

基于信号量机制的理发师模型在复杂语义下的研究

张宝哲 帖 军 蒋天发

(中南民族大学计算机科学学院 武汉 430074)

摘 要 理发师问题是计算机科学中多系统进程间通信和同步的经典问题之一,研究者们多在原始版本基础上进行扩展,形成较多版本,没有考虑问题语义变化产生的新问题。从生活原型和问题语义的角度出发,对理发师问题进行新的语义分析,提出面向资源的模式分析方法,给出了利用 Try-p 信号量机制和 PV 原语解决具有复杂语义进程间同步问题的算法和方案。

关键词 进程,同步,互斥,信号量,原语

中图法分类号 TP312 **文献标识码** A

Research of Barber Model Based on Semaphore with Complicated Semantics

ZHANG Bao-zhe TIE Jun JIANG Tian-fa

(Department of Computer Science, South Central University for Nationalities, Wuhan 430074, China)

Abstract The sleeping barber model problem in computer science is a classic inter-process communication and synchronization problem between multiple operating system processes, however, most researchers was used to introduce extra functions to extend the original model, ignoring new complicated problems arose from semantic changes and not giving innovative solutions. Based on life prototype and semantics of new model, this paper redefined the barber model with complicated semantics, and proposed analysis pattern of resource oriented and provided solutions and algorithms of inter-process synchronization problems within complicated semantics based on Try-p semaphore and p-v primitives.

Keywords Process, Synchronization, Mutex, Semaphore, Primitive

1 引言

进程同步是指在异步环境下的一组并发进程因直接制约而互相发送消息且进行互相合作、互相等待,使得各进程按一定的速度执行的一种行为;信号量是同步的进程间进行通信而发送的一种制约信号,它代表了同步进程对共享资源的一种合作使用方式。进程同步作为现代应用中一个重要的角色,可以对复杂分支和同步 workflow 模型进行分析和应用,推算系统行为^[1],并广泛应用于操作系统和底层系统构建中的经典问题^[2,3]。理发师问题是多进程间通信和同步互斥等并发问题的经典问题,来源于 Edsger W. Dijkstra 在 1965 年第一次提出的 PV 原语和信号量机制^[4]。当前,多数研究者以面向过程的模式^[5,6],只在原型基础上添加其它功能来进行变化和扩充。如果改变理发师问题的语义,从面向资源的角度和复杂信号量机制来分析和解决理发师等进程同步问题,解决思路和算法会有很大不同。本文从复杂语义角度剖析理发师模型,并引入 Try-p 的信号量机制对新的语义模型进行分析并提出优良的解决思路。

2 经典理发师问题的语义模型

解决进程间同步互斥问题主要有两种思路,即面向过程

和面向资源的方法。传统的进程并发主要从面向过程的角度进行建模分析^[7],符合现实世界观察和解决问题的思维,经典模型如图 1 所示。作为进程并发控制问题的经典案例,该问题已被学者熟知,并可变化为多种版本,例如理发师除了理发还可以进行收银活动;添加一个站区等待队列等。较全面版本的流程如图 2 所示,从面向过程的角度考察分析,主要思想在于进程推进中对资源的合理使用^[8,9]。

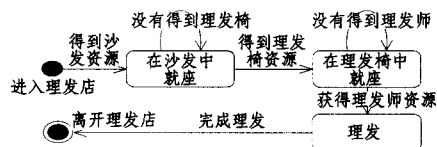


图1 理发师经典模型

但是,现实中的活动规则却可能颠倒理发师问题描述的流程。

语义的改变体现在两方面,活动流程发生变化或者活动内容发生变化。如果在图 2 的复杂语义下,纯粹从面向流程的角度分析问题来解决问题是有一定难度的。因此如果从面向资源的角度,讨论在不同复杂情况下问题的语义会较容易地解决问题,例如理发师不仅作为进程执行的驱动器,而且作为进程间互斥的资源使用。

本文受中南民族大学中央高校基本科研业务费专项资金项目(CZY11007),湖北省教育厅科研项目(B20110804)资助。

张宝哲(1987—),男,硕士生,主要研究方向为事务处理、分布式应用,E-mail:pautcher@gmail.com;帖 军(1976—),男,副教授,硕士生导师,主要研究方向为移动计算、分布式应用,E-mail:musa@scuec.edu.cn(通信作者);蒋天发(1954—),男,教授,硕士生导师,主要研究方向为高性能网络与信息安全。

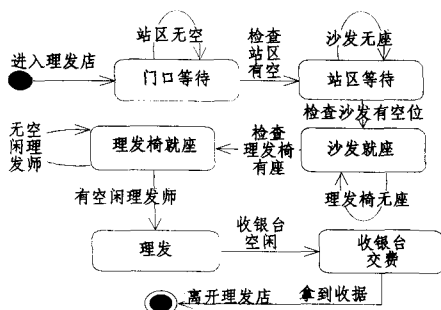


图2 理发师模型的一般流程

3 复杂语义下的理发师模型

3.1 活动流程语义改变的理发师模型

活动流程语义变化是指当顾客进入理发店后,等待理发的顺序不是按照经典理发师模型的等待顺序,而是以贪婪模式试图从获得理发师资源开始,如果得不到资源才进入下一级等待序列。其完全颠覆了经典理发师模型的语义流程。

问题提出:

理发店有 m 个理发师、 m 把理发椅、 n 个等候理发的沙发和 1 个收款机。它们的活动遵循如下条件:(1)如果没有顾客,则理发师在理发椅上睡觉;(2)当有一个顾客到达时,首先查看理发师在干什么,如果在睡觉则通知理发师理发,然后坐到理发椅上开始理发;如果理发师正在理发,则查看是否有空的沙发可坐,如果有,就坐下等待,如果没有,则在门口等待;(3)理发师为一位顾客理完发后,查看是否有人等待,如果有则唤醒一位为其理发,如果没有,则在理发椅上睡觉;(4)顾客不分优先级,理发完毕后,去交费并取收据,然后离开;(5)理发师可以收银,且任何时刻最多只能有一个理发师在收银。该语义问题的状态图如图 3 所示。

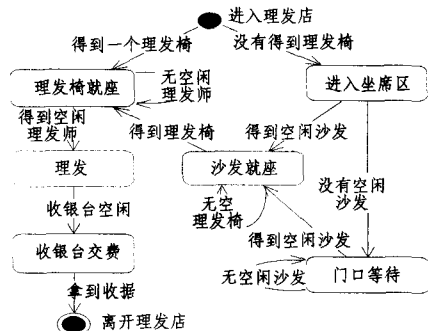


图3 活动流程语义改变的理发师模型

模型分析:

这是一类加强版的理发师问题。与常规理发师问题不同,语义变化主要体现在顾客的活动流程次序的改变。用常规信号量方案显然不能解决该问题,需要引入 Try-p 的信号量机制。Try-p 机制是一种改进的信号量方案,需要根据具体的环境语义灵活运用。

不同理发师可以并行收银或理发,因此需要创建新的收银进程。同一理发师只能进行理发或者收银,需要一个互斥信号量 $\text{mutex_count_barber}$,以控制同一理发师的理发与收银的互斥操作。理发师有多个,因此 $\text{mutex_count_barber}$ 的初值为理发师的数量,否则一个理发师进程进行理发或者收银时,其它理发师被挂起,直到该理发师释放 mutex_count_

barber 信号。

需要给理发师编号。在理发过程中,顾客需要保证自己理发完毕,才能够离开理发椅进行付款等活动。这个机制是通过 customer 进程获得给他理发的理发师编号来实现的。当该编号的理发师释放对应的 $s_finished[i]$ 信号时,表明该顾客理发完毕。具体解决方案如下。

初值设定:

```

count_barbers=m; //空闲的理发师的数量
mutex_count_barber=1; //控制 count_barbers 变化的互斥信号量
mutex_queue=1; //队列访问的互斥信号量
mutex_barber[] = {1, ..., 1}; //理发或收银的互斥信号量
s_fullSofa=0; //满沙发的同步信号量
s_emptySofa=n; //空沙发的同步信号量
s_chairs=0; //与顾客理发的同步信号量
s_finish[] = {0, ..., 0}; //理发完毕的同步信号量
s_leave=0; //顾客离开理发椅的同步信号量
s_payment=0; //等待顾客付账信号
s_receipt=0; //顾客收据信号量

```

Customer 进程:

```

int barber_id;
进入理发店;
P(mutex_count_barber)
if(count_barbers>0){
    count_barbers--;
    V(mutex_count_barber)
}
else{
    V(mutex_count_barber)
    P(s_emptySofa)
    V(s_fullSofa)
}
P(s_chairs)
V(s_emptySofa)
P(mutex_queue)
barber_id = dequeue();
V(mutex_queue)
理发;
P(s_finish[barber_id])
离开理发椅;
V(s_leave)
V(s_payment)
付款结账;
P(s_receipt)
离开理发店

```

Barber(id)进程:

```

While(true)
    P(mutex_queue);
    enqueue(id);
    V(mutex_queue);
    P(s_fullSofa)
    V(s_chairs)
    P(mutex_barber[id]) //理发
    V(mutex_barber[id])
    V(s_finish[id])
    P(s_leave)
    P(mutex_count_barber)
    count_barbers++;

```

```

V(mutex_count_barber)
do-while
Cash 进程:
While(true)
    P(mutex_queue)
    barber_id = dequeue();
    V(mutex_queue)
    P(s_payment)
    P(mutex_barber[id])
    收银;
    给顾客收据;
    V(mutex_barber[id])
    V(s_receipt)
do-while

```

顾客进入理发店,首先查看空闲理发师数量 count_barbers,每一个理发师对应一把理发椅。这时 count_barbers 就成为多个理发师访问的互斥资源,需要添加互斥信号量 mutex_count_barber。如果 count_barbers 大于零,则直接用 p(s_chairs)申请理发椅并进行理发。如果 count_barbers 小于或者等于零,则进入等待室进行 p(s_emptySofa)操作。如果 s_emptySofa 大于零,说明有可用空闲沙发资源,直接拿到资源后用 p(s_chairs)申请理发椅资源。否则,顾客进程会挂起在信号量 s_emptySofa 的等待队列中。这里对 count_barbers 和 s_emptySofa 采用了不同的方案,对 s_emptySofa 是常规 PV 原语,而对 count_barbers 使用 Try-p 方式的信号量方案。该版本的顾客流程符合现实世界的活动规则,解决该类问题更具现实意义。

3.2 活动内容语义改变的理发师问题

活动内容语义改变是指在理发师模型中加入新的活动内容,增加或改变理发师模型的缓冲区队列或缓冲区。例如增加一个站区等待队列,或者改变当理发师发现站区满时在门口等待。在理发师流程语义改变的基础上进行活动内容的语义改变。问题描述的状态图如图 4 所示。

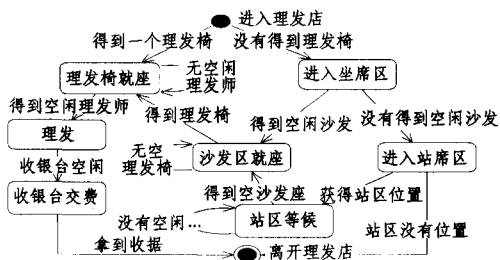


图 4 活动内容语义改变的理发师模型

问题提出:

理发店内有 1 个可站 w 位顾客的站席区、 n 个单人沙发、 m 个理发师及其专用理发工具、一个收银台。新来的顾客先看是否有空闲的理发师,如果有就直接去理发;如果没有,则坐在沙发上等待;没有空沙发时,可在站席区等待;站席区满时,就选择离开。理发师可从事理发、收银和睡觉 3 种活动。理发店的活动满足下列条件:(1)睡觉的理发师在自己专用的理发椅上,不会占用顾客的沙发,可为在沙发上等待时间最长的顾客理发;(2)理发时间长短由理发师决定;(3)在站席区等待时间最长的顾客可坐到空闲的理发上;(4)顾客理发完毕后,去交费并取收据,然后离开。(5)任何时刻最多只能有一个理发师在收银。

模型分析:

基于以上版本,该加强版继续增加语义复杂度。语义变化主要是问题内容的变化,主要出现两个难点。

首先,缓冲区队列由一个变为两个,即分沙发座区和站区,问题结构变得复杂。不同进程可以并发使用不同队列,但是同一进程只能串行按序申请不同缓冲区。比如,顾客到来时,先要申请理发椅,没有空闲理发椅再到沙发区,依次到站区,最后可能离开。按照申请资源的顺序流程,语义清晰且容易理解。

其次,顾客发现站区已满时选择离开,而不是在门口等待。语义变化,导致上节解决方案不再适用。如果按照图 4 的流程,问题将变得简单,常规信号量和 pv 原语即可实现。但是该问题语义按照图 4 所示的流程实现,需要多重 Try-p 机制保证,解决方案如下。

初值设定:

```

count_barbers=m; //空闲的理发师的数量
count_emptySofa=n; //空闲沙发的数量
count_stand=w; //站区可用位置数量
mutex_count_barber=1; //控制 count_barbers 的互斥信号
mutex_count_sofa=1; //控制沙发数量的互斥信号
mutex_barber[] = {1,...,1}; //理发或收银的互斥信号量
mutex_count_stand=1; //控制站区数量变化的互斥信号
mutex_queue=1; //队列访问的互斥信号量
s_fullSofa=0; //满沙发的同步信号量
s_emptySofa=n; //空沙发的同步信号两
s_chairs=0; //与顾客理发的同步信号量
s_finish[] = {0,...,0}; //理发完毕的同步信号量
s_leave=0; //顾客离开理发椅的同步信号两
s_payment=0; 等待顾客付账信号;
s_receipt=0; 顾客收据信息量
s_fullSofa=0; //满沙发的同步信号量
s_emptySofa2=n; //空沙发的同步信号量

```

Customer 进程:

```

int barber_id
P(mutex_count_barber)
if(count_barbers>0){
    Count_barbers--;
    V(mutex_count_barber)
}
else{
    V(mutex_count_barber)
    P(mutex_count_emptySofa)
    if(count_emptySofa>0){
        count_emptySofa--;
        V(s_fullSofa)
        V(mutex_count_emptySofa)
        P(s_chairs)
        P(mutex_emptySofa)
        count_emptySofa++;
        V(mutex_emptySofa)
    }
    else{
        V(mutex_count_emptySofa)
        P(mutex_count_stand)
        if(count_stand>0){

```

```

        count_stand--;
        V(mutex_count_stand)
        P(s_emptySofa2)
        V(s_fullSofa)
        P(mutex_count_stand)
        count_stand++;
        V(mutex_stand)
        P(s_chairs)
        V(s_emptySofa2)
    }
    else{
        V(mutex_count_stand)
        Exit();
    }
}
P(mutex_queue)
barber_id = dequeue();//得到理发师编号
V(mutex_queue)
    理发;
P(s_finish[barber_id])
    离开理发椅;
V(s_leave)
V(s_payment)
    付款结账拿收据;
P(s_receipt)
离开理发店
Barber(id)进程:
While(true)
    P(s_fullSofa)
    P(mutex_queue);
    enqueue(id);
    V(mutex_queue);
    V(s_chairs)
    P(mutex_barber[id])
        理发;
    V(mutex_barber[id])
    V(s_finish[id])
    P(s_leave)
    P(mutex_count_barber)
        count_barbers++;
    V(mutex_count_barber)
do-while
    收银进程不发生变化。

```

在此复杂语义的情况下,控制进程同步互斥的信号量有十几个;客户进程以一种弹簧推进的方式进行,弹簧顶端到尾端可以类比为理发椅到沙发区再到站区的阶段。当客户到来时,先看理发椅是否可用。如果可用,则直接在理发椅理发,否则进入沙发区,但是还要再申请理发椅资源。当到达站区时,要再先申请沙发座,再申请理发椅。客户到站区之后,不能再直接申请理发椅资源。由此看见,客户进程以一种弹簧式的推进方式向前进行。在不同的阶段,需要考虑同步信号还是需要互斥信号。例如,理发师或者沙发在第一次申请的时候,以 Try-p 的方式申请,所以需要对资源进行互斥访问。再次申请高层资源的时候,只要同步信号即可,没有可用资源

则挂起,无需再以 Try-p 的方式进行。因此同一资源处于不同阶段表现有不同的信号形式,例如 count_emptySofa 和 s_emptySofa2、count_barbers 和 s_chairs 这两对信号表示同样资源信号。其一般不要超过两行。

4 语义模式分析与应用

4.1 Try-p 机制的分析应用

Try-p 是一种基于基本 PV 原语的扩展信号量机制。它从面向资源角度试图预先申请临界资源,当申请不到时,转入其它问题处理分支,而不是挂在该资源的等待队列上占用和争夺处理器资源。这有些类似在事务处理中的乐观并发控制,在提交数据前有一个预先有效性确认阶段,确认满足条件才提交,否则事务回滚或者重启或者调整操作次序作为其它的问题处理分支。Try-p 的引用以问题语义变化为基础,例如当理发店内顾客已满时,再来顾客直接离开理发店,而非在门口等待,这样内容语义变化,利用 Try-p 机制能更好地解决该问题。引入 Try-p 的位置的每一个同步信号量将变为同步计数器,并且由一对互斥信号量保护。

语义为门口等待时的部分解决方案如下。

```

Initial values;
s_full=0;// number of full chairs
s_empty=n;// number of empty chairs
barbers=0;// semaphore of available barbers
Customer Process:
P(s_empty)
//sitting in chairs in waiting room;
V(s_full)
P(barbers)
//cutting hair;

```

当语义为选择离开理发店时,引入 Try-p 机制的解决方案如下。

```

Changed Customer Process with Try-p:
P(mutex_empty)//mutex for s_empty
if(s_empty > 0){// Try-p 的使用位置
    //check empty chairs
    s_empty ++;
    V(mutex_empty)
    V(s_full)
    P(barbers)
    //cutting hair
}else{
    V(mutex_empty)
    //leave
}

```

引入 Try-p 机制后,同步信号变为受互斥信号量保护的同步计数器。例如在未使用 Try-p 机制的解决方案中,s_empty 作为同步信号量使用;而在后面的解决方案里则作为同步计数器使用,并由互斥信号量 mutex_empty 保护使用,可以确保复杂语义下的进程同步和通信。对于更复杂情况下的语义描述,需要多重 Try-p 机制实现。尤其对于上节更复杂的理发师模型,信号量控制结构也变得复杂,顾客、理发师和收银 3 个进程需要同时考虑到对同一活动的互斥性和同步性。

(下转第 128 页)

络质量评估过程的合理性。

结束语 IP 网络质量问题一直是一个热门课题。本文将 D-S 证据理论引入网络管理评价中,体现并处理评估中的未知信息。以常见网络应用为导向,建立网络质量评价的两级 D-S 证据融合推理模型,极大程度上削减了不确定性因素的影响,使得评价更趋合理性。后续的研究中,可以将根据两级 D-S 证据融合得出的网络评价结果用于指导网络应用调整、升级或网络故障检测中,也可以进行算法整合,将 D-S 证据理论与其他优化算法相结合用于网络质量评价,对用户的主观体验进行有效分析。

参 考 文 献

[1] 张冬艳,胡铭曾,张宏莉. 基于测量的网络性能评价方法研究[J]. 通信学报,2006,27(10):74-79
[2] 谈杰,李星. 网络测量综述[J]. 计算机应用研究,2006,23(2):5-8
[3] IETF. IP performance metrics(IPPm) metrics registry [M]. RFC4148, Aug. 2005
[4] ITU-T. Network performance objectives for IP-based services [M]Y. 1514, Feb. 2006

[5] 刘怀亮,王东,徐国华. 一种基于流量工程的网络端到端性能分析算法[J]. 系统工程与电子技术,2003,25(9):1165-1168
[6] 蒋序平. 网络性能综合评估方法 IEMONP 的设计和实现[J]. 海军工程大学学报,2006,18(5):74-78
[7] Hershey P, Runyon D, Wang Y W. Metrics for end-to-end monitoring and management of enterprise systems[C]// Proc. 2007 IEEE military communication conference(MILCOM2007). IEEE Press, Oct. 2007:1-7
[8] 阙伟科,张国清,魏郑浩. IP 网络综合性能评估模型[J]. 计算机工程,2008,34:99-101
[9] 陈小帅. IP 网络运行质量分析与系统实现[D]. 北京:北京邮电大学,2008
[10] 杨风暴,王肖霞. D-S 证据理论的冲突证据合成方法[M]. 北京:国防工业出版社,2010
[11] 曹菊英,赵跃龙. 基于 DS 证据推理的信息安全风险评估方法研究[J]. 计算机工程与应用,2009,45(11):129-131
[12] 王玉波. 基于基本概率赋值调整的数据融合方法及应用[D]. 大连:大连理工大学,2010
[13] 倪国强,梁好臣. 基于 Dempster-Shafer 证据理论的数据融合技术研究[J]. 北京理工大学学报,2001,21(5):603-609

(上接第 116 页)

4.2 语义分析与应用

语义可以表现为流程次序或者问题内容的变化。活动流程的变化需要引入多重 Try-p 机制,活动内容的变化则需要从面向资源的角度分析问题。基于语义的理发师问题也是生产者-消费者模型的特例化,需要根据不同的语义环境,灵活运用信号量和 pv 原语机制。

用面向资源的方法,更方便分析和解决进程同步问题中的资源竞争。例如,对于理发师进程,理发师作为进程的驱动者,同时又是 barber 进程和 cashier 进程的互斥资源,也是 barber 进程和 customer 进程同步的临界资源。将面向流程和面向资源的方法结合,更容易解决进程并发中的同步互斥问题。根据语义分析解决问题,更加有现实意义。

理发师的经典模型及其语义变化形成的处理进程同步问题的新模型应用参数如表 1 所列。从表 1 中可清晰地看到复杂语义变化形成的许多新的问题,利用复杂信号量机制从面向资源的角度可以较容易分析和解决复杂语义产生的新问题。例如,在入党流程或者住院流程步骤较多的多信号量的并发工作流程模型中都可以进行应用。

表 1 各种理发师模型使用参数比较

模型	理发师	流程语义	内容语义
参数	经典模型	改变模型	改变模型
顾客数量	M	M	M
收银台数量	—	1	1
理发师数量	1	P	P
理发椅数量	1	P	P
沙发数量	N	N	N
站区位置数量	—	—	Q
缓冲队列数量	1	3	4
是否门口等待	否	是	否
模型分析方法	面向过程	面向资源和 面向过程	面向资源和 面向过程

结束语 根据前述问题的论述和总结可以看出,问题语义不同,解决方案就会有变化。改变理发师问题的原有语义,利用不同的机制对应的解决方案需要修改。用面向资源的方法结合面向流程的模式,能更好地解决进程间同步互斥问题。在本复杂版本的基础上,还可以继续改变问题语义进行变形,比如将收银工作由单独收银员执行,理发师不可以进行收银工作;收银台可以有多个。只要抓住解决该类问题的关键点,选取合理的分析模式,根据具体的语义去分析解决问题就是有经验规则可遵循的。

参 考 文 献

[1] 郭小群,郝克刚,侯红,等. 复杂分支和同步模式的 Pi 演算描述[J]. 计算机科学,2010,37(2):139-141
[2] 李小群,赵慧斌,孙玉芳. 进程间通信机制的分析与比较[J]. 计算机科学,2002,29(11):16-21
[3] 萧美阳,叶晓俊. 并发控制实现方法的比较研究[J]. 计算机应用研究,2006(06):19-22
[4] Dijkstra E W. Co-operating Sequential Processes [M]. Academic Press,1968:43-112
[5] 杨建书,吴尽昭,周瑾. 基于通信顺序进程的 OWL-S 语义分析与建[J]. 计算机应用,2010,30(8):2173-2196
[6] Raghunathan S. Extending Inter-Process Synchronization with Robust Mutex and Variants in Condition Wait [C]// IEEE. 2008:121-128
[7] 王文磊,徐汀荣. 经典进程同步问题的研究与实现[J]. 计算机工程与设计,2006,27(12):2248-2253
[8] Zissulescu C, Kienhuis B, Deprettere E. Communication Synthesis In A Multiprocessor Environment [C]// IEEE. 2005:360-365
[9] 汤子瀛,等. 计算机操作系统 [M]. 西安:西安电子科技大学出版社,1996:121-134