteristics in multivariate traffic flow time series from complex network structure[J]. Physica A: Statistical Mechanics & Its Applications, 2017, 477.

[16] LAKHINA A, CROVELLA M, DIOT C. Mininganomalies using traffic feature distributions[C]// Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. ACM, 2005: 217-228.

[17] PENG T, LECKIE C, RAMAMOHANARAO K. Proactively detecting distributed denial of service attacks using source IP address monitoring[C]// International Conference on Research in Networking. Springer Berlin Heidelberg, 2004: 771-782.

[18] SUN Q D, ZHANG D Y, GAO P. Distributed Denial of Service Attack Detection Based on Time Series Analysis[J]. Chinese Journal of Computers, 2005, 28(5): 767-773. (in Chinese)
孙钦东，张德运，高鹏. 基于时间序列分析的分布式拒绝服务攻击检测[J]. 计算机学报，2005，28(5)：767-773.

[19] XU Z, ZHU S, FU B, et al. Motion coherence based abnormal behavior detection [C] // Control and Decision Conference. IEEE, 2017: 214-218.

[20] HAN D Z, BI K, XIE B L, et al. An Anomaly Detection on the Application-Layer -Based QoS in the Cloud Storage System[J]. Computer Science and Information Systems, 2016, 13(2): 659-676.

[21] YUAN Y, WANG D, WANG Q. Anomaly Detection in Traffic Scenes via Spatial-Aware Motion Reconstruction [J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(5): 1198-1209.

[22] CHANG R K C. Defending against flooding-based distributed denial-of-service attacks: a tutorial[J]. IEEE Communications Magazine, 2002, 40(10): 42-51.

[23] LEMON J. Resisting SYN Flood DoS Attacks with a SYN Cache[C]// Bsdcon Conference. 2002.

[24] WU J S, ZHANG W P, MA Y. The Data Analysis of KDD-CUP99 Data Set [J]. Computer Applications and Software, 2014 (11): 321-325. (in Chinese)
吴建胜，张文鹏，马垣. KDDCUP99 数据集的数据分析研究[J]. 计算机应用与软件，2014(11)：321-325.

[25] AHMED H, ISMAIL M A, HYDER M F, et al. Performance Comparison of Spark Clusters Configured Conventionally and a Cloud Service[J]. Procedia Computer Science, 2016, 82: 99-106.

[26] ZAHARIA M, DAS T, LI H, et al. Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters[C]// Usenix Conference on Hot Topics in Cloud Computing. USENIX Association, 2012.

(上接第 122 页)

参 考 文 献

[1] SANDHU R S, COYNE E J, FEINSTEINH L, et al. Role-based access control models[J]. Computer, 1996, 29(2): 38-47.

[2] COYNE E J. Role engineering[C]// Proceedings of the first ACM Workshop on Role-based access control. ACM, 1996.

[3] MITRA B, SURAL S, VAIDYA J, et al. A survey of role mining [J]. ACM Computing Surveys (CSUR), 2016, 48(4): 1-37.

[4] SCHLEGELMILCH J, STEFFENS U. Role mining with ORCA [C]// Proceedings of the tenth ACM symposium on Access control Models and Technologies. ACM, 2005: 168-176.

[5] ZHANG D N, RAMAMOHANARAO K, EBRINGER T. Role engineering using graph optimization[C]// Proceedings of the 12th ACM Symposium on Access Control Models and Technologies. 2007: 139-144.

[6] GUO Q, VAIDYA J, ATLURI V. The role hierarchy mining problem: discovery of optimal role hierarchies[C]// Computer Security Applications Conference. IEEE, 2008: 237-246.

[7] SARMAH A K, HAZARIKA S M, SINHA S K. Formal concept analysis: current trends and directions[J]. Artificial Intelligence Review, 2015, 44(1): 47-86.

[8] MOLLOY I, CHEN H, LI T, et al. Mining roles with multiple objectives[J]. ACM Transactions on Information and System Security (TISSEC), 2010, 13(4): 1-35.

[9] SOBIESKI S, ZIELINSKI B. Modelling role hierarchy structure using the Formal Concept Analysis[J]. Annales Umcs Informatica, 2010, 10(2): 143-159.

[10] KUMAR C. Designing role-based access control using formal concept analysis[J]. Security and Communication Networks, 2013, 6(3): 373-383.

[11] KUMAR C A, MOULISWARAN S C, LI J, et al. Role based access control design using triadic concept analysis[J]. Journal of Central South University, 2016, 23(12): 3183-3191.

[12] ZHANG L, ZHANG H L, HAN D J, et al. Theory and Algorithm for Roles Minimization Problem in RBAC Based on Concept Lattice[J]. Acta Electronica Sinica, 2014, 42(12): 2371-2378. (in Chinese)
张磊，张宏莉，韩道军，等. 基于概念格的 RBAC 模型中角色最小化问题的理论与算法[J]. 电子学报，2014，42(12)：2371-2378.

[13] GANTER B, WILLE R. Formal concept analysis: mathematical foundations[M]. New York: Springer Science & Business Media, 2012.

[14] ZHI H L. Extended Model of Formal Concept Analysis Oriented for Heterogeneous Data Analysis[J]. Acta Electronica Sinica, 2013, 41(12): 2451-2455. (in Chinese)
智慧来. 面向异构数据分析的形式概念分析扩展模型[J]. 电子学报，2013，41(12)：2451-2455.

[15] GODIN R, MINEAU G, MISSAOUI R, et al. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications[J]. Revued'Intelligence Artificielle, 1995, 9: 105-137.

[16] GANTER B. Two Basic Algorithms in Concept Analysis[C]// International Conference on Formal Concept Analysis. Springer-Verlag, 2010: 312-340.