基于确定有限状态机的测试输入序列选取

张 涌 钱乐秋 王渊峰
(复旦大学计算机科学系 上海 200433)
(yongzhang@fudan.edu.cn)

摘 要 有限状态机可以精确地刻画软件系统或其子系统的行为,其在软件建模中被广泛应用. 基于有限状态机的测试问题得到了广泛的研究,其中Wp 方法由于限制条件较少,并且其可以达到较高的错误覆盖度,因此被广泛使用. 但当有限状态机的实现中的状态数目的估计值m 较大时,产生的测试输入序列数目过多,影响了其使用效率. 提出一种Wp 方法的改进方法R-Wp 方法,使用该方法在m 值较大时可以产生相对较少的测试输入序列数目,可以提高测试效率;同时本文对Wp 方法和R-Wp 方法产生的测试输入序列数目进行了讨论,并证明了该方法与Wp 方法相同的错误覆盖能力.

关键词 软件测试,规约说明,有限状态机,测试输入序列选取

中图法分类号 TP311

TEST SEQUENCES SELECTION BASED ON DETERMINISTIC FINITE-STATE MACHINES

ZHANG Yong, QIAN Le-Qiu, and WANG Yuan-Feng
(*Department of Computer Science, Fudan University, Shanghai* 200433)

Abstract The behavior of a software system or its subsystems can be described precisely using finite-state machines, so it has been used widely in software modeling. Many researchers have proposed some test selection methods based on FSM. Among these methods, the Wp-method, which has fewer use constraints and higher fault coverage, can be used more widely. But if the estimated states number (m) of the implementation of a specification is large, a large number of test sequences will be generated using the Wp-method, which will decrease the testing efficiency. In this paper, an improved method of the Wp-method, named R-Wp method, is presented. When the m is large, fewer test sequences can be obtained through this method than through the Wp-method. In addition, the number of test sequences generated from the R-Wp method and the Wp-method, and factors which influence the number of test sequences generation, are discussed. Finally, it is proved that the R-Wp has the same fault detecting ability as the Wp-method.

Key words software testing, specification, finite-state machine, test sequences selection

1 引 言

有限状态机可以精确地刻画软件系统或其子系统的行为,因此有限状态机已经被广泛应用于许多领域的应用系统建模,例如,通信协议、实时系统、面向对象软件中类的行为及其交互等等.

软件测试是软件质量保证的一种主要手段,由于软件系统的规模和复杂性的日益增加,软件测试在软件开发过程中的作用也越来越重要. 由于有限

原稿收到日期: 2001-07-05; 修改稿收到日期: 2002-07-23

状态机在建模中的广泛使用, 因此基于有限状态机的测试方法也得到了广泛的研究

许多研究人员已经提出了基于有限状态机的测试输入序列产生方法, 其中Wp 方法由于限制条件较少, 并且其可以达到较高的错误覆盖率, 因此被广泛使用. 但通过Wp 方法产生的测试输入序列数目较多, 存在一些冗余的测试输入序列, 影响了测试的效率. 特别是当对被测试的系统实现中状态数目的估计值太大, 则测试输入序列的产生数目将增长很快. 如果系统较为复杂, 该方法在实践中往往不太可行, 可能会无法保证测试工作的顺利完成

本文提出一个Wp 方法的改进方法 R-Wp (reduced Wp) 方法, 该方法可以有效地减少测试输入序列的产生数量, 特别是当构造测试输入序列时对实现中状态数目的估计值较大时, 可以大大减少测试输入序列的数量, 同时该方法可以保证具有与Wp 方法同样的错误覆盖度. 此外, 本文证明了 R-Wp 方法与Wp 方法相比减少了测试输入序列的数目, 以及具有与Wp 方法相同的错误覆盖度, 并讨论了影响测试输入序列产生数量的各种因素

2　相关研究工作

基于有限状态机的测试方法主要有 5 种:

① T 方法[1]

T 方法较为简单, 测试输入序列对应于规约说明中的状态迁移随机地产生, 直到所有的状态迁移都被覆盖. 该方法的缺点是测试输入序列中存在大量的冗余, 另外其检错能力也较差

② U 方法[2, 3].

U 方法首先为状态机的每一个状态得到一个识别序列, 该识别序列叫做单一输入输出序列(unique input/output sequence, UIO), 该识别序列可以区分每一个状态, 然后根据该识别序列构造测试输入序列. 但并不是所有的有限状态机都存在UIO, 如果状态机不存在UIO, 则无法构造测试输入序列

③ D 方法[4]

该方法首先对有限状态机构造一个区分序列(distinguishing sequence), 然后根据该区分序列构造测试输入序列. 该方法产生的测试输入序列数目较少, 但并不是每一个状态机都存在区分序列, 因此限制了该方法的使用

④ W 方法[5].

该方法基于状态机的状态识别集来构造测试输入序列, 每个状态机只要是精简的都存在状态识别集, 因此该方法的适用性较为普遍, 但该方法产生的测试输入序列数目太多, 在实际应用中使测试效率降低

⑤ Wp 方法[6].

该方法是对W 方法的简化, 同时保证了与W 方法具有相同的错误覆盖度, 但该方法产生的测试输入序列数目仍然较多.

以上这些方法使用的前提条件是有限状态机都是完备的、强连通的并且是精简的. Luo 等人[7, 8]使用泛化的Wp 方法来对非确定性有限状态机进行测试. Bernhard[9]提出了使用产生式的 3 种优化W-方法的方案, 但 Petrenko[10]说明这 3 种方法无法获得与W 方法相同的错误覆盖度. 另外, 状态识别集、状态覆盖集、迁移覆盖集等的构造直接影响到测试输入序列的长度和数量. 文献[11～14]提出了对测试输入序列长度进行优化的方法, Inan[15]等人提出了一种构造最小长度的测试输入序列的模型, 该模型基于测试输入序列和状态识别以及验证序列的特性, 给出了构造最短测试输入序列长度的产生式算法. 以上这些优化算法仅考虑了对构造方法中的参数进行优化来减少每个测试输入序列的长度, 并没有对测试输入序列的构造方法进行改进来减少测试输入序列数目.

3　本文中使用的术语定义以及Wp 方法回顾

3.1　术语及定义

有限状态机可以表示为 $M = (S, X, Y, \delta, \lambda)$, 其中, S 表示所有状态的集合, X 表示所有输入符号的集合; Y 表示所有输出符号的集合; δ 表示状态迁移函数, $\delta: S \times X \to S$; λ 表示输出函数, $\lambda: S \times X \to Y$.

我们使用 $M_i - x/y \to M_j$ 表示有限状态机 M 处于 M_i 状态时接受输入 x 使状态转移到 M_j 状态并产生输出 y.

定义 1　给定一有限状态机 A, 若 $\forall S_i, S_j \in S_A: (\exists p \in X^*: (\delta_A(S_i, p) = S_j))$, 则我们称该有限状态机为强连通的.

定义 2　给定一有限状态机 A, $\forall S_i \in S_A: (\forall x \in X: (\exists S_j \in S_A: \delta(S_i, x) = S_j))$, 则称有限状态机 S 为完备的.

定义 3　给定一完备有限状态机 A, 若 $\forall S_i, S_j \in$

S_A: $(\forall x \in X: (\lambda(S_i, x) \neq \lambda(S_j, x)))$, 则我们称该完备有限状态机为精简的

定义 4 给定一有限状态机A, 其初始状态为S_0, $\forall S_i \in S_A$: $(\exists p \in X^*: (S_0 - p \rightarrow S_i))$, 则称$S$ 中的状态是可达的

令V_1 和V_2 是两个输入序列集合, 则$V_1 \cdot V_2$ 表示两个输入序列的串联, 即$V_1 \cdot V_2 = \{v_1 \cdot v_2 \mid v_1 \in V_1, v_2 \in V_2\}$, $V^n = V \cdot V^{n-1}$; $X[K]$表示集合$\{\epsilon\} \cup X \cup X^2 \cup \cdots \cup X^K$; 令$A$ 和B 表示两个有限状态机, 在本文中用A 表示规约说明中的状态机, B 表示对A 的实现中的状态机

定义 5 给定一个输入序列集合V, 以及两个有限状态机A 和B, $\forall S_i \in S_A$: $(\exists I_k \in S_B: (\forall v \in V: \lambda(S_i, v) = \lambda(I_k, v)))$, 则我们称$S_i$, I_k 关于V-等价, 记为$S_i \approx_V I_k$

定义 6 若对于任意输入序列集合V, S_i 和I_k 都是关于V-等价的, 则我们称S_i 和I_k 等价, 记为$S_i \approx I_k$

定义 7 若两个有限状态机A 和B 的初始状态S_0 和I_0 是等价的, 则我们称这两个状态机是等价的

定义 8 给定一有限状态机A, 令Q 是一个输入序列的集合, 若$\epsilon \in Q$ 并且$\forall S_i \in S_A$: $(\exists q \in Q: (S_0 - q \rightarrow S_i))$, 则称$Q$ 是S 的状态覆盖集

定义 9 给定一有限状态机A, 令P 是一个输入序列的集合, 若$\epsilon \in P$, 并且对于任意状态迁移$S_i - x/y \rightarrow S_j$, $S_i, S_j \in S_A$, $\exists p, p \cdot x \in P$: $(S_0 - p \rightarrow S_i) \wedge (S_0 - p \cdot x \rightarrow S_j)$, 则称$P$ 为S 的状态迁移覆盖集

定义 10 给定一有限状态机A, 令W 是一个输入序列的集合, 若$\forall S_i, S_j \in S_A$: $(\exists w \in W: (\lambda(S_i, w) \neq \lambda(S_j, w)))$, 则称$W$ 是A 的特征集

定义 11 给定两个确定有限状态机A 和B, 令V-等价是从A 到B 上的函数, 若该函数是双射的, 则我们可以称其为A 到B 上的同构关系

3.2 Wp 方法回顾

Wp 方法的前提条件是规约说明和实现中的有限状态机是精简的、强连通的、完备的, 并且实现中的每个状态都存在 reset 操作且都已正确实现. 规约说明中的有限状态机是精简的, 是基于有限状态机测试的一个充分必要条件, 否则我们无法确定被测试状态是否为正确状态, 它也是特征集存在的必要条件; 强连通可以通过添加 reset 操作来实现, 当有限状态机处于任意状态时, 通过应用 reset 操作它都可以返回到初始状态, 这样使得每一个测试输入序列都从初始状态开始, 否则无法保证测试的正确性

Wp 方法可以分为 3 个步骤:

① 估计实现中的有限状态机中可能存在的状态数目的上界m. 对某个有限状态机的正确实现应该与规约说明中的状态机具有相同的状态数目, 这里的估计值m 就是为了能够检测出某个有限状态机的实现中存在的额外状态, 因此估计值m 要大于或等于规约说明中的有限状态机中的状态数目n. 估计值的选取是根据设计进行猜测得到的

② 检验在规约说明中的状态在实现中可被识别并验证其实现与规约说明中一致, 同时从初始状态到这些状态所经历的状态迁移也被验证, 并且检验实现中是否存在额外状态, 该部分的测试输入序列可如下构造:

$$T_1 = Q \cdot X[m - n] \cdot W.$$

③ 验证在上面没有被验证的状态迁移, 测试输入序列可构造如下:

$$T_2 = (P - Q) \cdot X[m - n] \otimes W = \bigcup_{p_1 \in (P - Q)} \{p_1\} \cdot \left(\bigcup_{p_2 \in X[m-n]} \{p_2\} \cdot W_j \right).$$

4 R-Wp 方法及其应用实例

4.1 R-Wp 方法

对某个有限状态机的实现中可能包含以下类型的错误:

① 状态错误;

② 状态迁移中的输出错误;

③ 状态迁移的指向错误;

④ 多余/丢失状态

我们对有限状态机设计测试输入序列的标准是: 测试输入序列的数目要尽可能的少, 并且测试输入序列要能够检测到以上的错误

有限状态机在逻辑上可以使用有向图的形式来表示, 假定规约说明中的有限状态机的图形表示A 是正确的, 若两个图A 和B 同构, 则B 中不会存在以上错误, 否则B 中可能会有一种或多种以上类型的错误. 因此, 对有限状态机的测试问题可以转化为判定两个有向图是否同构的问题, 也就是我们设计出的测试输入序列要能够检测出两个有向图是否同构, 对于测试输入序列全部正确执行的某个有限状态机的实现B, 则可确保其与规约说明A 同构; 否则, 可以判定其与A 不同构

经过研究我们发现, 在Wp 方法中产生的测试

输入序列不仅覆盖了正确的状态迁移, 并且对于实现中额外的状态上的状态迁移同样覆盖了. 如图 1 表示规约说明中的有限状态机A, 图 2 表示A 的某个实现中的有限状态机B, 其中B 中存在一个额外状态 I_3, 用Wp 方法产生的测试输入序列见第 4 3 节, 其中有些测试输入序列覆盖了 I_3 上的状态迁移. 对于测试来说, 我们要用最少的测试输入序列来找到尽可能多的错误, 额外状态上的状态迁移对于测试来说必定都是错误的, 因此我们不必对其覆盖就可确认. 我们使用Wp 方法的假定条件是规约说明和实现中的有限状态机是完备的、强连通的、精简的, 并且有相同的输入集合, 如果能够检测出某个状态是额外的, 那么就不必对它出发的状态迁移进行测试, 就可以减少测试输入序列的数量. 若A 的实现B 是精简的, 那么 $X[m-n]W$ 必定能够区分B 中的任意状态, 可见文献[5]中的引理 0. 在W 方法和Wp 方法中第 1 部分生成的测试输入序列可以完成对 I 中的每个状态进行区分和验证. 据此, 我们对Wp 方法进行改进, 使构造的测试输入序列数目减少并且保证与Wp 方法具有相同的错误覆盖能力.

本文提出的R-Wp 方法也是根据特征集来构造测试输入序列. R-Wp 方法的前提条件同Wp 方法, 本方法分为 3 步:

① 估计实现中的有限状态机中状态数目的上界m, m 要大于或等于规约说明中的有限状态机中的状态数目 n. Gill[16]已经证明了如果没有这种假设, 将不存在任何算法能够验证对某个有限状态机的实现与规约说明等价.

② 检验在规约说明中的状态在实现中可被识别并验证其正确性, 同时从初始状态到这些状态所经历的状态迁移也被验证, 该部分的测试输入序列可如下构造:

$$T_1 = Q.X[m-n]W.$$

③ 验证在上面没有被验证的状态迁移, 测试输入序列可构造如下:

$$T_2 = (P-Q)\otimes W = \bigcup_{p_1\in(P-Q)}\{p_1\}.W_j).$$

我们在 T_2 的构造过程中没有使用 $X[m-n]$ 与 $(P-Q)$ 组合来生成测试输入序列, 这样可以减少测试输入序列的数量.

4 2　实例及与Wp 方法的比较

图 1 中的A 为规约说明中的有限状态机, 它是完备的以及精简的. 图 2 中的B 为A 的实现, 我们可以看出其包含多余的状态并且有错误的状态迁移.

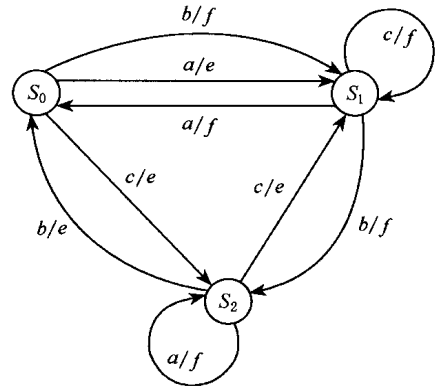

图 1　有限状态机A

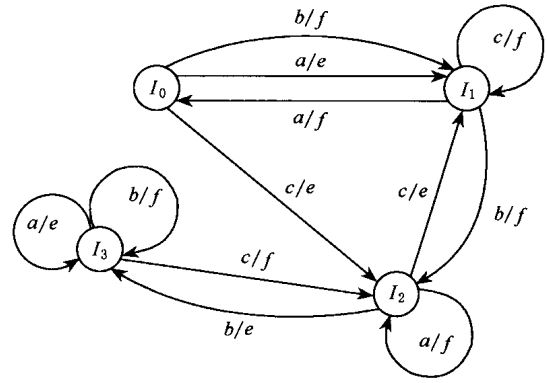

图 2　A 的一个实现B

下面我们分别使用R-Wp 方法和Wp 方法从A 中构造测试输入序列来检测B 中存在的错误. R-Wp方法对于图 1 中的S 产生的测试输入序列如下:

$T_1 = Q.W \cup Q.X.W$;

$Q.W = \{a, c, b, ba, bc, bb, ca, cc, cb\}$;

$Q.X.W = \{aa, ab, ac, ba, bb, bc, ca, cb, cc, baa, bab, bac, bba, bbb, bbc, bca, bcb, bcc, caa, cab, cac, cba, cbb, cbc, cca, ccb, ccc\}$;

$T_2 = R\otimes W = \{a\}.W_1 \cup \{bc\}.W_1 \cup \{ba\}.W_0 \cup \{bb\}.W_2 \cup \{ca\}.W_2 \cup \{cc\}.W_1 \cup \{cb\}.W_0 = \{ac, bcc, baa, bbb, cab, ccc, cba\}$.

Wp 方法对于图 1 中的S 产生的测试输入序列如下:

$T_1 = Q.W \cup Q.X.W$;

$T_2 = R\otimes W \cup R.X\otimes W$;

$R\otimes W = \{a\}.W_1 \cup \{bc\}.W_1 \cup \{ba\}.W_0 \cup \{bb\}.W_2 \cup \{ca\}.W_2 \cup \{cc\}.W_1 \cup \{cb\}.W_0 = \{ac, bcc, baa, bbb, cab, ccc, cba\}$;

$R.X\otimes W = \{aa\}.W_0 \cup \{bca\}.W_0 \cup \{baa\}.W_1 \cup \{bba\}.W_2 \cup \{caa\}.W_2 \cup \{cca\}.W_0 \cup \{cba\}.W_1$

$\cup\{a.b\}.W_2\cup\{b.c.b\}.W_2\cup\{b.a.b\}.W_1\cup\{b.b.b\}.W_0\cup\{c.a.b\}.W_0\cup\{c.c.b\}.W_2\cup\{c.b.b\}.W_1\cup\{a.c\}.W_1\cup\{b.c.c\}.W_1\cup\{b.a.c\}.W_2\cup\{b.b.c\}.W_1\cup\{c.a.c\}.W_1\cup\{c.c.c\}.W_1\cup\{c.b.c\}.W_2=\{a.a.a, b.c.a.a, b.a.a.a, b.b.a.b, c.a.a.b, c.c.a.a, c.b.a.c, a.b.b, b.c.b.b, b.a.b.c, b.b.b.a, c.a.b.a, c.c.b.b, c.b.b.c, a.c.c, b.c.c.c, b.a.c.b, b.b.c.c, c.a.c.c, c.c.c.c, c.b.c.b\}$.

首先我们构造特征集 W:

$$W_0=\{a\}, W_1=\{c\}, W_2=\{b\},$$

$$W=\{\{a\},\{c\},\{b\}\},$$

然后可以构造状态覆盖集和迁移覆盖集 Q 和 P:

$$Q=\{\epsilon, b, c\},$$

$$P=\{\epsilon, a, b, b.c, b.a, b.b, c, c.a, c.c, c.b\},$$

$$R=(P-Q)=\{a, b.c, b.a, b.b, c.a, c.c, c.b\}.$$

W, Q, P 的构造方法参见文献[5]. 在构造测试输入序列之前, 我们首先对实现中的状态数目进行估计, 这种估计需要设计人员的经验, 我们假定实现中的状态数目为 $m=n+1$(n 为规约说明中的有限状态机中的状态数目).

在 R-Wp 方法中产生的测试输入序列中, 对于测试输入序列 $c.b.c$, 期望的输出为 $e.e.e$, 然而实现 B 中对于该测试输入序列的输出为 $e.e.f$, 则可知在输入序列 $c.b$ 之后产生了错误的状态迁移, 并使状态机达到了一个错误的状态, 则我们可认定其上的所有状态迁移都是错误的, 不必对它们进一步检测. 我们可以看到在这种情况下, R-Wp 方法与 Wp 方法相比减少了测试输入序列的数目. 在第 5 节我们将对两种方法产生的测试输入序列的数目进行比较.

5 算法讨论与证明

5.1 两种方法产生的测试输入序列数量的比较

(1) Wp 方法

测试序列的数目

$$l_1=|T_1|+|T_2|=|Q|\times(1+|X|+\cdots+|X|^{m-n})\times\sum_{i=1}^{n}|W_i|+|P-Q|\times(1+|X|+\cdots+|X|^{m-n})+k_1.$$

(2) R-Wp 方法

测试序列的数目

$$l_2=|T_1|+|T_2|=|Q|\times(1+|X|+\cdots+|X|^{m-n})\times\sum_{i=1}^{n}|W_i|+|P-Q|+k_2.$$

其中符号 $|h|$ 代表某个集合的势, k_1 和 k_2 是两个变量, 它们的值在构造测试输入序列时动态地确定; 如果每个 W_i 仅包含 1 个元素时, $k_1=k_2=0$; 当存在某个 W_i, 其中的元素个数大于 1 时, k_1 和 k_2 的值分别由 $(P-Q)$ 和 $(P-Q).X[m-n]$ 所能到达 S_i 状态的次数决定, 假定 $(P-Q)$ 能够到达 S_i 状态 t 次, 那么 $(P-Q).X[m-n]$ 将至少能到达 S_i 状态 t 次, 即 $k_1\geq k_2\geq 0$. 由上面两个公式我们可以看出:

① 当 $m=n$ 时, R-Wp 方法产生的测试输入序列数目与 Wp 方法相同.

② 当 $m>n$ 时, Wp 方法产生的测试输入序列数目将大于 R-Wp 方法; 影响测试输入序列数量的因素有输入集合 X 的大小以及 m 的值, 当输入集合 X 的势较大时, Wp 方法产生的测试输入序列数目也较大, 而且随着 m 值的增大, Wp 方法在第 2 部分产生的测试输入序列数目会呈指数型增长.

5.2 错误覆盖度的证明

如前面所述, 基于有限状态机的测试问题可以转化为判定两个有限状态机是否同构的问题, 因此我们只要证明本文提出的方法产生的测试输入序列可以确保两个有限状态机是同构的, 即符合全部测试输入序列的两个有限状态机必定是同构的, 否则它们不同构. 令 $Z=X[m-n].W$.

在下面证明中的有限状态机都假定是确定的、精简的以及强连通的.

引理 1. 假定 W-等价可以把 B 中的状态至少分为 n 个类, 则 Z 将区分 B 中的每一个状态(参见文献[5]).

引理 2. 给定两个有限状态机 A 和 B, A 与 B 等价 $(A\approx B)\Leftrightarrow$ 对于某些输入序列 V, V-等价是从 A 到 B 的同构关系(参见文献[5]).

引理 3. Z-等价是从 A 到 B 上的同构关系 $\Leftrightarrow$:

(1) $\forall S_i\in S_A$: $\exists I_k\in S_B$ $(I_k\approx_Z S_i)$, 特别是 $I_0\approx_Z S_0$.

(2) 对于任意 $I_k\text{-}x/y\rightarrow I_l$, 那么在 A 中存在两个状态 S_i 和 S_j, $I_k\approx_Z S_i$, $I\approx_Z S_j$, 并且 $S_i\text{-}x/y\rightarrow S_j$.

证明 $\Rightarrow$ 部分: 由同构关系的定义可得;

$\Leftarrow$ 部分: 我们把 Z-等价看做一个函数, 因为条件(1)以及 $Z\supseteq W$, 那么对于 $\forall S_i\in S_A$, 在 B 中存在 $I_k\in S_B$, 使得 $I_k\approx_W S_i$, 因为 A 有 n 个状态, 从 W 定义可知, 它可以把 B 中的状态分为 n 类, 又从引理 1 得知, Z 将区分 B 中的每一个状态, 因此在 B 中至多存在一个状态与 A 中的某个状态 Z-等价. 因此 Z-等价是单射的, 又从条件(2)可知, Z-等价是满射的. 由同构的定义可知, Z-等价是 A 到 B 上的同构

关系　　证毕

引理 4　对于每个 $S_i, S_i \in S_A$, 在 B 中都存在 I_k, $I_k \in S_B$, $I_k \approx_Z S_i \Rightarrow \forall I_l \in S_B: \exists S_j \in S_A: (S_j \approx_W I_l)$. (参见文献[6]).

引理 5　假定引理 3 中的条件(1)成立, 那么 B 中的 I_k 与 A 中的 S_i 关于 W-等价的充分必要条件是 I_k 和 S_i 关于 W_i-等价, 即 $I_k \approx_W S_i \Leftrightarrow I_k \approx_{W_i} S_i$.

证明

$\Rightarrow$部分: 由 $I_k \approx_W S_i$ 以及 $W_i \subset W \Rightarrow I_k \approx_{W_i} S_i$;

$\Leftarrow$部分: 假设 $I_k \approx_W S_i$ 不成立, 由引理 3 中的条件(1)成立以及引理 4 可得 $\exists S_i' \in S_A$, $I_k \approx_W S_i'$, 则可得 $I_k \approx_{W_i} S_i'$, 由 $S \approx_{W_i} I_k$, 可得 $S \approx_{W_i} S_i'$, 这与 W_i 的定义相违背, 因此 $I_k \approx_W S_i$ 成立　　证毕

引理 6　对于某个 $x \in X$, 若 B 中任意 $I_k - x \rightarrow I_l$, 在 A 中存在 $S_i - x \rightarrow S_j$, 且 $I_k \approx_W S_i$, $I \approx_W S_j \Rightarrow$ 对于 $\forall x \in X$, 在 I_k 和 S_i 上应用 x, 所得到的输出相同, 即 $\lambda(I_k, x) = \lambda(S_i, x)$, 且 $\delta(I_k, x) \approx_W \delta(S_i, x)$.

证明　假设 $\exists x_1 \in X$, $\lambda(I_k, x_1) \neq \lambda(S_i, x_1)$, 或者 $\delta(I_k, x_1) \approx_W \delta(S_i, x_1)$ 不成立. 由题设可知, $\exists S_i', S_j' \in S_A: (\lambda(I_k, x_1) = \lambda(S_i', x_1) \wedge S_i' - x_1 \rightarrow S_j')$, $S_i' \neq S_i$, $S_j' \neq S_j$, 并且 $I_k \approx_W S_i'$, $I \approx_W S_j'$, 由题设以及等价关系是传递的, 可得 $S \approx_W S_i'$, $S_j \approx_W S_j'$, 这与 W 的定义相违背, 所以若题设条件成立, 则对于 $\forall x \in X$, 在 I_k 和 S_i 上应用 x, 所得到的输出相同, 即 $\lambda(I_k, x) = \lambda(S_i, x)$, 且 $\delta(I_k, x) \approx_W \delta(S_i, x)$.　　证毕

引理 7　假定引理 3 中的条件(1)成立, 那么引理 3 中的条件(2) $\Leftrightarrow$对于 B 中任意 $I_i - x/y \rightarrow I_l$, 那么在 A 中存在两个状态 S_i 和 S_j, $I_k \approx_W S_i$, $I \approx_W S_j$, 并且 $S_i - x/y \rightarrow S_j$.

证明

$\Rightarrow$部分: $W \subset Z$, 结论显然

$\Leftarrow$部分:

(1) 若 $m - n = 0$, $Z = W$, 可得 $I_k \approx_Z S_i$, $I \approx_Z S_j$ 引理 3 中的条件(2)成立;

(2) 假设 $m - n = i$ 时, 引理 3 中的条件(2)成立;

当 $m - n = i + 1$ 时, 对于某个 $x \in X$, $I_k - x/y \rightarrow I_l$, 在 A 中存在 $S_i - x/y \rightarrow S_j$, $I_k \approx_W S_i$, $I \approx_W S_j$, 根据引理 6 可得对于 $\forall x \in X$, $I_k - x/y \rightarrow I_l$, 在 A 中存在 $S_i - x/y \rightarrow S_j$, 并且 $I_k \approx_W S_i$, $I \approx_W S_j$; 对于任意 $p = p_1 x \in X^{i+1}$, $p_1 \in X^i$, $x \in X$, $I_k - p \rightarrow I_l''$, 若 B 中存在 $I_k - p_1 \rightarrow I_l'$, $I_l' - x \rightarrow I_l''$, 由于 A 是完备的且确定的, 可得在 A 中存在 $S_j' - x \rightarrow S_j''$, $S_j', S_j'' \in S_A$. 由 $m - n = i$ 时的假设, I_k 和 S_i 关于 $X[i]W$ 等价, 可知 $I_l' \approx_W S_j'$, 根据引理 6 可得 $I_l'' \approx_W S_j''$, 即对于 $\forall p \in X^{i+1}$, $I_l'' \approx_W S_j''$, 由此可得 I_k 和 S_i 关于 $X^{i+1}.W$ 等价; 又由 $m - n = i$ 时的假设可知 $S \approx_{(\{\epsilon\} \cup X \cup \cdots \cup X^i).W} I_k$, 综上可得 $S \approx_{(\{\epsilon\} \cup X \cup \cdots \cup X^{i+1}).W} I_k$, 即 $S \approx_Z I_k$ 成立; 同理可得 $S_j \approx_Z I_l$ 也成立; 由此得到引理 3 中的条件(2)成立　　证毕

根据引理 5 和引理 7 我们可以把引理 3 改写为引理 8.

引理 8　Z-等价是从 S 到 I 上的等价关系 $\Leftrightarrow$

(1) $\forall S_i \in S_A: \exists I_k \in S_B: (I_k \approx_Z S_i)$, 特别是 $I_0 \approx_Z S_0$;

(2) 对于 B 中任意的 $I_k - x/y \rightarrow I_l$, 那么在 A 中存在两个状态 S_i 和 S_j, $I_k \approx_{W_i} S_i$, $I \approx_{W_i} S_j$, 并且 $S_i - x/y \rightarrow S_j$.

引理 9　若引理 8 中的条件(1)和(2)成立 $\Leftrightarrow S$ 和 I 关于 $(Q, Z \cup R \otimes W)$ 等价.

证明

$\Rightarrow$部分: 条件(1)和(2)成立, 则 A 与 B 是同构的, 即对于任意的输入集 V, B 和 A 都是关于 V-等价的. 因此 A 和 B 关于 $(Q, Z \cup R \otimes W)$ 等价成立.

$\Leftarrow$部分: Q 是 A 的状态覆盖集, 令 $p_i \in Q$, 则从初始状态 S_0 应用输入序列 p_i 可以到达 S 中的任意状态, 即 $S_0 - p_i \rightarrow S_i$. 因为 B 是完备的并且与 A 有相同的输入集合, 所有 $I_k \in S_B$, 可以从 I_0 应用相同的输入序列 p_i 到达, 即 $I_0 - p_i \rightarrow I_k$. 因为 I_0 与 S_0 是关于 Q, Z-等价的, 可以得到 S_i 和 I_k 是关于 Z-等价的. 特别是当 p_i 取 $\epsilon \in Q$ 时, 可以得到 $I_0 \approx_Z S_0$. 即条件(1)成立;

P 是 A 的迁移覆盖集, $P = Q \cup R$, 对于任意 p_i, $p_i x \in (Q \cup R).X[m - n]$ 可以使得 $I_0 - p_i \rightarrow I_k$, $I_0 - p_i x \rightarrow I_l$; 因为 A 是完备, 则在 A 中存在 $S_0 - p_i \rightarrow S_i, S_0 - p_i \rightarrow S_j$.

① 若 $p_i \in Q.X[m - n]$, 因为 $I_0 \approx_{Q.Z} S_0$, 可得 $I_k \approx_Z S_i$, 因为 $W_i \subseteq W \subseteq Z$, 得 $I_k \approx_{W_i} S_i$, 并且 $I \approx_{W_i} S_j$;

② 若 $p_i \in R$, 因为 $I_0 \approx_{R \otimes W} S_0 \Leftrightarrow I_0 \approx_{\{P_i\}.W_i} S_0$, 可得 $I_k \approx_{W_i} S_i$, 同理可得 $I \approx_{W_i} S_j$;

③ 若 $p_i \in R.X[m - n]$, 因为 A 和 B 关于 Q, Z-等价, 因此对于 $\forall I_k \in S_B$, 存在有 $S_i \in S_A$, 使得 $I_k \approx_{Q.Z} S_i$, 由此我们可得 $I_k \approx_Z S_i$, 又 $W_i \subseteq W \subseteq Z$, 得到 $I_k \approx_{W_i} S_i$, 对于 $I_k - x \rightarrow I_l, S_i - x \rightarrow S_j$, 由于 $I_k \approx_Z S_i$, 以及 $x \in X[m - n]$, 我们可以推出 $I \approx_W S_j$, 因为 $W_i \subseteq W$, 所以 $I \approx_{W_i} S_j$; 并且由 Z-等价的定义我们得到 $\lambda(I_k, x) = \lambda(S_i, x)$.

综上所述, 所以条件(2)成立　　证毕

定理 1 A 与 B 等价($S \approx I$) $\Leftrightarrow S$ 和 I 关于(Q. $Z \cup R \otimes W$)等价.

证明 由引理 2、引理 8 和引理 9 可以得到.

由以上证明可知,由(Q. $Z \cup R \otimes W$)可以确定两个状态机是否等价,从而可以检测出第 4.1 节中所述的 4 种类型的错误,即 R-Wp 方法的检错能力与 Wp 方法相同.

6 结 论

Wp 方法由于应用限制条件较少,并且可以达到较高的错误覆盖度,因此可以被广泛使用;但若 Wp 方法对规约说明中有限状态机的某个实现中的状态数目的估计值 m 较大以及输入集合中的元素较多时,产生的测试输入序列数目较多,限制了其在实际使用时的测试效率.本文提出一种 Wp 方法的改进方法 R-Wp 方法,使用该方法在 m 值较大时可以产生相对较少的测试输入序列数目,可以提高测试效率;同时本文对 Wp 方法和 R-Wp 方法产生的测试输入序列数目进行了讨论,并证明了该方法与 Wp 方法具有相同的错误覆盖能力.

参 考 文 献

1 S Naito, M Tsunoyama. Fault detection for sequential machines by transition tours. In: Proc of IEEE Fault Tolerant Computing Conf. 1981

2 K Sabnani, A Dahbura. A protocol test generation procedure. Computer Networks ISDN System, 1988, 15: 285～297

3 A V Aho *et al*. An optimization technique for protocol conformance test sequence generation based on U I O sequence and rural Chinese postman tour. IEEE Trans on Communications, 1991, COM-39(11): 1604～1615

4 J K Ousterhout *et al*. Medusa—An experiment in distinguish operating system structure. Communications of ACM, 1980, 92～104

5 T S Chow. Testing software design modeled by finite-state machines. IEEE Trans on Software Engineering, 1978, SE-4(3): 178～187

6 S Fujiwara *et al*. Test selection based on finite state models. IEEE Trans on Softaw re Engineering, 1991, 17(6): 591～603

7 G Luo, G V Bochmann, A Petrenko. Test selection based on communicating nondeterministic finite-state machines using a generalizing Wp-method. IEEE Trans on Software Engineering, 1994, 20(2): 149～161

8 G Luo, R L Probert, H Ural. Approach to constructing software unit testing tools. Software Engineering Journal, 1995, 245～252

9 P J Bernhard. A reduced test suite for protocol conformance testing. ACM Trans on Software Engineering and Methodology, 1994, 3(3): 201～220

10 A Petrenko. Technical correspondence comments on 'A reduced test suite for protocol conformance testing'. ACM Trans on Software Engineering and Methodology, 1997, 6(3): 329～331

11 H Ural, X Wu, F Zhang. On minimizing the lengths of checking sequences. IEEE Trans on Computers, 1997, 46(1): 93～99

12 H Ural, K Zhu. Optimal length test sequence generations on using distinguishing sequences. IEEE/ACM Trans on Networking, 1993, 1(3): 358～371

13 A Rezaki, H Ural. Construction of checking sequences based on characterization sets. Computer Communications, 1995, 18(12): 911～920

14 刘积仁,都军. 基于多 U I O 序列的协议一致性测试生成. 软件学报, 1995, (增刊): 52～59

15 K Inna, H Ural. Efficient checking sequences for testing finite state machines. Information and Software Technology, 1999, 41: 799～812

16 A Gill. Introduction to the Theory of Finite-State Machines. New York: McGraw-Hill, 1962

张 涌 男,1973 年生,博士研究生,主要研究方向为软件测试

钱乐秋 男,1942 年生,教授,博士生导师,主要研究方向为软件复用技术、软件测试等

王渊峰 男,1974 年生,博士研究生,主要研究方向为构件技术、软件构件库