

协议一致性测试生成的路径叠加方法^①

汪志宾, 周 颢, 赵保华

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘 要: 基于 UIO 序列提出了一种新的协议一致性测试生成方法——路径叠加方法。其核心思想是寻找有限状态机中无重复转换的路径作为路径叠加转换序列, 在其后连接该序列尾状态的 UIO 序列, 从而验证该序列中所有转换。理论与实验结果表明, 由路径叠加算法生成的测试序列长度要比其他基于 UIO 序列的算法生成的测试序列明显缩短。特别的, 利用路径叠加转换序列来替代 OUIO 方法中对子序列直接叠加而生成的完全叠加转换序列, 在检错性不降低的前提下, 大大降低了叠加计算的复杂度。

关键词: UIO 序列; 路径叠加转换序列; 协议一致性测试; 有限状态机

Path Overlapped Method for Protocol Conformance Test Generation

WANG Zhi-Bin, ZHOU Hao, ZHAO Bao-Hua

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: A new formal method named path overlapped method is proposed for protocol conformance test generation based on UIO sequences. The main idea is to find a path without repeating transition in FSM as a POTS(path overlapped transition sequence), which is a sequence of distinct transitions such that if it is followed by a UIO sequence for the end state of the last transition in the sequence then all the transitions in the sequence are verified. According to the theoretical and experimental results, the test sequences generated by the path overlapped method are substantially shorter than those generated by other methods employing UIO sequences. Particularly, because of employing path overlapped transition sequence as a substitute for FOTS (fully overlapped transition sequence) in OUIO, generation time of overlapped sequence is greatly decreased.

Key words: UIO sequence; path overlapped transition sequence; protocol conformance testing; FSM

协议一致性测试作为衡量协议实现与协议描述之间是否保持一致的一种尺度而得到越来越广泛的使用。在协议一致性测试中, 关键问题是如何生成高效的测试序列^[1]。针对这一问题, 目前已被广泛采用的是基于传统的有限状态机(FSM)理论的 UIO 测试序列生成方法^[2-6]。其中主要有 UIO 方法^[2]、SUIO 方法、MUIO 方法、OUIO 方法^[3]等。其中 OUIO 方法提出了完全叠加转换序列(FOTS, Fully Overlapped Transition Sequence)的概念, 尽可能的将每个转换的测试序列在重叠的地方连结, 从而产生更短的测试序列。结果表明, OUIO 方法产生的测试序列长度比 MUIO 方法缩短

近 30%, 比 SUIO 方法缩短近 40%, 是效率最高的一种方法。但是, 在对 OUIO 方法进行实际应用时, 发现 OUIO 方法存在两个方面的不足: 1)当 FSM 中状态的最短 UIO 序列比较长时, OUIO 在进行叠加生成完全叠加转换序列时, 进行的不充分, 从而回归到 MUIO 方法; 2)FOTS 的计算复杂度非常高。本文在 OUIO 方法的基础上提出了一种路径叠加方法, 既缩短了测试序列的长度, 又在一定程度上避免了以上不足。1 有限状态机路径叠加方法

一个有限状态机 M 可以用一个六元组 $M = (S, I, O, \delta, \lambda, s)$ 表示, 其中:

① 基金项目:国家自然科学基金(60872009);国家高技术研究发展计划(863)(2009AA01Z148);安徽高校省级自然科学基金计划(ZD2008005-2, ZD200904, JK2009A013, JK2009A025)

收稿时间:2010-11-05;收到修改稿时间:2010-12-10

$S = \{s_1, s_2, \dots, s_n\}$ 是一个有限状态集;
 $I = \{i_1, i_2, \dots, i_n\}$ 是一个输入事件集合;
 $O = \{o_1, o_2, \dots, o_n\}$ 是一个输出事件集合;
 $\delta: S \times I \rightarrow S$ 是一个转移函数;
 $\lambda: S \times I \rightarrow O$ 是一个输出函数;
 $s \in S$ 是 M 的初始状态.

有限状态机 M 可由有向图 $G = (V, E)$ 表示, 其中顶点集 $V = \{v_1, v_2, \dots, v_n\}$ 表示 M 的有限状态集 S , 有向边集 E 中的每条有向边代表着 M 中相应的状态间转换. 在 M 中每个状态有其输入集 I 和输出集 O , 相应的, G 中的每条边可以用一个输入事件 $i_k \in I$ 和一个输出事件 $o_l \in O$ 来标记. 因此, G 中的每条有向边都可以用一个三元组 $(v_i, v_j; i_k / o_l)$ 来唯一表示, 图 1 就是一个用图来表示 FSM 的实例.

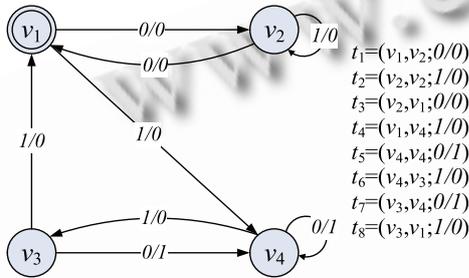


图 1 一个 FSM 实例

G 的一条路径(path)是一个有限非空连续有向边的序列^[7]. 如图 1 中: $W = (v_1, v_2; 0/0), (v_2, v_2; 1/0), (v_2, v_1; 0/0), (v_1, v_4; 1/0), (v_4, v_3; 1/0)$ 就是一条路径. 注意到, 一条路径也可以仅由一条有向边组成. 我们将 W 中初始节点 v_1 记作 $head(W)$, 尾节点 v_3 记作 $tail(W)$.

本文总是假定有限状态机 M 满足:

- 1) M 是确定的, 具有最小规模的, 而且不考虑 FSM 的复回能力;
- 2) G 是强连通的^[7];
- 3) 不区分 M 中的状态与 G 中的顶点, M 中的转换与 G 中的有向边.

2 路径叠加方法

2.1 测试序列叠加过程

OUIO 方法^[3]的基本思想: 对每个状态转换 t_i 启发式的生成最优测试序列(TEST(t_i)), 以便在叠加的过程可以产生尽量多的重复, 以便更好的叠加.

对图 2 所示的 FSM 从状态 v_1 开始的一个最长 FOTS $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$ ^[3] 的测试序列生成过程可以用表 1 来说明.

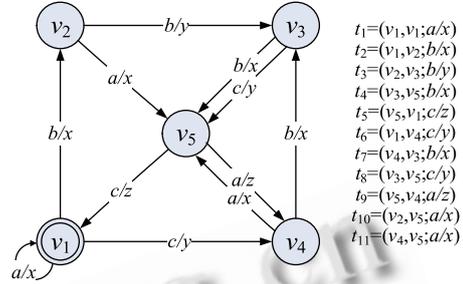


图 2 文献[3]中 FSM 实例

表 1 OUIO 测试序列叠加过程

转换	测试序列(TEST(t_i))
t_1	a/x b/x b/y
t_2	b/x b/y
t_3	b/y b/x c/z
t_4	b/x c/z
t_5	c/z c/y b/x
t_6	c/y b/x
t_7	b/x c/y a/z
t_8	c/y a/z
t_9	a/z b/x c/y
total	a/x b/x b/y b/x c/z c/y b/x c/y a/z b/x c/y

最终的测试序列 a/x b/x b/y b/x c/z c/y b/x c/y a/z b/x c/y 对 $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$ 中所有转换都进行了验证. 显然, 这大大缩短了最终的测试序列, 但是对于最短 UIO 较长的情况(如图 1 所示的 FSM, 表 2 为各状态的全部最短 UIO), 结果却不尽如人意.

表 2 图 1 中各状态的 UIO 序列

状态	UIO
v_1	0/0 1/0 0/0
v_2	0/0 1/0 0/1
v_3	1/0 0/0 1/0 0/0
v_4	1/0 1/0 0/0 1/0 0/0

利用 OUIO 方法几乎找不到长度大于 1 的 FOTS. 我们可以看到在 OUIO 方法中, 对每个转换的测试序列生成仅仅借助于最短的 UIO, 认为这样才能生成最短的测试序列, 从而导致寻找 FOTS 时受到很大的限制, 甚至出现找不到的现象. 以下我们将介绍一种简

单的叠加方法来解决这一问题。

定义 1. 设 $G=(V,E)$ 表示一个给定的 FSMM, 在 M 中, 如果有状态 s_1, \dots, s_k 经过同样的输入/输出 i/o 转换到同一个状态 S , 则相应的在 G 中, 称这些带有 i/o 标记的有向边 $(v_1, v; i/o), \dots, (v_k, v; i/o)$ 为汇聚边 (converging edges)^[8]。

定理 1. 设 $G=(V,E)$ 表示一个给定的 FSMM, $(v_j, v_k; i_{jk}/o_{jk})$ 是 G 的一条转换且非汇聚边, 则 $TEST(v_j, v_k; i_{jk}/o_{jk})=i_{jk}/o_{jk}@UIO_k$ (UIO_k 是 v_k 的一个 UIO 序列, @为连结符) 是状态 v_j 的一个 UIO 序列。

证明: 反证法。假设 $TEST(v_j, v_k; i_{jk}/o_{jk})$ 不是状态 v_j 的一个 UIO 序列, 那么在中至少存在一个不同于 v_j 的状态也可以通过 $TEST(v_j, v_k; i_{jk}/o_{jk})$ 正确转移到另一个状态, 不妨设为 v_i 。因为 $(v_j, v_k; i_{jk}/o_{jk})$ 非汇聚边, 则状态 v_i 通过 i_{jk}/o_{jk} 不会到达状态 v_k , 不妨设为状态 v_l 。而此时, 状态 v_l 具有输入输出 UIO_k , 这与 UIO_k 是 v_k 的一个 UIO 序列矛盾。综上, 可以得出结论: $TEST(v_j, v_k; i_{jk}/o_{jk})=i_{jk}/o_{jk}@UIO_k$ 是 v_j 的一个 UIO 序列。

根据定理 1, 我们可以先移除所有汇聚边, 在剩下的转换中寻找无重复转换的路径, 在其后连结尾状态的 UIO 序列, 就可以验证该路径中所有转换, 我们将这样的路径称之为路径叠加转换序列。本文仅考虑最长的路径叠加转换序列。

对于图 1 所示的 FSM, 先将所有汇聚边都移除, 即移除 t_5, t_7 , 然后寻找从初始状态 v_1 开始的最长的路径叠加转换序列, 如 $\{t_4, t_6, t_8, t_1, t_2, t_3\}$, 其测试序列叠加过程如表 3 所示。

表 3 路径叠加转换序列叠加过程

转换	测试序列(TEST(t_i))
t_4	1/0 1/0 1/0 0/0 1/0 0/0 0/0 1/0 0/0
t_6	1/0 1/0 0/0 1/0 0/0 0/0 1/0 0/0
t_8	1/0 0/0 1/0 0/0 0/0 1/0 0/0
t_1	0/0 1/0 0/0 0/0 1/0 0/0
t_2	1/0 0/0 0/0 1/0 0/0
t_3	0/0 0/0 1/0 0/0
total	1/0 1/0 1/0 0/0 1/0 0/0 0/0 1/0 0/0

由定理 1, 路径叠加转换序列 $\{t_4, t_6, t_8, t_1, t_2, t_3\}$ 的尾状态 v_1 后添加 $UIO(v_1)=0/0 1/0 0/0$ 后, $TEST(t_3)=0/0$

$0/0 1/0 0/0$ 既可以作为转换 $t_3=(v_2, v_1; 0/0)$ 的测试序列, 又可以作为 v_2 的一个 UIO 序列, 从而紧跟在 t_2 的后面可以作为 t_2 的测试序列, 以此类推, 最终测试序列 1/0 1/0 1/0 0/0 1/0 0/0 0/0 1/0 0/0 对 $t_4, t_6, t_8, t_1, t_2, t_3$ 都进行了验证。我们有以下定理。

定理 2. 设 $G=(V,E)$ 表示一个给定的 FSMM, W 是 G 的一个路径叠加转换序列, 在 W 后连接状态 $tail(W)$ 的一个最短 UIO 序列后得到的测试序列 V 能验证 W 中的所有转换, 即对 W 中每个转换, 都能找到 V 中一个子序列为其测试序列。

证明 设 $W=(v_{j_1}, v_{j_2}; i_{j_1j_2}/o_{j_1j_2}) (v_{j_2}, v_{j_3}; i_{j_2j_3}/o_{j_2j_3}) \dots (v_{j_{k-2}}, v_{j_{k-1}}; i_{j_{k-2}j_{k-1}}/o_{j_{k-2}j_{k-1}}) (v_{j_{k-1}}, v_{j_k}; i_{j_{k-1}j_k}/o_{j_{k-1}j_k})$ 。 $V=W@UIO(v_{j_k})$, 其中 $UIO(v_{j_k})$ 为 v_{j_k} 的一个最短 UIO 序列, 那么 $(v_{j_{k-1}}, v_{j_k}; i_{j_{k-1}j_k}/o_{j_{k-1}j_k})@UIO(v_{j_k})$ 是转换 $(v_{j_{k-1}}, v_{j_k}; i_{j_{k-1}j_k}/o_{j_{k-1}j_k})$ 的一个测试序列。由 W 的构造过程可以确定 W 中不存在汇聚边, 由定理 1, $(v_{j_{k-1}}, v_{j_k}; i_{j_{k-1}j_k}/o_{j_{k-1}j_k})@UIO(v_{j_k})$ 是状态 $v_{j_{k-1}}$ 的一个 UIO 序列, 那么 $(v_{j_{k-2}}, v_{j_{k-1}}; i_{j_{k-2}j_{k-1}}/o_{j_{k-2}j_{k-1}})(v_{j_{k-1}}, v_{j_k}; i_{j_{k-1}j_k}/o_{j_{k-1}j_k})@UIO(v_{j_k})$ 是转换 $(v_{j_{k-2}}, v_{j_{k-1}}; i_{j_{k-2}j_{k-1}}/o_{j_{k-2}j_{k-1}})$ 的一个测试序列。以此类推, $V=W@UIO(v_{j_k})$ 是转换 $(v_{j_1}, v_{j_2}; i_{j_1j_2}/o_{j_1j_2})$ 的测试序列。

2.2 完整的测试序列生成过程

对一个用 $G=(V,E)$ 表示的有限状态机 M , 测试序列生成过程包括以下几个步骤:

步骤 1. 将 G 分割成 $m+1$ 个子图 (m 为汇聚边的数量)。该过程可以分为以下 3 步:

- 1) 找出 G 中所有 m 个汇聚边;
- 2) 将所有汇聚边分别组成子图, $\{G_i=(V_i, E_i) | E_i$ 中仅包含一条汇聚边, V_i 包含 E_i 中汇聚边的起点和终点};
- 3) 将移除汇聚边及孤立状态后剩下的所有转换及状态组成一个子图。

步骤 2. 找出每个子图所有路径叠加转换序列。

路径叠加转换序列寻找策略:

- 1) 优先将 M 的初始状态作为序列的初始状态;
- 2) 如果存在多个以同一状态开始的转换时, 优先选择到自身的转换。

路径叠加转换序列生成算法如下:

算法 1.

输入: 由步骤 1 生成的子图 G_s.
 输出: G_s 的路径叠加转换序列集合 S.
 Step 1 将 G_s 中所有转换置为未标记状态, 并将 S 置为空;
 Step 2 v0 = null; L=null; //L 为转换序列
 Step 3 if(G_s 中存在原图 G 的初始状态 v 且存在以 v 开始的未被标记的转换)
 Step 4 v0 = v;
 Step 5 else if(G_s 中存在一个未标记的转换 t)
 Step 6 v0 = head(t);
 Step 7 else
 Step 8 return;
 Step 9 endif
 Step 10 while(G_s 中存在以 v0 开始的未被标记的转换)
 Step 11 t 为以 v0 开始的未标记转换(如果存在以 v0 开始的未标记转换时, 优先选择到自身的转换);
 Step 12 将 t 添加到 L 的末尾;
 Step 13 标记 t;
 Step 14 v0 = tail(t);
 Step 15 endwhile
 Step 16 将 L 添加到 S 中, 返回 Step2.

关于算法 1 我们有以下定理。

定理 3. 对于图 G 的任一个子图, 至少存在一条可终止的路径叠加转换序列, 并且其中无重复转换。

证明: 步骤 1 中生成的子图至少存在一个转换, 由算法 1, 该转换必然能通过 while 循环被添加到 L 中, 所以必然存在一条路径叠加转换序列。

在算法 1 中, 只有 while 循环才向路径叠加转换序列中添加转换, 终止条件是已有序列的尾状态后的转换都被标记, 而由 while 循环体可知, 在向序列 L 中添加转换 t 时会标记 t, 只要 L 中已经存在转换 t, 那么该转换 t 一定不会再次进入到 while 循环内, 也就不会再次出现在 L 中, 因此路径转换序列中没有重复转换, 又因为 G 中的转换是有限的, 那么路径叠加转换序列中的转换也是有限的。

定理 4. 路径叠加转换序列可以在线性时间 O(n) 内求出(n 为图 G 中边的个数)。

证明: 对一个表示 FSM 的图 G, 首先寻找汇聚边, 分割子图, 可以在线性时间 O(n)内求出; 对每个子图,

根据算法 1, Step 1 存在一个针对图中所有边的循环, 时间复杂度为 O(n), Step 3 查找的最坏时间也为线性时间 O(n), Step 10 仅对所有边做一次循环, 时间也为 O(n)。所以一个路径叠加转换序列可以在线性时间 O(n)内求出。

例 1: 图 1 所示 FSM 的所有路径叠加转换序列为:

$$L_1 = {}^3 0 / 1^4; L_2 = {}^4 0 / 1^4;$$

$$L_3 = {}^1 1 / 0^4 \ 1 / 0^3 \ 1 / 0^1 \ 0 / 0^2 \ 1 / 0^2 \ 0 / 0^1$$

这里, 每对输入输出标号的左上角数字表示之前所处状态, 右上角数字表示之后所处状态。

步骤 3. 构造所有路径叠加转换序列尾状态的全部最短 UIO 序列。构造 UIO 的算法参考文献[6,9]。

步骤 4. 构造有向图 $G' = (V', E')$, 其中 $V' = V \cup U$, $E' = E \cup L \cup Q$, 其中 $U = \{u_k | v_k = tail(W), W\}$ 是路径叠加转换序列;

$L = \{l_{jk} | l_{jk} = (v_j, u_k; i_{jk} / o_{jk}), v_j = head(W), v_k = tail(W), u_k \in U, i_{jk} / o_{jk}$ 是 W 中所有转换的标号顺序连结的标号序列, W 是路径叠加转换序列};

$Q = \{q_{jk} | q_{jk} = (u_j, v_k; uio_j), u_j \in U, v_k = tail(uio_j), uio_j$ 为 v_j 的一个 UIO 序列}.

例 1 对图 1 所示的 FSM 根据步骤 1-3 求出路经叠加转换序列和 UIO 序列, 构造有向图 (图 3) $G' = (V', E')$:

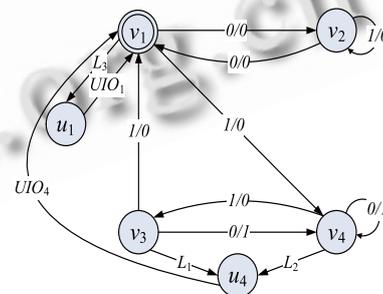


图 3 根据路径叠加转换序列构造的图 G'

步骤 5. 利用最小费用最大流算法^[10]构造 G' 的对称增广图 $G^* = (V^*, E^*)$ 。该过程可以分为以下两步:

1) 构造 G' 的一个网络流模型 $G_f = (V_f, E_f)$, 其中 $V_f = V' \cup \{s, t\}$, $E_f = E \cup Q \cup \{(s, u_k) | u \in U\} \cup \{(v_j, t) | v_j \in V\}$ 。s, t 分别代表 G_f 的源点和汇点; $E \cup Q$ 中边的容量无限; (s, u_k) 的容量为 u_k 在 G' 中的入度 (indegree)^[7]; (v_j, t) 的容量为 v_j 在 G' 中的出度 (outdegree)^[7]; E_f 中所有边的费用都为单位 1。

算法是个简单而有效的算法。

目前协议的安全性已经越来越重要,在对协议测试的过程中也需要考虑到安全性测试^[12-14],所以还需要更进一步的研究,希望能在一致性测试的基础上,能有效的对协议进行安全性测试。

表4 ISDN BRI 网络层协议测试序列(自左向右按行读)

UI_1?setup_bad/relcom	UI_1!setup/UI_1!setupack
UI_1?info/UI_1!info	UI_1?info/null
UI_1?info_last/UI_1!callproc	UI_2?netalert/UI_1!alert
NI_2?netconn/UI_1!conn	UI_1?connack/null
UI_1?connack/null	UI_1?disc/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
T302?timeout/UI_1!prog•UI_1!disc	T305?timeout/UI_1!rel
UI_1?disc/null	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info_last/UI_1!info•UI_1!setupack
UI_2?netalert/UI_1!alert	NI_2?netclear/UI_1!disc
T305?timeout/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info_bad/UI_1!prog•UI_1!disc
T305?timeout/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?disc/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
T302?timeout/UI_1!prog•UI_1!disc	T305?timeout/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
T302?timeout/UI_1!prog•UI_1!disc	UI_1?disc/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
UI_1?info/UI_1!info	T302?timeout/UI_1!prog•UI_1!disc
T305?timeout/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info/UI_1!info
UI_1?disc/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info/UI_1!info
UI_1?info_bad/UI_1!prog•UI_1!disc	T305?timeout/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
UI_1?info_last/UI_1!info•UI_1!setupack	NI_2?netclear/UI_1!disc
T305?timeout/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info_last/UI_1!info•UI_1!setupack
UI_1?disc/UI_1!rel	T308?timeout/UI_1!rel
UI_1!setup/UI_1!setupack	UI_1?info_last/UI_1!info•UI_1!setupack
UI_1?disc/UI_1!rel	UI_1?disc/UI_1!rel
T308?timeout/UI_1!rel	UI_1!setup/UI_1!setupack
UI_1?info_last/UI_1!info•UI_1!setupack	UI_2?netalert/UI_1!alert
NI_2?netconn/UI_1!conn	NI_2?netclear/UI_1!disc

参考文献

- 1 龚正虎. 计算机网络协议工程. 长沙: 国防科技大学出版社, 1993. 140-164.
- 2 Sabnani K, Dahbura A. A protocol test generation procedure. *Computer Networks*, 1988, 15(4): 285-297.
- 3 Yang B, Ural H. Protocol conformance test generation using multiple UIO sequences with overlapping. *SIGCOMM Comput Commun Rev*, 1990, 20(4): 118-125.
- 4 赵保华, 钱兰, 郭雄辉. 基于 UIO 测试序列的错误诊断算法. *电子与信息学报*, 2006, 28(11): 2152-2156.
- 5 赵保华, 陈波, 屈玉贵. 一种改进的转换可执行分析测试序列生成算法. *中国科学技术大学学报*, 2007, 37(9): 1096-1100.
- 6 徐军委, 赵保华, 屈玉贵. 一种新的 UIO 测试序列生成算法. *中国科学技术大学学报*, 2004, 34(6): 710-715.
- 7 Bondy JA, Murty USR. *Graph theory with application*. New York: Elsevier North Holland, 1976. 26-237.
- 8 Miller RE, Paul S. On the generation of minimal-length conformance tests for communication protocols. *IEEE/ACM Trans. Netw*, 1993, 1(1): 116-129.
- 9 Naik K. Efficient computation of unique input/output sequences in finite-state machines. *IEEE/ACM Trans. Netw*, 1997, 5(4): 585-599.
- 10 Cormen T, Leiserson C, Rivest R, et al. *Introduction to Algorithms*. 2nd ed. Cambridge, Massachusetts: MIT Press, 2001. 643-699.
- 11 ISDN User-Network Interface Layer 3 Specification for Basic Call Control. In, ITU-T Q931.
- 12 Lee JD, Jung JI, Lee JH, et al. Verification and conformance test generation of communication protocol for railway signaling systems. *Computer Standards & Interfaces*, 2007, 29(2): 143-151.
- 13 陈伟琳, 周颢, 赵保华. 利用构造类别代数的协议安全测试方法. *西安交通大学学报*, 2008, 42(12): 1481-1485.
- 14 章志燮, 周颢, 赵保华. 一种基于变异分析的协议安全测试方法. *西安交通大学学报*, 2009, 43(12): 11-15.