基于多UIO序列的协议一致性测试生成*

刘积仁 都 军

(计算机软件国家工程研究中心,沈阳 110006)

摘要 本文基于多UIO序列提出了一种最优化协议一致性测试生成方法.可以证明本方法生成的测试序列的长度比其它基于UIO序列的测试方法生成的测试序列短.由于本方法采用了形式叠加技术,因此生成叠加转换序列所需要的计算时间大大减少了.

关键词 协议一致性测试,有限状态机,测试序列,UIO序列.

随着形式化描述技术在复杂的分布式系统中的广泛使用,如何从协议描述中构造测试实例以测试协议实现是否与其描述相一致,已经引起了人们越来越大的关注.针对这一问题,近年来,国内外学者提出了不少方法,其中已被广泛采用的是基于传统的有限状态机(FSM)理论的一类方法.这类方法,这里我们仅仅考虑基于FSM的协议的控制部分的实现,方法的实现归纳起来通常包括以下两部分:第1部分,构造FSM中每一状态的状态区别(SI)序列.这样,FSM中的任一状态就可由该状态的一个SI序列唯一标识.典型的SI序列有识别序列(DS)[1],特征集序列(CS)[2]和UIO序列[3].第2部分,构造FSM中每个转换的一个测试子序列,然后将所有这些测试子序列连结起来即构成最终所求的测试序列.这里,一个转换生成的一个测试子序列是由该转换本身和该转换的尾状态的SI序列顺序连结而成的一个转换序列.

由于UIO序列的长度永远小于相应的识别序列或特征集序列的长度,并且绝大多数FSM都存在UIO序列,所以在实际应用中,人们通常采用UIO序列作为状态的SI序列.基于UIO序列的形式化方法有UIO方法[3],SUIO方法[4],MUIO方法[5]和叠加方法[6]等.这些方法的使用大大提高了测试序列生成的效率,使得生成的测试序列在保证较高的错误覆盖的前提下,长度越来越短,其中,尤以叠加方法的效果最好.但是,在计算机软件国家工程研究中心测试环境的实际应用中,我们发现叠加方法存在以下两个方面的不足:(1)由于在叠加过程中仅仅使用最短的UIO序列作为SI序列从而使得叠加进行的不充分,影响了最终叠加的效果;(2)由于叠加过程的计算复杂性非常高,使得当协议规模达到一定程度时,叠加的效率非常低.针对这两个方面的缺陷,我们提出了一种形式叠加方法.实验表明,这种方法较好地解决了上述叠加方法的不足,同时又进一步缩短了测试序列的长度.但是,这种方

* 本文1994-01-12收到,1994-05-03定稿

本课题得到国家863项目基金资助.作者刘积仁,1955年生,教授,主要研究领域为协议工程,分布式多媒体.都军,1966年生,助教,主要研究领域为协议测试,分布式多媒体系统.

本文通讯联系人:刘积仁,沈阳 110006,东北大学,计算机软件国家工程研究中心

法的使用存在一个约束条件,即一个强连通有向图,当它排除某些特定边后也应满足强连通性.尽管在实际协议描述中,这种约束条件不是很难满足,但是为了提高它的通用性,我们又提出了一种新的测试方法,该方法不仅去掉了形式叠加方法的约束条件,而且也较叠加方法[6]大大缩短了测试序列的长度.

1　FSM 模型

本文所涉及的所有形式化方法都基于一个确定的 FSM 模型,该模型由一个五元组表示 $M=(S,I,O,\delta,\gamma)$,其中 $S=\{s_1,s_2,\cdots,s_n\}$是一个有限状态集;$I=\{i_1,i_2,\cdots,i_n\}$是一个输入事件集合;$O=\{o_1,o_2,\cdots,o_n\}$是一个输出事件集合;$\delta:S\times I\to S$ 是一个转移函数;$\gamma:S\times I\to O$是一个输出函数.

一个 FSM M 由一个有向图 $G=(V,E)$表示,其中顶点集 $V=\{v_1,v_2,\cdots,v_n\}$表示 M 的有限状态集 S,有向边集 $E=\{(v_j,v_k;i_{jk}/o_{jk}):v_j,v_k\in V\}$表示 M 的有限转换集,即一条边 $(v_j,v_k;i_{jk}/o_{jk})\in E$ 表示一个从状态 v_j 到状态 v_k 的状态转换(t_{jk}),产生该转换的输入事件和该转换产生的输出事件分别是 i_{jk} 和 o_{jk}.为了方便表示,今后我们用 $head(t_{jk})$,$tail(t_{jk})$和 $label(t_{jk})$分别表示转换 t_{jk}的头状态,尾状态和转换标号.

这里,我们总是假定一个 FSM M 满足:

(1)确定的、具有最小规模和一个初始状态 v_1;

(2)图 G 是强连通的.

只要不引起混淆,本文以下各部分不区分 FSM M 和它的表示图 G,M 的状态和 G 中的顶点,M 中的转换和 G 中的边.

2　定义和引理

定义 2.1. 设图 $G=(V,E)$表示一个给定的 FSM M.不失一般性,设 G 的选择集 $SS(v_j,i/o)$为由 G 中具有相同的尾状态 v_j 和相同的输入输出标号 i/o 的所有转换组成的一个转换集合.

引理 2.1. 设图 $G=(V,E)$表示一个给定的 FSM M.不失一般性,设$(v_j,v_k;i_{jk}/o_{jk})$是 G 中的一个转换.设 $T_{jk}=Label((v_j,v_k;i_{jk}/o_{jk}))$ @ UIO(v_k),其中 UIO(v_k)是状态 v_k 的一个 UIO 序列.如果 G 中以状态 v_k 为尾状态的所有转换中,i_{jk}/o_{jk}作为输入输出标号是唯一的,则 T_{jk}也是状态 v_j 的一个 UIO 序列.

证明:反证法.假设 T_{jk}不是状态 v_j 的一个 UIO 序列,那么至少存在 G 中的一个不同于状态 v_j 的状态,不妨设为状态 v_i,则该状态的输入输出标号是 T_{jk};否则,T_{jk}即是状态 v_j 的一个 UIO 序列.因为 G 中所有以状态 v_k 为尾状态的转换中,i_{jk}/o_{jk}是唯一的,所以当输入 i_{jk} 事件到状态 v_i 时,该状态 v_i 一定转移到一个不同于状态 v_k 的状态,不妨假设为状态 v_l;否则,与假设 i_{jk}/o_{jk}是唯一的矛盾.由于 UIO(v_k)是状态 v_k 的一个 UIO 序列,因此状态 v_l 的输入输出序列一定与 UIO(v_k)序列不同,所以状态 v_i 的输入输出标号序列一定与 T_{jk}不同,显然与假设矛盾.□

引理 2.2[4]. 一个有向图 G 存在一条欧拉回路的充要条件是图 G 强连通且对称.

已知一个强连通有向图 $G=(V,E)$ 和由 G 构造的图 $G'=(V',E')$，其中 $V'=V$，$E'=E\cup E_c$. $G[E_c]$ 是满足下面条件的 G' 的子图：(1)$G[E_c]$ 的顶点集是 E_c 中所有边的头、尾顶点组成的集合；(2)$G[E_c]$ 的边集是 E_c. 当 $G[E_c]$ 的顶点集等于 V' 时，我们称 $G[E_c]$ 是 G' 的一个由边生成的子图. 基于此，我们引入以下两个引理：

引理 2.3. 如果图 G 具有复回能力(即对 G 中的任一顶点，当输入复回输入事件 ri 到该顶点时，伴随着一个 *null* 输出事件的产生，G 直接回到初始状态 v_1)，则 G' 的任一由边生成的子图 $G[E_c]$ 是弱连通的.

证明：见参考文献[4]. □

引理 2.4. 如果图 $G[E_c]$ 是弱连通的，则图 $G[E_c]$ 在 G' 上的对称增广图 G^* 是强连通的.

证明：见参考文献[4]. □

推论 2.1. 如果图 G 具有复回能力，则一定存在 E_c 上 G' 的一条中国邮递员回路.

证明：由引理 2.3、引理 2.4 和引理 2.2 可知，G^* 一定存在一条欧拉回路. 该回路包含 $G[E_c]$ 中的所有边和 E 中的部分边. 显然，该欧拉回路即是 E_c 上 G' 的一条中国邮递员回路. □

3 基于多 UIO 序列的形式叠加改进算法

本算法由以下 6 个步骤实现.

步骤 1. 构造图 G 的选择集；

步骤 2. 构造图 G 的形式叠加转换序列. 这一过程包括以下 3 步：

(a)构造图 G 的一条中国邮递员回路；

(b)按所有选择集中所有转换的个数分割中国邮递员回路，使得分割后的各段转换序列满足：(1)其第一个转换属于某个确定的选择集；(2)除第一个转换外各段不再包含任何选择集中的任何转换；(3)各段的第一个转换互不相同；

(c)检查各段转换序列. 如果某段转换序列的最后一个转换是构造中国邮递员回路时复制的，则删除它，继续检查；否则停止检查，此时的转换序列即构成一个形式叠加转换序列.

步骤 3. 构造最短的 UIO 序列集合 $\{S_j\}$.

这里，不失一般性，设 $O=\{o_{jk}\mid o_{jk}$ 是任一形式叠加转换序列，$head(o_{jk})$ 表示该序列的第一个转换的头状态，$tail(o_{jk})$ 表示该序列的最后一个转换的尾状态$\}$，$F=\{f_{jk}\mid f_{jk}=(v_j,v_k;i_{jk}/o_{jk}),v_j=head(o_{jk})\in V,v_k=tail(o_{jk})\in V,i_{jk}/o_{jk}$ 是 o_{jk} 中所有转换的标号顺序连结而成的标号序列，$o_{jk}\in O\}$. 对每个 $f_{jk}\in F$，构造状态 v_k 的全部最短 UIO 序列并将其组成一个集合，不妨记与 S_k. 显然，状态 v_k 不同，集合 S_k 也不同.

步骤 4. 构造一个强连通图 $G'=(V',E')$，其中 $V'=V\cup H$，$E'=E\cup P\cup Q$. 这里，$H=\{h_k\mid h_k=tail(f_{jk}),f_{jk}\in F\}$；

$P=\{p_{jk}\mid p_{jk}=(v_j,h_k;i_{jk}/o_{jk}),v_j\in V,h_k\in H,i_{jk}/o_{jk}=Lable(f_{jk}),f_{jk}\in F\}$；

$Q=\{q_{km}\mid q_{km}=(h_k,v_m;\mathrm{UIO}_k),h_k\in H,v_m=tail(\mathrm{UIO}_k)\in V,\mathrm{UIO}_k\in S_k\}$.

步骤 5. 构造图 G' 的对称增广图 $G^*=(V^*,E^*)$，这一过程包括以下 2 步：

(a)构造一个网络流模型 $G_f=(V_f,E_f)$，其中 $V_f=V'\cup\{s,t\}$，$E_f=E\cup Q\cup\{(s,h_k)\mid h_k$

$\in H\}\cup\{(v_j,t)|v_j\in V\}$. 这里,$s$、$t$ 分别表示图 G_f 的源点和汇点;$E\cup Q$ 中的边的费用是单位 1,容量无限;$(v_j,t)(v_j\in V)$的费用是 0,容量为 v_j 在 P 中的出度;$(s,h_k)(h_k\in H)$的费用是 0,容量是 h_k 在 P 中的入度.

(b)计算图 G_f 的最小费用最大流 F^*. P 中所有的边和 $E\cup Q$ 中流量大于 0 的边即构成图 G^*. 这里,一个边的流量表示该边在图 G^* 中的个数.

步骤 6. 构造图 G^* 的一个欧拉回路,该回路即是图 G 的一个最小长度的测试序列.

定理 3.1. 当一个形式叠加转换序列在其后面接上该序列的最后一个转换的尾状态的一个最短 UIO 序列后,我们称得到的这个序列为该形式叠加转换序列生成的一个生成序列. 当一个形式叠加转换序列生成一个生成序列时,该形式叠加转换序列中的每一个转换都存在该生成序列中的一个相应的测试子序列.

证明:不妨设一个形式叠加转换序列是:$T_{ij}=(v_i,v_{i+1};i_{i,i+1}/o_{i,i+1})@(v_{i+1},v_{i+2};i_{i+1,i+2}/o_{i+1,i+2})@,\cdots,@(v_{j-2},v_{j-1};i_{j-2,j-1}/o_{j-2,j-1})@(v_{j-1},v_j;i_{j-1,j}/o_{j-1,j})$,其中$(v_i,v_{i+1};i_{i,i+1}/o_{i,i+1})\in SS(v_{i+1},i_{i,i+1}/o_{i,i+1})$,@是转换连结算子. 令 $S_{ij}=T_{ij}@\mathrm{UIO}(v_j)$,其中 $\mathrm{UIO}(v_j)$是状态 v_j 的一个最短 UIO 序列. 显然,$(V_{j-1},v_j;i_{j-1,j}/o_{j-1,i})@\mathrm{UIO}(v_j)(\in S_{ij})$是转换$(v_{j-1},v_j;i_{j-1,j}/o_{j-1,j})$的一个测试子序列. 从形式叠加转换序列的构造过程,我们可以看到 T_{ij} 中除第一个转换$(v_i,v_{i+1};i_{i,i+1}/o_{i,i+1})$外,其余的转换的输入输出标号在以各自的尾状态为尾状态的所有转换中是唯一的,因此,由引理 2.1 可得$(i_{j-1,j}/o_{j-1,j})@\mathrm{UIO}(v_j)$是状态 v_{j-1}的一个 UIO 序列,由此$(v_{j-2},v_{j-1};i_{j-2,j-1}/o_{j-2,j-1})@(v_{j-1},v_j;i_{j-1,j}/o_{j-1,j})@\mathrm{UIO}(v_j)$是转换$(v_{j-2},v_{j-1};i_{j-2,j-1}/o_{j-2,j-1})$的一个测试子序列,显然它在 S_{ij} 中;同理,$(i_{j-2,j-1}/o_{j-2,j-1})@(i_{j-1,j}/o_{j-1,j})@\mathrm{UIO}(v_j)$是状态 v_{j-2}的一个 UIO 序列,$\cdots$,$(i_{i+1,i+2}/o_{i+1,i+2})@\cdots@(i_{j-1,j}/o_{j-1,j})@\mathrm{UIO}(v_j)$是状态 v_{i+1}的一个 UIO 序列,因此$(v_{j-3},v_{j-2};i_{j-3,j-2}/o_{j-3,j-2})@(v_{j-2},v_{j-1};i_{j-2,j-1}/o_{j-2,j-1})@(v_{j-1},v_j;i_{j-1,j}/o_{j-1,j})@\mathrm{UIO}(v_j)(\in S_{ij})$是转换$(v_{j-3},v_{j-2};i_{j-3,j-2}/o_{j-3,j-2})$的一个测试子序列,$\cdots$,$S_{ij}$是转换$(v_i,v_{i+1};i_{i,i+1}/o_{i,i+1})$的一个测试子序列. □

定理 3.2. 如果任一由图 $G=(V,E)$表示的 FSM 具有复回能力,则根据本算法一定存在 E_c 上 G' 的一个中国邮递员回路.

证明:首先做一个变换 $\Gamma:H\to V_h$　　$h_k|\to v_k$

这里,$V_h=\{v_j|j=k,h_k\in H\}$. 在图 G' 上作变换 Γ,可得到一个强连通有向图 $G''=(V'',E'')$,这里 $V''=V$,$E''=E\cup E_c'\cup E_u'$,其中:

$E_c'=\{(v_j,v_k;label(f_{jk}));v_j,v_k\in V,f_{jk}\in F\}$;

$E_u'=\{(v_k,v_l;\mathrm{UIO}_k);v_k\in V,v_l=tail(\mathrm{UIO}_k)\in V,\mathrm{UIO}_k\in S_k\}$.

同时,可得到图 $G\hat{}=(V\hat{},E\hat{})$,这里 $V\hat{}=V$,$E\hat{}=E\cup E_u'$,其中 E_u' 如上定义.

由形式叠加转换序列的构造过程可知,对任意一个 $v_j\in V$,一定存在 E_c' 中的一个转换,该转换的头状态即是 v_j. 因此,$G\hat{}[E_c']$是 G''的由边生成的子图. 因为 G 具有复回能力,所以 $G\hat{}=(V\hat{},E\hat{})$也具有复回能力,由推论 2.1,则一定存在 E_c' 上 G''的一条中国邮递员回路.

由于变换 Γ 是一一变换,在 G''上作变换 Γ 的逆变换 Γ^{-1},则一定存在 E_c 上 G' 的一条中国邮递员回路. □

定理 3.3. 假设 FSM 处于初始状态 v_1,则由本算法得出的 E_c 上 G' 的一条中国邮递员

1.3.3 数据采掘的分类

从不同的视角看,数据采掘技术有几种分类方法[2]:根据发现知识的种类分类;根据采掘的数据库的种类分类和根据采用的技术分类.

• 根据发现知识的种类分类 这种分类方法有:总结(Summarization)规则采掘、特征(Characterization)规则采掘、关联(Association)规则采掘、分类(Classification)规则采掘、聚类(Clustering)规则采掘、趋势(Trend)分析、偏差(Deviation)分析、模式分析(Pattern Analysis)等.如果以采掘知识的抽象层次划分,又有原始层次(Primitive Level)的数据采掘、高层次(High Level)的数据采掘和多层次(Multiple Level)的数据采掘等.

• 根据采掘的数据库分类 数据采掘基于的数据库类型有:关系型(Relational)、事务型(Transactional)、面向对象型(Objected-Oriented)、主动型(Active)、空间型(Spatial)、时间型(Temporal)、文本型(Textual)、多媒体(Multi-Media)、异质(Heterogeneous)数据库和遗留(Legacy)系统等.

• 根据采用的技术分类 最常用的数据采掘技术[11]是:

(1) 人工神经网络:它从结构上模仿生物神经网络,是一种通过训练来学习的非线性预测模型.可以完成分类、聚类、特征采掘等多种数据采掘任务;[12]

(2) 决策树:用树形结构来表示决策集合.这些决策集合通过对数据集的分类产生规则.典型的决策树方法有分类回归树(CART),典型的应用是分类规则的采掘;

(3) 遗传算法:是一种新的优化技术,基于生物进化的概念设计了一系列的过程来达到优化的目的.这些过程有基因组合、交叉、变异和自然选择.为了应用遗传算法,需要把数据采掘任务表达为一种搜索问题而发挥遗传算法的优化搜索能力;

(4) 最近邻技术:这种技术通过 K 个最与之相近的历史记录的组合来辨别新的记录.有时也称这种技术为 K-最近邻方法.这种技术可以用作聚类[13]、偏差分析[14]等采掘任务;

(5) 规则归纳:通过统计方法归纳、提取有价值的 If-Then 规则.规则归纳的技术在数据采掘中被广泛使用,例如关联规则的采掘;

(6) 可视化:采用直观的图形方式将信息模式、数据的关联或趋势呈现给决策者,决策者可以通过可视化技术交互式地分析数据关系.

1.3.4 关联规则(Association Rules)的采掘

关联规则表示数据库中一组对象之间某种关联关系的规则(例如"同时发生"或者"从一个对象可以推出另一个").

在数据采掘的研究领域,对于关联规则采掘的研究开展得比较积极和深入.[15,16,17]介绍一下关联规则采掘的研究情况,可以使人家对于数据采掘的研究有一定的感性认识.

关联规则采掘的一般对象是事务(Transactional)数据库.这种数据库的一个主要应用是零售业,比如超级市场的销售管理.条码技术的发展使得数据的收集变得更容易、更完整,从而存储了大量交易资料.关联规则就是辨别这些交易项目(Item,指交易中的内容.比如,面包、牛奶等都是项目)之间是否存在某种关联关系.例如,关联规则可以表示"购买了项目 A 和 B 的顾客中有 95% 的人又买了 C 和 D".这种关联规则提供的信息可以用作商品销售目录设计、商场布置、生产安排、针对性的市场营销等.

问题是这样描述的[16]:设 $I=\{i_1,i_2,\ldots,i_m\}$ 是 m 个不同项目的集合,D 是针对 I 的交易的集合,每一笔交易包含若干项目 $i_i,i_j,\ldots,i_k\subset I$.关联规则表示为 $X\Rightarrow Y$,其中 $X,Y\subset I$,并且 $X\cap Y=\varnothing$.X 称作规则的前提,Y 是结果.

一般把一些项目的集合称作 itemset.在 itemset 中项目的数量叫作 itemset 的长度.每一个 itemset 都有一个统计的度量称为"支持"(Support):对于 $X\subset I$,如果交易集合 D 中包含 X 的交易个数为 s,则 $support(X)=s$.而一个规则也有衡量的标准称为"置信度"(Confidence),定义为 $Confidence(X\Rightarrow Y)=support(X\cup Y)/support(X)$.

采掘关联规则的问题就是找出这样一些规则,它们的 support 和 confidence 分别大于用户指定的最小 support 和最小 confidence 限度,因此,该问题可以分解成如下两个子问题:

(1) 产生所有 support 大于指定的最小 support 值的 itemset,这些 itemset 称为 large itemset,而其它的称为 small itemset;

(2) 对于每个 large itemset,产生所有比最小 confidence 大的规则,如下:

对于一个 large itemset L 和任何 $S\subset L$,如果 $support(L)/support(L-S)\geq$ minimum-confidence,那么规则 $L-S\Rightarrow S$ 就是有效规则.例如,设数据库中有 4 次交易为 $T_1=\{A,B,C\}$,$T_2=\{A,B,D\}$,$T_3=\{A,D,E\}$,$T_4=\{A,B,D\}$.设

$L_2={}^5ri/null^1$

$L_3={}^2ri/null^1$

$L_4={}^3ri/null^1b/0^2a/0^3b/0^4$

$L_5={}^4ri/null^1$

这里,每个转换的左上角和右上角的数字分别表示发生转换前 M 所处的状态和转换后 M 所处的状态.

接下来,按照广度优先搜索法求解 S_1,S_4,S_5(已表示在表 1 中).

由以上各步的结果,我们可以构造图 G'(如图 2 所示);然后,利用网络流算法构造图 G' 的对称增广图 G^*(如图 3 所示). 构造 G^* 的一个欧拉回路即可得到如下所示的该实例的最优测试序列:

$^1ri/null^1/a/1^4\ b/0^1\ a/1^4\ a/1^5\ a/null^5\ b/1^1\ b/0^2\ b/1^2\ a/0^3\ a/0^5$

$a/null^5\ ri/null^1\ b/0^2\ b/1^2\ ri/null^1\ b/0^2\ a/0^3\ ri/null^1\ b/0^2\ a/0^3$

$b/0^4\ a/1^5\ b/1^1\ a/1^4\ ri/null^1\ a/1^4\ b/0^1$

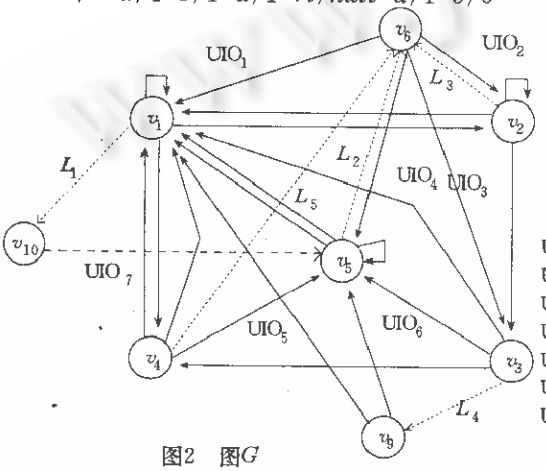

图2　图G

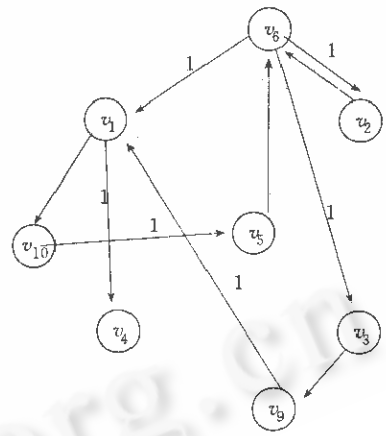

图3　图G　的对称增广G^*

显然,该测试序列的长度是 28,而由叠加算法[6]、MUIO 算法[5]和 SUIO 算法[4]产生的测试序列的长度分别是 31、44、和 51,长度各缩短了 10%、36%和 45%.

这里我们给出的都是弱一致性测试序列,用同样的方法也可产生强一致性测试序列.

5　结　论

本文利用形式叠加技术给出了协议一致性测试生成的一种新算法. 实验表明,该算法与目前测试生成效果最好的叠加算法[6]相比具有如下两个显著的优点:

(1)由于利用形式叠加转换序列来代替对测试子序列直接进行叠加而生成的完全叠加转换序列[6],因而使得叠加计算的复杂性大大降低;

(2)由于本算法的叠加序列是通过一个中国邮递员回路生成的,而回路的走法并没有具体的限制,因此本算法的测试子序列形式叠加效果一定好于叠加算法[6]的直接叠加的效果.

许多实例表明,直接叠加的效果越差,本算法的效果越好. 另外,本算法生成的测试序列的长度的上界是 $O(n^3q)$,这是 n 表示 M 的状态数,q 表示 M 接受的输入符号的个数. 由于

本算法保证了给定 FSM M 的所有转换的输出验证和转移验证因而具有与其它基于 UIO 序列的测试方法相同的错误覆盖能力.

参考文献

1 Gonenc G. A method for the design of fault detection experiments. IEEE Trans. Computer, 1970, **19**: 551—558.

2 Chow T S. Testing software design modeled by finite—state machines. IEEE Trans. Soft. Eng., 1978, **3**: 178—187.

3 Sabnani K K, Dahbura A T. A protocol test generation procedure. Computer Network & ISDN System, 1988, **15**: 285—297.

4 Aho A V, Dahbura A T, Lee D *et al*. An optimization technique for protocol conformance test sequence generation based on UIO sequence and rural Chinese Postman Tours. In: Aggarwal S, Sabnani K eds., Protocol Specification, Testing and Verification 8, North—Holland, 1988. 75—86.

5 Shen Y N, Lombardi F, Dahbura A T. Protocol conformance testing using multiple UIO sequences. In: Brinksma E, Scollo G eds. Protocol Specification, Testing and Verification 9, North—Holland, 1989. 131—144.

6 Yong B, Ural H. Protocol conformance test generation using multiple UIO sequences with overlapping. In: Proc. ACM SIGCOMM'90, Philadelphia, DA, 1990. 118—125.

PROTOCOL CONFORMANCE TEST GENERATION BASED UPON MULTIPLE UIO SEQUENCES

Liu Jiren Du Jun

(*National Engineering Research Center of Computer Software, Shenyang* 110006)

Abstract This paper proposes an optimization method for protocol Conformance test generation based upon multiple UIO sequences. It can be shown that test sequences generated by this method are shorter than those generated by other methods using UIO sequences. Because of employing formally overlapping technique, generation time of overlapped transition sequences is greatly decreased.

Key words Protocol conformance testing, finite state machine, test sequences, UIO sequences.