

自动生成测试用例的方法设计及实现^①

张 岩

(江苏自动化研究所, 连云港 222006)

摘 要: 为了实现软件测试用例的自动生成, 提高软件开发效率和软件质量, 目前很多方法都是针对有限状态机自动生成测试用例进行介绍, 而在本文中根据扩展有限状态机对象状态迁移的特点, 利用扩展有限状态机模型, 提出了扩展有限状态机自动生成测试用例的测试方法, 并应用于自动售货机 Coinbox 类软件测试中。

关键词: 扩展有限状态机; 自动; 软件测试; 测试用例; 自动售货机

Design and Realization of Automatic Generation Test Case

ZHANG Yan

(Jiangsu Automation Research Institute, Lianyungang 222006, China)

Abstract: A method of testing case generation based on state transfer is proposed. Extended Finite State Machine is selected as the testing model. Combined with regular expression, this method passes state transfer according to test demand, and gains test case. It has been applied to the software test of control system.

Key words: EFSM; automatic; software test; test case; Coinbox

软件测试是一个非常耗费资源的活动, 其中比较重要的工作就是找一组合适的测试集合进行测试, 以此达到测试目的。由于当前软件系统的不断扩大, 手工生成测试案例既不科学、也不可能, 这就需要软件测试用例自动生成技术, 实现此技术将大大改变以往靠直觉、经验生成测试用例的传统做法, 无疑将使软件测试的效率显著提高, 同时减轻人们在编写大量测试用例过程中所付出的劳动。

目前, 在基于模型自动生成软件测试用例的方法中有限状态机(FSM)的测试模型成为软件测试研究领域的热点, 它能够很好地反映被测系统中动态行为的互交信息, 容易被测试人员理解, 它的使用降低了软件测试成本, 提高了软件测试效率和自动化程度, 改进了传统测试方法技术中的很多不足。而在对变量和变量值相对复杂的部分进行建模时 FSM 还不能做到简洁高效, 因此, 引入了扩展有限状态机。

本文根据扩展有限状态机的对象状态迁移序列的这一特点, 将扩展有限状态机模型依据广度优先遍历

的方法提出, 给出其自动生成测试用例的扩展有限状态机实现的算法, 并将此方法应用于自动售货机 Coinbox 类的软件测试中。

1 展有限状态机(EFSM)

1.1 展有限状态机定义

扩展有限状态机(EFSM)^[1]是有限状态机(finite state machine, FSM)的一个扩展, 它在继承 FSM 模型自动生成测试用例特点的基础上增加了变量、操作、迁移的前置条件等。前置条件是程序执行前变量必须满足的一组约束条件, 从而使程序执行特定的路径。只有当前置条件为真时, 状态迁移才能进行, 即迁移到下一个状态, 所以前置条件的存在使得状态迁移存在一定的不确定性。如果直接用有限状态机模型的测试方法(如 T 方法、U 方法、D 方法、W 方法等四种经典的测试方法)不能使前置条件任何时候均为真, 势必会产生大量的不可执行的测试序列, 这使得测试工作无法正常进行, 大大降低测试的效率和可信度。

^① 基金项目:安徽省教育厅自然科学基金(2005KJ004ZD)

收稿时间:2011-01-08 收到修改稿时间:2011-02-16

因此, 利用 EFSM, 我们可以更加精确地刻画类的动态行为。在该模型上对类或对象进行测试, 产生的测试用例为对象状态迁移序列。

1.2 FSM 模型

EFSM 形式化地表示为一个六元组 (S, S_0, I, O, T, V) ^[2]。其中, S 为非空的状态集合; S_0 为初始状态, 任何其它状态在给定的上下文中总能随时到达初始状态, 可以把它理解为复位或重置功能; I 为非空的输入集合, 可理解为发送的消息或方法的调用; O 为非空的输出集合, 可理解为产生的动作; T 为一个非空的状态迁移集合; V 为变量的集合, 这些变量可以是方法的参数或前置条件中判定参数。

作为测试模型的 EFSM 还要求具备以下几个条件: ①EFSM 是确定的, 就是任何状态的转换是被明确定义的, 即当前状态 S_i 对于一个输入 a , 将转换到唯一的后继状态并产生唯一的输出; ②EFSM 是最小的, 即状态集合中任意的 2 个状态都是不等价的, 即 $\forall t \in T^*, t(s) \leftrightarrow t(s')$; ③EFSM 是强连通的, 即从初始状态开始通过有限次转换能到达状态机中的任何其它状态。

2 EFSM 自动产生测试用例序列

2.1 EFSM 测试覆盖准则

在测试过程中, 为了衡量测试的充分度, 需要利用相应的覆盖准则^[3]来进行测试。下面对其中几个覆盖准则进行介绍, 并针对迁移覆盖准则给出相应的算法。

(1) 点覆盖准则: 点覆盖测试准则要求测试用例集使得 EFSM 中的每一个状态至少被访问一次。

(2) 边覆盖准则(迁移覆盖准则): 边覆盖准则要求测试用例集使得 EFSM 中的每一条迁移至少被遍历一次。

2.2 自动产生测试用例算法

在此根据 EFSM 的特点, 设计了从 EFSM 自动产生测试用例序列的算法, 该算法采用了广度优先遍历的方法, 具体如下:

① 定义枚举类型 PLACE 为每个动作的两个状态--没完成、已完成

② 定义状态 STATUS 结构, 变量为 EFSM 模型中的各 I 输入状态

③ 定义 STATUS 结构中各变量的起始动作未开

始状态

④ 定义 STATUS 结构中各变量的终止动作为完成状态

⑤ 定义状态转换运算枚举类型 OPERATOR, 变量为 EFSM 模型中的 I 输入命令集, 及开始命令 OP_FIRST、结束命令 OP_LAST

⑥ 定义已经搜索过的状态堆栈 history、需要搜索的状态堆栈 active

⑦ 定义保存转换路径的结构 TURN, 由状态 STATUS 和指针构成

⑧ 对保存转换路径的结构 TURN, 由状态 STATUS 和指针构成中变量进行初始化, 即生成起始结点, 并对 history、active 指向 TURN 的初始化信息

⑨ While (active 非空)

a V 获得待搜索的当前结点信息

b If (当前结点状态不等于终止状态)

① 计算当前状态的所有可能的下一个状态

② for (OPERATOR 值等于开始命令值, to OPERATOR 值等于结束命令)

if(下一个状态有一个转换运算从当前状态转换到此状态 &&下一个状态是有效状态 &&下一个状态是否是一个新的状态, 即从未被搜索过)

a 为 TURN 添加下一个结点, 将信息保存到 TURN 结构中

b history 指向下一个结点

c else if (当前结点状态等于终止状态)

① 搜索成功, 并保存各结点信息

② 通过指针分别读取 TURN 信息, 即用例结果, 并打印

③ break; //跳出 while 循环, 结束搜索

⑩ if(active 是空)

打印“没有发现!”

3 实例分析

3.1 EFSM 构造的自动售货机的 Coinbox 类

以自动售货机的 Coinbox 类^[4]为例, 图 1 给出 Coinbox 类的 EFSM 状态模型图。

EFSM 中任一迁移 t_{ij} 对应着类中的一个方法^[5](见表 1), 以 t_{ij} 作为该方法的标示。

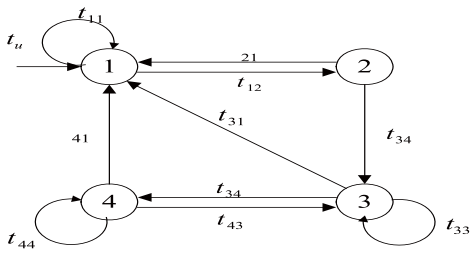


图 1 Coinbox 类的 EFSM 状态模型

表 1 迁移 t_{ij} 对应的方法

迁移	方法	迁移	方法
t_{11}	$\langle s_1, s_1, \text{ReturnQtr}() \rangle$	t_{33}	$\langle s_3, s_3, \text{AddQtr}() \rangle$
t_{12}	$\langle s_1, s_2, \text{AddQtr}() \rangle$	t_{34}	$\langle s_3, s_4, \text{ReturnQtr}() \rangle$
t_{21}	$\langle s_2, s_1, \text{ReturnQtr}() \rangle$	t_{41}	$\langle s_4, s_1, \text{VendQtr}() \rangle$
t_{23}	$\langle s_2, s_3, \text{AddQtr}() \rangle$	t_{43}	$\langle s_4, s_3, \text{AddQtr}() \rangle$
t_{31}	$\langle s_3, s_1, \text{Vend}() \rangle$	t_{44}	$\langle s_4, s_4, \text{ReturnQtr}() \rangle$

Coinbox 类的形式化模型为 $EFSM=(S,S_0,I,O,T,V)$, 其中:

状态集 $S=\{s_1, s_2, s_3, s_4\}$;

$s_1 \in S$ 是起始状态;

输入命令集合 $I=(t_u, \text{ReturnQtr}(), \text{AddQtr}(), \text{Vend}());$

输出消息集合 $O=(\text{ReturnQtr}(), \text{AddQtr}(), \text{Vend}())$

转移函数 T 为 $t_u \rightarrow s_1, s_1 \times \text{ReturnQtr}() \rightarrow s_1, s_1 \times \text{AddQtr}() \rightarrow s_2, s_2 \times \text{ReturnQtr}() \rightarrow s_1, s_2 \times \text{AddQtr}() \rightarrow s_3, s_3 \times \text{Vend}() \rightarrow s_1, s_3 \times \text{AddQtr}() \rightarrow s_3, s_3 \times \text{ReturnQtr}() \rightarrow s_4, s_4 \times \text{Vend}() \rightarrow s_1, s_4 \times \text{AddQtr}() \rightarrow s_3, s_4 \times \text{ReturnQtr}() \rightarrow s_4。$

变量集合 $V=(t_{11} / \text{ReturnQtr}(), t_{12} / \text{AddQtr}(), t_{21} / \text{ReturnQtr}(), t_{23} / \text{AddQtr}(), t_{31} / \text{AddQtr}(), t_{33} / \text{AddQtr}(), t_{34} / \text{ReturnQtr}(), t_{41} / \text{VendQtr}(), t_{43} / \text{AddQtr}(), t_{44} / \text{ReturnQtr}())。$

3.2 Coinbox 类的 EFSM 软件测试用例自动生成实现

根据 2 EFSM 产生测试用例序列的算法, 结合 Coinbox 类的 EFSM 模型, 对其进行测试用例的自动生成。在此只介绍算法中与此软件相关的内容。

enum PLACE

```

{
    NO, //每个动作的两个状态--没完成
    YES, //每个动作的两个状态--已完成
};
//状态
typedef struct
{
    PLACE Return; //ReturnQtr()
    PLACE Add; //AddQtr()
    PLACE Vend; //Vend()
}STATUS;
const STATUS c_start = { NO, NO, NO }; //起始动作未开始状态
const STATUS c_end = { YES, YES, YES }; //终止动作完成状态
//状态转换运算
enum OPERATOR
{
    START_U, // t_u
    RETURN_END, //ReturnQtr()
    ADD_COIN, //AddQtr()
    VEND_GOODS, //Vend()
    OP_FIRST = START_U, //初始命令
    OP_LAST = RETURN_END, //结束命令
};
  
```

3.3 应用算法的测试情况

对于其自动售货机的 Coinbox 类软件中的每一个动态转换行为的测试用例, 测试数据可从以下几个方面生成: ①边界值; ②邻近左边界点无效值; ③邻近左边界点有效值; ④随机的有效值; ⑤邻近右边界点有效值; ⑥邻近右边界点无效值; ⑦随机的无效值。生成的测试用例能够满足迁移准则。通过此方法对自动售货机的 Coinbox 类进行测试, 能够自动生成其测试用例, 并覆盖全部路径, 发现错误率能够达到 80%。经过试验发现, 此方法所采用的测试用例能够有效地发现自动售货机的 Coinbox 类的动态转换行为中的错误。

4 结语

本文提出了一种基于 EFSM 模型的状态迁移生成测试用例的测试方法, 并将此方法应用于自动售货机

(下转第 126 页)

储。运行 Windows 的上位机通过 CGI 技术以网页的形式从开发板 B 上读出比较结果。

结果分析:图 7 中(a)图是当 Windows 机从开发板 B 读出的比较结果。横轴为时间轴,单位为秒,纵轴为接受数据量,单位为千字节。根据显示的曲线结果,可以看出通过网络传输的趋势曲线与通过串口传输的趋势曲线是一致的。假设串口传递的数据是实时的且没有数据包丢失,而通过协议栈网络传递的数据是非实时的或在传递过程中发生数据包的丢失,则产生的曲线不会吻合,会产生时移。实验结果所显示的两条曲线完全的一致表明协议栈可以实时的把数据传输到网络上且没有数据包丢失。当设置串口在传输数据

前先缓冲数据 20ms,然后再发送数据时,可以得到图 (b) 比较结果,从图中可以看出网络传输实时性高于串口传输,但两条曲线走势完全相同,说明传输过程没有丢失数据包。

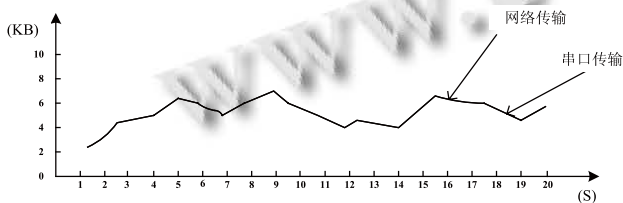
5 结语

本协议栈是在 μ clinux 系统内嵌的 TCP/IP 协议栈各功能模块基础上进行设计的。网络适配器用的是 DM9000E 芯片,设计过程中,对 dm9000.c 驱动程序中的接收包中断处理函数进行了修改,以满足协议栈性能需要。考虑到 μ clinux 对 C 支持性比较好,对设计的模块全部用标准 C 进行开发。最后通过设计的实验平台对 EPA 通信协议栈性能进行测试,测试结果显示该 EPA 通信协议栈通信过程稳定,速度快。

参考文献

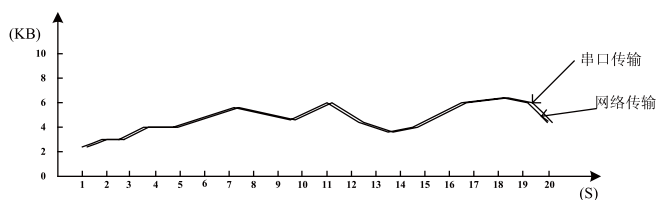
- 1 自动化在线网站.[2010-12-16]. <http://www.autooo.net/autooo/wuxiantongxun/Nesw/2008-01-16/47899.html>
- 2 陈磊,冯冬芹,金建祥,褚健.以太网在工业应用中的实时特性研究.浙江大学学报:工学版,2004,38(6):670-675.
- 3 国家质量技术监督局.GB/T20171-2006.用于工业测量与控制系统的 EPA 系统结构与通信规范.北京:中国标准出版社,2006.
- 4 张鑫龙,王平,孙攀.EPA 通信协议栈的设计与实现.重庆邮电学院学报:自然科学版,2006,18(4):512-515.
- 5 王兰,梁杰申.基于 EPA 的实时通信方案.机电工程技术,2008,37(5):49-50.
- 6 温现杰.基于 Linux 内核的实时调度机制的研究和实现.哈尔滨:哈尔滨理工大学,2009.

网络传输和串口传输数据趋势曲线



(a)

网络传输和串口传输数据趋势曲线



(b)

图 7 实验结果

(上接第 240 页)

的 Coinbox 类软件测试中。该方法证实了可以很好地描述自动售货机的 Coinbox 类的动态转换行为,生成的测试用例可以发现动态行为转换中的错误,便于测试用例生成和测试实现。但该方法仍存在很多不足,它不能对自动售货机的 Coinbox 类组合中存在的一些状态切换等问题进行很好的测试,生成的测试用例也还不能很好地优化。

参考文献

- 1 Dssouli R, Saleh K, Aboulhamid E, et al. Test Development for Communication Protocols: Towards Automation. Computer

Networks, 1999,31(17):1835-1872.

- 2 年晓玲.基于扩展有限状态机软件测试用例自动生成的研究.西南交通大学,2005.24-44.
- 3 Ammann P, Offutt J. Introduction To Software Testing. New York:Cambridge University Press, 2008.
- 4 Kung D, Suchak N, Hsia P, et al.On Object State Testing. Proc of COMPSAC'94. 1994.222-227.
- 5 卢炎生,雷波,谢晓东,毛澄映.基于扩展有限状态机的定义清除路径的生成方法.计算机工程与科学,2006,28(7):67-69.