

基于 Petri 网的软件测试分析

方贤文 赵 艳 殷志祥

(安徽理工大学,安徽 淮南 232001)

摘 要 软件测试是发现软件中错误而检查文档、运行程序的一个过程。对软件文档和代码进行测试,是贯穿于整个开发生命周期的,尤其在开发早期,其作用更为重要。Petri 网是一种好的分析工具,它能够动态运行程序模型,分析模型中存在的死锁、并发、冲突等问题,对程序正确性进行验证和分析。

关键词 Petri 网 软件测试 C 程序 Petri 网模型

中图分类号:TP301.4

文献标识码:A

文章编号:1673-629X(2007)02-0096-03

The Software Test Analysis Based on Petri Net

FANG Xian-wen ZHAO Yan YIN Zhi-xiang

(Anhui University of Science and Technology, Huainan 232001, China)

Abstract The software test is a process to inspect the documents and to run program for discovering the mistake in the software. Carrying on the test to the software documents and the code is to pass through the entire development life cycle, it is especially more important in the development early time. The Petri net is one kind of good analysis tool, it can operate dynamically the program model, and analyze model whether exist the deadlock, concurrent, conflict and so on or not, and carry on the program validation and analysis.

Key words Petri net software test C program Petri net model

0 引 言

软件测试就是在软件投入运行前,对软件需求分析、设计规格说明和编码的最终复查,是软件质量保证的关键步骤。它是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例(即输入数据及其预期的输出结果),并利用这些测试用例去运行程序,以发现程序错误的过程。软件测试贯穿于软件定义和开发的整个期间^[1,2]。因此,需求分析、概要设计、详细设计,以及程序编码等各个阶段所得到的文档,包括需求规格说明、概要设计规格说明、详细设计规格说明以及源程序,都是软件测试的对象。目前,软件测试主要采用两类测试方法,即白盒测试和黑盒测试。关于软件测试的新方法,已有相当多的研究成果,在文献[3]中,提出了软件测试过程的 POCERM 模型。文献[4]中,提出了软件测试过程框架 FDIWM。文献[5]中,提出了通过测试和失败测试两种基本软件测试

方法。还有一些文献利用程序流程图或数据流图对软件进行测试,但分析时存在着数据流动不清晰的问题,也不容易找到程序中的错误存在原因及地点。文中采用基于 Petri 网的方法对软件测试进行分析,能够容易验证程序的正确性,并易于确定出现问题的地点。

1 Petri 网的基本知识

这里仅给出几个与此文密切相关的概念,其它 Petri 网术语可参见文献[6,7]。

定义 1^[6,7] 满足下列条件的三元组 $N = (P, T; F)$ 称为一个网。

- 1) $P \cup T \neq \emptyset$
- 2) $P \cap T = \emptyset$
- 3) $F \subseteq ((P \times T) \cup (T \times P))$
- 4) $\text{don}(F) \cup \text{cod}(F) = P \cup T$

其中:

$\text{don}(F) = \{x \in P \cup T \mid \exists y \in P \cup T (x, y) \in F\}$

$\text{cod}(F) = \{x \in P \cup T \mid \exists y \in P \cup T (y, x) \in F\}$

P 和 T 称为网 N 的基本元素集, P 的元素称为 P -元或库所, T 的元素称为 T -元或变迁, F 是网 N 的流关系。

定义 2^[6,7] 设 $N = (P, T; F)$ 为一个网,对 $x \in$

收稿日期:2006-04-26

基金项目:国家自然科学基金(30570431);安徽省优秀青年基金(06042088);安徽理工大学青年基金(2004020)及硕博基金项目

作者简介:方贤文(1975-),男,河南信阳人,讲师,硕士,研究方向为并行算法、Petri 网;殷志祥,教授,博士,主要研究方向为 DNA 计算、智能计算。

$P \cup T$ 记

$${}^*x = \{y \mid y \in P \cup T \wedge (y, x) \in F\}$$

$$x^* = \{y \mid y \in P \cup T \wedge (x, y) \in F\}$$

称 x 为 x 的前集或输入集, x^* 为 x 的后集或输出集。

定义 3^[6,7] 六元组 $\Sigma = (P, T, F, K, M)$ 称为一个库所 / 变迁系统, 其中

1) (P, T, F) 是一个网;

$K: P \rightarrow N_0$ (N_0 是自然数集合) 称为容量函数;

$W: F \rightarrow N_0 - \{0\}$ 称为权函数;

$M: P \rightarrow N_0$ 是 Σ 的一个标识, 满足条件:

$$\forall p \in P: M(p) \leq K(p)$$

2) Σ 满足下面变迁发生规则:

a. 对 $t \in T, M \llbracket t \rrbracket > 0$ (表示变迁 t 可发生) 的条件为:

$$\forall p \in {}^*t: M(p) \geq W(p, t)$$

$$\forall p \in t^* - {}^*t: M(p) + W(t, p) \leq K(p)$$

$$\forall p \in t^* \cap {}^*t: M(p) + W(t, p) - W(p, t) \leq K(p)$$

b. 若 $M \llbracket t \rrbracket > M'$ 则对 $\forall p \in P$:

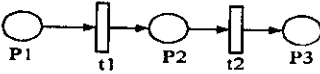
$$M'(p) = \begin{cases} M(p) - W(p, t) & \text{当 } p \in {}^*t - t^* \\ M(p) + W(t, p) & \text{当 } p \in t^* - {}^*t \\ M(p) - W(p, t) + W(t, p) & \text{当 } p \in {}^*t \cap t^* \\ M(p) & \text{当 } p \notin {}^*t \cup t^* \end{cases}$$

2 程序的基本结构及其 Petri 网模型

(1) 顺序语句及其 Petri 网模型:

Sentence 1 ;

Sentence 2 ;

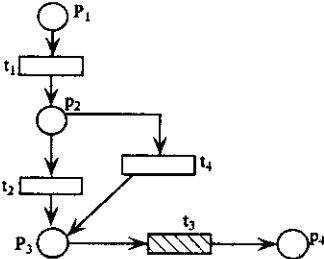


(2) 条件语句 1 及其 Petri 网模型:

if (exp1)

sentence 1 ;

sentence - following ;



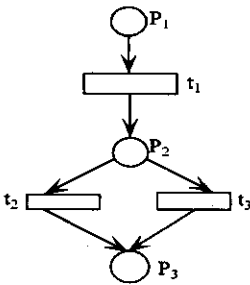
(3) 条件语句 2 及其 Petri 网模型:

if (exp1)

sentence 1 ;

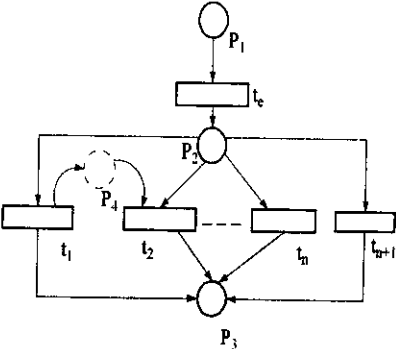
else

sentence 2 ;



(4) switch 语句及其 Petri 网模型^[8]:

```
switch (exp) {
    case (const 1):
        sentence 1 ;
    case (const 2):
        sentence 2 ;
    .....
    case (const n):
        sentence n ;
    default: sentence n + 1 ;
}
```

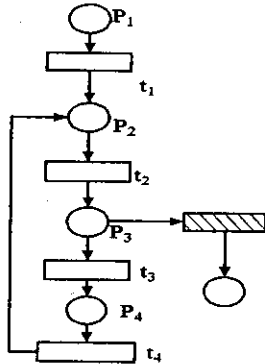


(5) for 语句及其 Petri 网模型:

for (exp1; exp2; exp3)

sentences ;

sentences - following ;

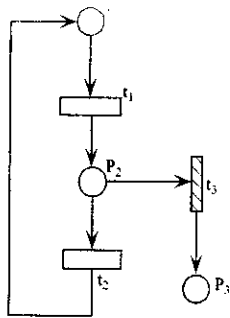


(6) While 语句及其 Petri 网模型:

While (exp)

Sentences ;

Sentences - following ;



3 基于 Petri 网的测试分析举例

下面用基于 Petri 网的软件测试分析方法来分析求 100 之内的素数的程序 (用筛法实现)。

程序代码如下：

```
#include <math.h>
main()
{ int i, j, n;
  for(i=1; i<=100; i++)
  { i]=i;
    for(j=2; j<=sqrt(100); j++)
    for(k=i+1; k<=100; k++)
    {if(i!=0 && k!=0)
      if(k% i==0)
        k]=0;
      print(" \n");
    for(i=2; n=0; i<=100; i++)
    { if(i!=0)
      { print(" %5d \n", i);
        n++;
      }
    if(n==10)
    {print(" \n");
      n=0;
    }
  }
}
```

在该 C 语言程序对应的 Petri 网模型(见图 1)中, 用软件测试的路径测试方法, 可以利用可达标识集和变迁的引发序列来表示每条路径是否被测试到, 最终完成整个软件的测试。由标识(token)所引发的包含库所和变迁的序列(即路径)共有 50 条, 经 Petri 网验证, 所有的序列都是合法序列, 程序正确性得到验证。

现选取其中的 6 条, 引发序列如下：

- (1) p1 t1 p2 t2 p3 t5 p5 t6 p6 t16 p12 t17 p13 t26 p19# # #
- (2) t18 p14 t26 p19# # #
- (3) p1 t1 p2 t2 p3 t5 p5 t6 p6 t16 p12 t17 p13 t25 p13 t18 p14 t26 p19# # #
- (4) t18 p14 t19 p15 t21 p16 t22 p17 t24 p18 t25 p13 t18 p14 t26 p19# # #
- (5) p1 t1 p2 t2 p3 t5 p5 t6 p6 t16 p12 t17 p13

- (6) t18 p14 t19 p15 t21 p16 t22 p17 t23 p18 t25 p13 t18 p14 t26 p19# # #
- (7) p1 t1 p2 t2 p3 t3 p4 t4 p2 t2 p3 t5 p5 t6 p6 t7 p7 t8 p8 t9 p5 t6 p6 t16 p12 t17 p13 t18 p14 t19 p15 t21 p16 t22 p17 t24 p18 t25 p13 t18 p14 t26 p19# # #
- (8) p1 t1 p2 t2 p3 t3 p4 t4 p2 t2 p3 t5 p5 t6 p6 t7 p7 t8 p8 t9 p5 t6 p6 t16 p12 t17 p13 t18 p14 t19 p15 t21 p16 t22 p17 t23 p18 t25 p13 t18 p14 t26 p19# # #
- (9) p1 t1 p2 t2 p3 t3 p4 t4 p2 t2 p3 t5 p5 t6 p6 t7 p7 t8 p8 t9 p5 t6 p6 t16 p12 t17 p13 t18 p14 t19 p15 t20 p16 t22 p17 t24 p18 t25 p13 t18 p14 t26 p19# # #

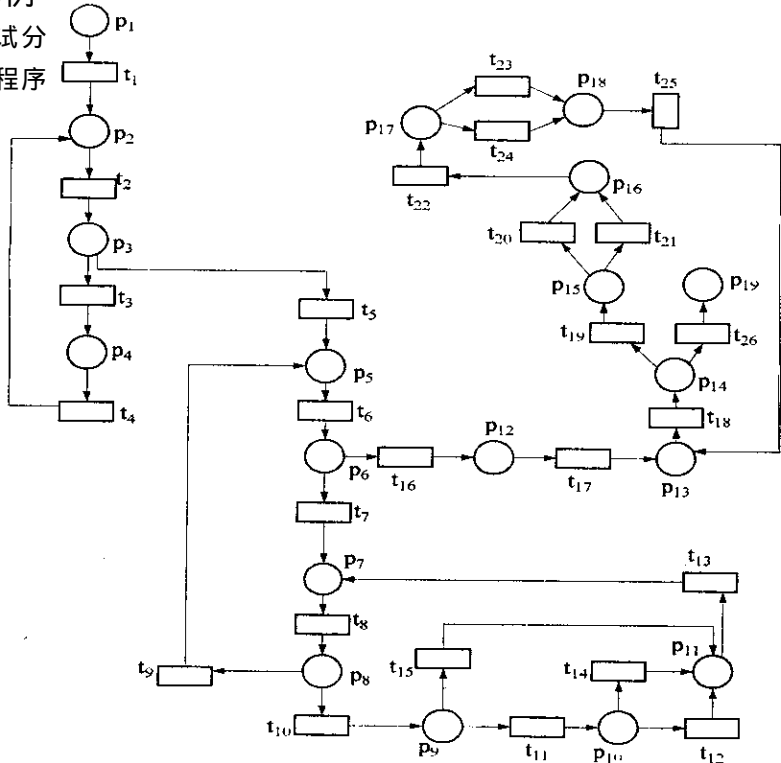


图 1 程序对应的 Petri 网模型

4 结束语

可以看出, 在软件测试中利用基于 Petri 网的方法, 首先将程序代码转变为 Petri 网模型, 然后利用 Petri 网的理论知识, 将程序中的可能出现的运行路径通过 Petri 网运行而实现。与传统中利用流程图或流程图的方法进行软件测试相比, 利用基于 Petri 网的软件测试方法, 能够动态运行程序模型, 可以看到其动态的流程走向, 方便了解其测试的进展和结果, 并分析模型中存在的死锁、并发、冲突等问题, 对程序正确性进行验证和分析。基于 Petri 网的软件测试方法在软件测试中有很大的优越性。

参考文献：

[1] 朱利群. 软件测试模型及其实现[J]. 湖南师范大学学报：(下转第 102 页)

Neuron 芯片由固件自动执行令牌协议,以防止总线冲突。在任何给定的时间,只有一方获得令牌。若主处理器获得令牌,则传送数据或者令牌到从处理器;若 Neuron 芯片拥有令牌,则将准备好的数据传送给主处理器或者交出令牌。主处理器的程序流程图如图 5 所示。

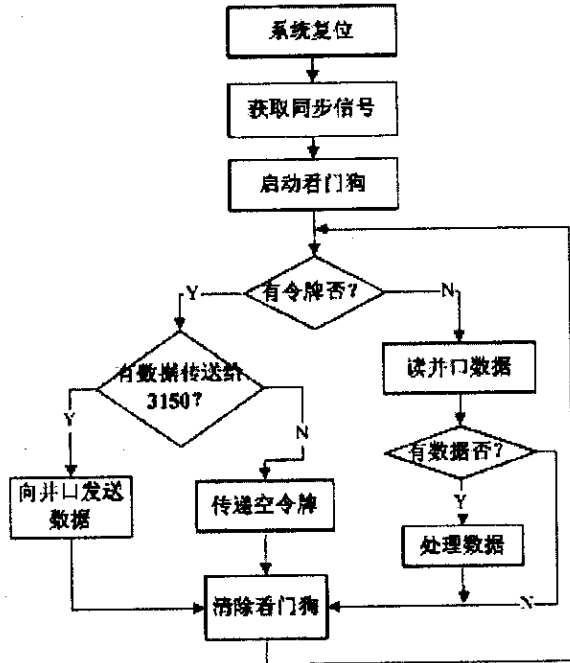


图 5 AT89S52 程序流程图

3 结束语

将 Lonworks 技术引入路灯控制系统,其完全的开放性及互操作性避免了控制级网络的不必要的更新。用户可以从一个本地网络、VPN 网络或者 Internet 访问 Lonworks 的智能路灯系统中的任何一个对象,并且可以通过鉴别机制仅允许授权的用户访问网络,使大范围的路灯的集中控制成为可能,可以节约人力资源,提高管理效率。同时,与现场环境相适应的调光控制,可以有力地减少光污染,节约电力资源,延长灯具使用寿命。

文中提出的系统结构也可以应用于远程抄表、智能家居设备控制、交通控制等领域。

参考文献:

- [1] 杨育红. LON 网络控制技术及应用[M]. 西安:西安电子科技大学出版社, 1999.
- [2] 王际业, 林锦国, 张光明, 等. 基于 Lonworks 技术的远程灯光集中控制系统[J]. 现代电子技术, 2004(3): 1-2.
- [3] Echelon 公司技术手册. i. LON100 用户指南[M]. [s.l.]: [s.n.], 2004.
- [4] Motorola 公司技术手册. Motorola Lonworks Technology Device Data[M]. [s.l.]: [s.n.], 1995.
- [5] 李根旺, 赵富海. 信号在电力线上传输应用中特性的研究[J]. 现代电子技术, 2005(9): 30-32.

(上接第 95 页)

决的方法是类似的。Petri 网提供了一种以图形和数学为基础的形式化建模的方法,采用抽象和求精的机制,很好地描述了企业铁路的作业系统,为下一步研究企业铁路智能调度的算法提供了有用的数学模型。

参考文献:

- [1] Jensen K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use[M]. Second edition. Berlin: Springer, 1996: 85-89.
- [2] 林 闯. 随机 Petri 网和系统性能评价[M]. 北京:清华大学出版社, 2000: 1-13.
- [3] 刘皓玮. 行车指挥系统的 Petri 网建模与列车运行调整的

遗传优化的研究[D]. 北京:铁道部科学研究院, 2000.

- [4] Jansen L, Meyer zu Horste M, Schnieder E. Technical Issues in Modeling the European Train Control System Using Colored Petri Nets and the Design/CPN Tools[C]//Proceedings of the Workshop on Practical Use of Colored Petri Nets and Design/CPN. Aarhus [s.n.], 1998.
- [5] Merlin P, Faber D J. Recoverability of communication protocol[J]. IEEE Transactions on Communication, 1976, 24: 1036-1043.
- [6] van der Aalst W M P. Interval Timed Petri Nets and their analysis[D]. Computing Science Notes 91/09. Eindhoven: Eindhoven University of Technology, 1991.

(上接第 98 页)

- [1] 自然科学版, 1998, 21(9): 88-91.
- [2] Patton R. 软件测试[M]. 周予滨, 姚 静译. 北京:机械工业出版社, 2002.
- [3] 刘 超, 金茂忠. 软件测试过程的基本模型 POCERM[J]. 北京航空航天大学学报, 1997, 23(1): 56-60.
- [4] 刘 群, 洪 帆. 软件测试的应用研究与分析[J]. 华中理工大学学报, 2000(11): 34-36.

- [5] 张南平, 陈小倩. 软件测试技术[J]. 微机发展, 2005, 15(7): 69-72.
- [6] Teterson J L. Petri 网理论与系统模拟[M]. 吴哲辉译. 徐州:中国矿业大学出版社, 1989.
- [7] 袁崇义. Petri 网原理[M]. 北京:电子工业出版社, 1998.
- [8] 崔焕庆. 基于 Petri 网的 MPI 并行程序建模与正确性验证[D]. 泰安:山东科技大学, 2004.