第 6 卷第 3 期
1995 年 3 月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 6, No. 3
Mar. 1995

一种基于形式化描述的测试序列生成改进方法*

张立东　刘积仁　李华天

(东北大学计算机系，沈阳 110006)

摘要　自动生成测试序列始终是软件工程中一项极为困难的工作. 国际标准化组织(ISO)一直致力于协议一致性测试方法与形式化描述技术的研究. 本文讨论了基于形式化描述的协议测试序列生成方法中的问题，特别是提出了"部分 T 序列叠加算法"对使用 UIO 序列生成测试序列的方法做了进一步改进，大大减少了测试开销.

关键词　计算机网络，协议工程，一致性测试，协议测试，软件测试，测试生成.

协议一致性测试方法的研究近年来得到了极大的关注. 它的问题主要集中在两大方面：测试序列生成与选择和测试器的构造. 测试器的构造近年来已经取得了很大进展，本文主要讨论的是测试序列自动生成问题. 一致性测试是用来保证协议的实现与所描述的协议一致. 而测试序列的覆盖率(用来保证测试的可靠性)和测试序列的长度(用来保证测试的可行性)始终是一种折衷. 我们很难达到一种非常满意的结果.

近年来，ISO 一直致力于对协议形式化描述方法的研究. 在抽象描述技术的支持下，人们可以忽略一些具体的细节而保证它的实现结构的正确性. 因而，也为协议描述后面的工作(实现、模拟、验证、一致性测试等)提供了技术保证. 在 FDT 技术支持下，我们可直接从形式化规范中生成测试序列. 它的好处是：(1)测试独立于实现. 这样能防止许多错误；(2)生成的测试序列更容易检查. 因为它比实现代码要抽象得多；(3)最重要的是这样生成测试序列的过程可避免许多不必要的细节提高生成效率.

一般来讲，从 FD 到生成测试序列有两种途径：一种是由形式化描述直接生成测试序列. 这种方法依然面临着涉及太多描述细节的问题. 另一种最常用的方法是把由 FDT 得到的 FD 映象到一个中间模型来，再从这个模型转换生成出测试序列. 这种方法因而引出的问题是：是否有一种方法能把用某种形式化描述语言描述的 FD 的全部内容都转换到这个模型中. 换句话说，是否能找到一种变换，使得它一方面适于生成测试序列；另一面这个变换提供了用某种 FDT 方法描述的信息. 于是，测试序列的生成问题首先是一个变换问题：即找到

* 本文 1992－04－06 收到，1992－08－07 定稿

本课题研究得到国家教委博士点基金的资助.作者**张立东**，1961 年生，博士，主要研究领域为协议工程，软件工程. **刘积仁**，1955 年生，教授，主要研究领域为计算机网络，协议工程，软件开发环境. **李华天**，1922 年生，教授，博士导师，主要研究领域为计算机应用，网络协议工程方法学，软件开发工具，环境方法学.

本文通讯联系人：张立东，沈阳 110006，东北大学软件中心

一种合适的中间模型(IM).它的目的是:(1)找到从 FD 到 IM 的(自动)变换方法;(2)找到从 IM 的描述中(自动)生成测试序列的方法.现有的研究结果集中在四种模型上:(1)ACT 模型(Asynchronous Communication Tree);(2)图模型;(3)IO－EFSM 模型(基于输入/输出的扩展有限状态机模型);(4)LTS 模型(Labelled Transition System).其中图模型是比较通用的方法(三种 FDT 方法都可使用),但它的缺点亦很明显,比较复杂且不易自动实现.而其它三种方法都只适用于一种 FDT 描述语言.但 IO－EFSM 模型与 LTS 模型是可以相互变换的.所以我们说,它们的本质是一致的.

本文,我们重点讨论基于有限状态机的模型.一个确定的有限状态自动机 M 可以用五元组(I,O,E,TF,OF)来表示.其中 I 表示输入集合,O 表示输出集合,E 为 Mi 的状态集合,其中 $M0$ 为初始状态,TF 为转移函数 $I\times E\rightarrow E$,OF 为输出函数 $I\times E\rightarrow O$(见图 1).

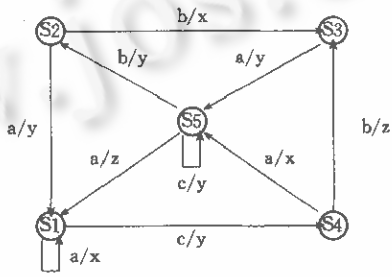

图1　用图来表示的FSM示例(示例选自Bosik的文章)

测试的基本思想是对 FSM 描述中所有的状态变换可使用以下 3 步[2]:

步骤 1:把 FSM 的实现置到某个状态;

步骤 2:施加某个输入并观察它的输出;

步骤 3:检查 FSM 新的状态并确定终状态满足规范要求.

基于 FSM 的测试序列生成方法,在近 10 年的研究中,主要有 4 种:(1)T 方法;(2)W 方法;(3)D 方法;(4)UIO 方法.T 方法(Transition Tour method)生成的测试序列是基于 FSM 的描述.它的基本思想是遍历状态机所有的变迁.开始的实现方法是随机的.直到所有的变迁至少已经过一次.Uyar 和 Dahbura[1]的工作,借助于"中国邮递员"方法,保证了生成的测试序列具有最小的开销.

D 方法(Distinguishing_sequences)的研究是基于 checking_experiments 的结果.这一方法被看作"解决了测试的观测性问题".它是由 3 部分构成:(1)初始序列;(2)状态识别序列;(3)变迁检查序列.DS 使每个状态对输入序列的反应不同于基它状态.这是通过构造可识别树得到的.

W 方法包括了两个输入序列集 W_set 和 P_set.W 集是 FSM 最小的特征集.它使得每对状态能够互相区分.即对不同的点应用 W 集观察到的输出应是能够区分的.P 集包括了所有部分路径,由一个测试树构成.它的根结点是初始状态,而每个变迁只出现一次.它的优点是这种方法适于描述非完备的自动机.但它对每一个点的测试都必须使用 Reset 功能.

UIO 方法的基本步骤是(1)对每个变迁 Si →Sj 找到从初始点到 Si 的序列.一般是采取 Dijikstra 的最短路径方法用 P(i)来表示.(2)使输入按照 Si →Sj 方向变迁;(3)对 Sj 点应用

UIO 序列.因此,这个算法的关键是如何选择 UIO 序列.

从这 4 种方法来看,测试的能力(或测试覆盖率)直接影响着测试序列长度.尽管 W 方法(或者是"部分 W 方法")它能发现比较复杂的错误,但我们认为:(1)它的测试序列长度在实际的复杂系统中很难使用;(2)在通常的实现中,已去掉了一些错误,因而有些测试能力"是多余的".D 方法的主要优点是能发现故障点.但由于(1)D 序列不能总是保证存在;(2)序列优化方法比较难,序列长度比较长;(3)一致性测试本身并不关心错误的原因和位置.因而许多人在实际中都更倾向于 UIO 方法.

1 使用 UIO 序列的基本方法与改进

UIO 序列的主要优点有两项:(1)由于 UIO 序列只是 D 序列或者 W 序列的子集,所以 UIO 序列要比它们短得多;(2)(在实际中)几乎所有的 FSM 都存在着 UIO 序列(除非 FSM 具有等价的状态),因而它具有普遍适用的特点.求某一点 UIO 序列的基本方法是:

(1)建立所有的边标号与输入输出集的关系;

(2)对每个状态求出所有长度为一的输入输出序列;

(3)检查它们是否唯一.如果是,这个状态的 UIO 序列就找到了;

(4)如果不是,对没有 UIO 序列的状态,长度为 2 的序列;

(5)从长度为 K 的 IO 序列中继续求出长度为 $K+1$ 的输入输出序列,检查它们是否唯一,直到对每个状态都找到 UIO 序列或者 K 的长度超过 $2n^2$(见表 1).

表 1 图 1 各点的 UIO 序列

状态	(最短)UIO 序列	UIO 序列开销
S1	a/x,a/x	2
S2	b/x	1
S3	a/y,a/z	2
S4	b/z	1
S5	a/z	1

UIO 基本方法对每个边 A(Si,Sj)的序列为:

ri/null@P(S0,Si)@A(Si,Sj)@UIO(Sj)

其中@ 表示边的相连,P(S0,Si)表示的是从状态 S0 到 Si 的最短路.

ri/null@P(S0,Si)在 UIO 序列的作用只是把系统的运行点置到状态 Si.因此,如果把 A(Si,Sj)@UIO(Sj)看成是图中的一个新边,只要保证在图的游历中每个新边至少经过一次(原有的边只有辅助作用),那么整个测试过程就都经历过了.基于 T 序列中所用的"中国邮递员问题"算法,1988 年 Aho,Dahbura 等人提出了 UIO 序列的优化方法.这一方法的基本思想是把所有点的特征序列中的头、尾两个端点连在一起构成一个图 G′,这个图与原有的图 G 一起构成一个新图.只要按照优化方法求出新图的 T 序列,就可得到一个优化的 UIO 序列(见表 2).这一思想带动了一系列新方法的出现.

表 2　Aho 等方法产生的 UIO 测试序列

a/x,a/x,a/x,C/y,/b/z,a/y,b/y,b/x,a/y,a/z,
c/y,a/x,a/z,c/y,a/x,c/y,a/z,c/y,a/x,b/y,
b/x,a/y,a/z,c/y,a/x,b/y,a/y,a/x,a/x,c/y,a/x,
a/z,a/x,a/x,c/y,b/z,a/y,a/z

1989 年 Shen、Lombardi 和 Dahbura 等人[11]又提出了多 UIO 序列的方法. 这一方法是前一方法的延伸. 它的优化思路是：每一个结点可能有多个 UIO 序列. 当一个节点的某个 UIO 序列与另一个节点的某一 UIO 序列相邻时，这样会减少整个路径的开销. 因而，它考虑从每个点的多个 UIO 序列中选择出“对接效果”最好的序列来.

其后根据实际经验，我们从 T 序列的角度出发，提出了重新认识 T 序列纠错能力的问题. 这是因为，在实践中我们发现 T 序列的纠错能力并不象人们说得那么差. 经过分析我们发现扩充的 T 序列常常等同于 UIO 序列的效果. 这一想法产生了“扩展 T 序列”的方法.

然后顺着这一思路我们和 Yang and Ural 等人，又先后提出了“多个 UIO 序列相互叠加”的不同方法. 这些方法进一步提高了 UIO 序列生成算法的效率. 但这些方法还未能有效地解决叠加中的选择问题. 所以，本文我们提出了“部分 T 序列叠加算法”来解决这一问题.

这一想法是先从 T 序列来考虑整个 UIO 序列. 即我们首先要检查一个 T 序列集在多大程度上属于 UIO 序列集. 然后，再把未包含着的 UIO 序列与已包含的那部分 T 序列按照“叠加算法”重新组合. 整个算法包含检查算法和叠加算法两大部分.

检查算法可按以下方法构造：

1.（按照“中国邮递员算法”）首先构造出该图的 T 序列；

A. 求出图的对称增广图（见图 2）. 即添加适当的变迁（增加图的边）使得该图每个结点的输入变迁等同于输出变迁；

这一算法有用线性规划方法来实现的，我们的经验是直接计算会比线性规划方法效率更高.

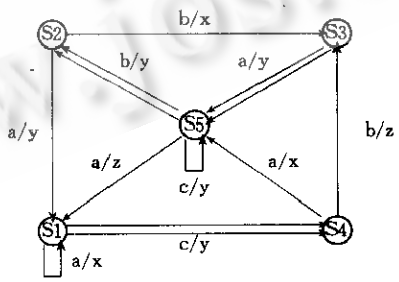

图2 图例的对称增广图

B. 求出该图的“欧拉回路”来.

如图例中的 T 序列为：

$$^{S1}c/y^{S4}a/x^{S5}c/y^{S5}b/y^{S2}b/x^{S3}a/y^{S5}$$

$$a/z^{S1}a/x^{S1}c/y^{S4}b/z^{S3}a/y^{S5}b/y^{S2}a/y^{S1}$$

2. 对图的每个边判断它在 T 序列上是否存在着 UIO 序列. 并把有 UIO 序列存在的那

部分序列构成一个新的集合.

A. 若不存在并且该边无标记则把这一个边从 T 序列中去掉.

判别准则为:(1)相同的 I/O 状态序列交于一点;(2)在 T 序列长度限制下无区别;

B. 若存在则对 UIO 序列所在的各个边做上标记. 把有 UIO 序列的边从边集合中去掉;

判别方法为:(同 UIO 的基本方法类似)只要无相同的 I/O 状态序列. 就可认为是 UIO 序列.

C. 若某一点的 UIO 序列中每一边都只有一个序列使用(无叠加边),则取消这个 UIO 序列. (在下一算法中重新优化)这样新的 T 序列可能是几个序列的集合.

叠加算法:

(1)把 T 序列构成的集合看成是一个固定的 UIO 序列,对图中剩下的各个边求出它们的多个 UIO 序列;

(2)构成一个增广图 G';

(3)产生出最小长度的测试序列;

更详细的实现过程可参考文章末尾的参考文献.

用"部分 T 序列叠加算法"所求得的示例中的 UIO 测试序列为:

ri/null,c/y,a/x,c/y,b/y,b/x,a/y,a/z,a/x,c/y,b/z,a/y,b/y,a/y,a/x,a/x

上述"部分 T 序列叠加算法"中测试序列与 UIO 序列的对应关系为:

c/y,a/x,c/y,b/y,b/x,a/y,a/z,a/x,c/y,b/z,a/y,b/y,a/y

c/y. a/x,c/y,b/y (S1,S4)

a/x. c/y,b/y (S4,S5)

c/y. b/y (S5,S5)

b/y. b/x,a/y,a/z (S5,S2)

b/x. a/y,a/z (S2,S3)

a/y. a/z (S3,S5)

a/z. a/x,c/y,b/z (S5,S1)

a/x. c/y,b/z (S1,S1)

b/z. a/y,b/y (S4,S3)

其中"."符号的前面部分表示的是 UIO 序列中所判断的边,后面部分是相应的特征序列,即该点的 UIO 序列.

考虑到初始化 ri/null 这一步,例子中的单一的 UIO 序列方法(Aho 的方法)所用的序列长度为 39;我们提出的"部分 T 序列叠加算法"长度为 16. 在这个例子里减少了约 59%. 同以前最好的叠加方法相比大概要多减去百分之二十几.

2 结 论

本文我们主要提出了"部分 T 序列优化算法"用于协议一致性测试生成的过程. 算法中最重要的思想是用 T 序列来制导整个叠加过程. 在本文的示例中,该算法不仅要比先前的 MUIO 和 SUIO 方法要好 60%左右,比二种简单的叠加方法(如 FOTS)也要好 20%左右. 因为 T 序列的制导方式要比叠加方式时间复杂度低,所以搜索时间要比简单叠加方式还要

短.它是在效率和效果方面都比较好的方法.

这一方法使用效果最好的情形是在图中每点的UIO序列比较多的时候.值得庆幸的是:实际中的协议大多是这样的.因此它也是比较实用的算法.在最坏的情形下(这种可能机会很小),算法的结果(序列长度)也要好于其它算法.所以即使我们还未能给出更确定性的证明,我们仍可确信它是一个有效的算法.

参考文献

1 Aho A V *et al*. An optimization technique for protocol conformance test generation based on UIO sequences and Chinese postman tours. Protocol Specificaition, Testing and Verification, VII(North Holland),75—86.

2 Bochmann G V, Sunshine C A. A survey of formal methods. IEEE Trans. on Software Eng., 1989,**15(4)**:413—426.

3 Sabnani K K, Dahbura A T. A protocol testing procedure. Comput. Net Works ISDN Systems, 1988,**15(4)**:285—297.

4 Shen Y N, Lombarde F, Dahbbura A T. Protocol conformance testing using multiple UIO sequences. Protocol Specification, Testing and Verification, XI(Holland), June 1989.

5 Sidhu D P, Leung T K. Formal methods for protocol testing: a detailed study. IEEE Trans. on Software Eng., 1989,**15(4)**:413—426.

6 Yang B, Ural H. Protocol conformance test generation using multiple UIO sequences with overlapping. Computer Communication Review, 1990,**(4)**:118—125.

7 张立东,刘积仁,李华天.一种新的网络协议测试序列生成算法.计算机工程与应用,1990,**(10,11)**:149—153.

8 张立东,刘积仁,李华天.协议一致性测试序列生成方法.小型微型计算机系统,1991,**12(9)**:19—23.

AN IMPROVED TEST SEQUENCES GENERATION METHOD BASED ON FORMAL DESCRIPTION TECHNIQUE

Zhang Lidong　Liu Jiren　Li Huatian

(*Department of Computer Science, Northeastern University, Shenyang* 110006)

Abstract　The selection of appropriate test cases is an important issue for conformance testing of protocol implementations as well as in software engineering. This paper describes an optimization method for reducing the length of protocol conformance test sequences by "Partial T sequences overlapping method", which are obtained using UIO sequences. This method provides a logical link between T method and UIO method. It is shown that test sequences generated by this method can shorter 10%—60% than those generated by other methods employing UIO sequences in an example.

Key words　Computer network, protocol engineering, conformance testing, protocol testing, software testing, testing generation.