

自恢复容错系统的模型和分析^{*}

郭成昊 刘凤玉

(南京理工大学计算机科学与技术系 南京 210094)

摘要 容错系统不仅会产生硬件故障,而且连续长时间的运行,系统的性能还会逐渐下降或失效,即老化现象。本文综合考虑容错系统中的硬件故障和老化现象,提出了将传统的冗余策略和软件抗衰策略相结合,并给出了该系统的时间颜色 Petri 网(timed-CPN)模型,并结合实例进行了分析。

关键词 容错系统,软件抗衰,软件老化,冗余策略,时间颜色 Petri 网

The Model of Fault-Tolerant System with Rejuvenation

GUO Cheng-Hao LIU Feng-Yu

(Department of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094)

Abstract A fault-tolerant system experiences a crash due to hardware components' faults or progressive performance degradation as an "aging" phenomenon, because of running continuously for very long periods. This paper considers hardware components' faults and "aging" phenomenon, and proposes composing redundant structure and rejuvenation schedule in the Fault-Tolerant system. Finally formalizes the system with timed-colored Petri Nets(timed-CPN), and evaluates quantitatively the performance of the model based on numerical examples.

Keywords Fault-tolerant system, Software rejuvenation, Software aging, Redundant strategy

1 引言

在容错系统中,人们一般只注意到系统硬件部分的错误,且硬件可靠性研究已有成熟的模型和分析方法,而大量实验表明由于异常的软件行为导致系统性能崩溃的可能性要远远高于硬件故障对系统产生的影响。这就需要在容错系统中考虑软件的可靠性和可用性。因此,软件可靠性的研究已经成为制约容错系统可靠性的一个重要问题。

一般来说,投入运行的软件系统由于设计阶段考虑得不周全以及测试时间较短往往会存在一些缺陷,当这些应用软件运行一段时间以后,其中存在的缺陷或是客户的不当操作会使系统性能下降,如响应时间或负载的增加。在性能持续衰退期间,如果不采取适当的干预,最终会导致整个系统的崩溃。这种现象叫做软件衰老(software aging),它已在许多目前广泛使用的系统中出现,如 Netscape、xrn、Patriot 系统^[1,2]。

Huang 等提出了自愈(rejuvenation)技术来处理软件衰老对系统造成的影响^[3]。该方法是一种预防性的软件容错策略,它包括周期性停止系统运行,清理内部状态,重新启动。研究软件衰老的方法主要有基于检测和基于模型的方法。基于检测的方法是通过系统运行时性能参数的样本值进行统计分析来预报软件衰老的时间和其对系统造成的影响^[4]。基于模型的方法主要是通过分析系统的状态来评价自愈的性能^[5]。

本文首先分析了出现衰老现象的容错系统状态,接着利用颜色 Petri 网(CPN)来对系统进行建模,并通过马尔可夫再

生过程分析该模型,定量地评价系统性能。

2 自愈系统状态分析

Huang 等^[3]提出了一个基于模型的方法来分析自愈系统的性能。模型如图 1,图中箭头表示自愈系统的各个状态之间转变的方向,圆圈表示自愈系统的四个状态:

状态 1:系统正常状态;

状态 2:系统可能出错状态;

状态 3:系统出错状态;

状态 4:系统自愈状态。

系统在开始运行时处于状态 1,运行一段时间以后,系统进入状态 2,该状态表明系统仍然能够提供服务,但出现错误的概率不为零。在进入状态 2 后,如果系统进入状态 3,软件失效,花费一定的时间和成本修复后,软件重新回到状态 1。否则进入状态 4,执行自愈操作后,系统将回到状态 1。自愈系统的状态变化周而复始进行下去。

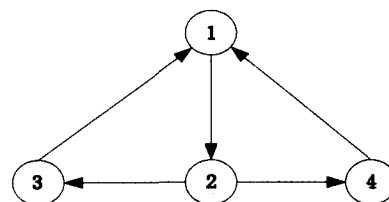


图 1 自愈系统的状态图

3 出现衰老现象容错系统状态分析

容错系统通过冗余的容错策略来处理由于硬件问题造成

^{*} 基金项目:国家自然科学基金(60273035);国防科工委基础应用项目(K1704060511)。郭成昊 博士研究生,主要研究领域为软件自愈与抗衰,体系结构;刘凤玉 教授,博导,研究方向:计算系统性能保持,自主计算,人工智能和信息安全。

整个系统的崩溃。系统相应的容错行为如图 2, 图中假设在系统中, 包含模块 n 、1、2, 其中模块 n 和模块 1 是具有特定功能的系统组件, 模块 2 是模块 1 的复本, 具有和模块 1 相同的结构和功能。当系统正常运行时, 模块 n 和模块 1 合作完成系统下达的任务, 当系统中模块 1 由于硬件问题出现故障时, 则模块 2 替代模块 1 提供同样的服务, 从而避免整个系统崩溃。

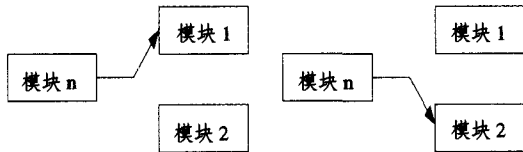


图 2 系统的容错行为

在容错系统中导致系统崩溃的原因除了硬件问题还包括软件问题, 在硬件出错之前软件的问题也可能使容错系统的性能衰退甚至使整个系统失效。系统的状态变化如图 3。图中箭头表示自愈系统的各个状态之间转变的方向, 圆圈表示容错系统的五个状态:

- 状态 1: 系统正常状态;
- 状态 2: 系统可能出错状态;
- 状态 3: 系统出错状态;
- 状态 4: 系统自愈状态;
- 状态 5: 系统硬件故障状态。

在系统进入状态 2 后, 如果系统进入状态 5, 则马上执行冗余容错策略, 回到状态 1。否则系统的状态变化如前所述。

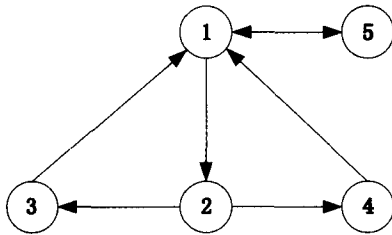


图 3 容错系统的状态图

4 容错系统时间颜色 Petri 网建模和分析

4.1 时间颜色 Petri 网 (timed-CPN)

定义 1 一个 timed-CPN 具有如下定义:

$\Sigma = (P, T, F, \lambda, C, I_-, I_+, M_0)$, 其中:

1) (P, T, F) 是一个网, P 元素是位置, T 元素是变迁, $F \subseteq (P \times T) \cup (T \times P)$, 弧的集合。

2) $C: P \cup T \rightarrow$ 颜色集的幂集合, 使得: 对 $p \in P, C(p)$ 是 P 上所有可能的标记原子颜色集合; 对 $t \in T, C(t)$ 是 T 上所有可能的出现色集合。

3) I_-, I_+ 分别是 $P \times T$ 上的负函数和正函数, 使得对所有的 $(p, t) \in P \times T$:

$I_-(p, t) \in [c(t)_{MS} \rightarrow c(p)_{MS}]_I$, 且当 $(p, t) \notin F$ 时, $I_-(p, t) = 0$;

$I_+(p, t) \in [c(t)_{MS} \rightarrow c(p)_{MS}]_I$, 且当 $(p, t) \notin F$ 时, $I_+(p, t) = 0$

4) M_0 称为 timed-CPN 的初始标识, 满足: $\forall p \in P: M_0(p) \in C(p)_{MS}$, 即 $M_0(p)$ 是 p 的标记原子颜色集合上的多重集。

5) λ 表示颜色集中的时间元素, $p \in P, \lambda \in C(p), t \in T, \lambda \in C(t)$ 。

4.2 容错系统时间颜色 Petri 网模型

容错系统的 timed-CPN 模型如图 4。

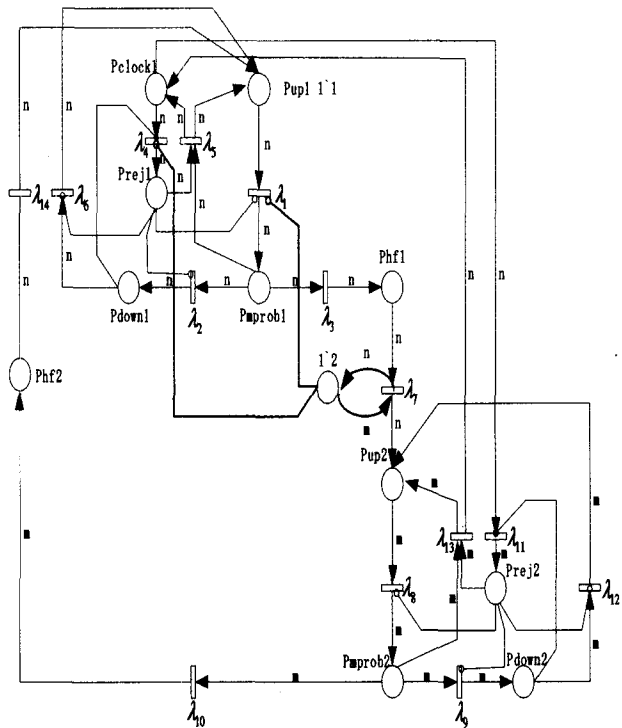


图 4 容错系统的 timed-CPN 模型

图 4 中, 位置 Pup1 表示模块 1 处于正常工作状态, 位置 Pdown1 表示模块 1 处于失效状态。初始时刻模块 1 处于 Pup1 状态, 当变迁 λ_1 触发, 模块 1 进入 Pmprob1 状态, 表示模块 1 处于可能出错状态。当变迁 λ_2 触发, 模块 1 进入 Pdown1 状态, 表示模块 1 失效, 这时变迁 λ_6 触发, 模块重新回到 Pup1。同时变迁 λ_4 禁止, 表示当模块重新回到 Pup1 时, 不执行自愈操作。模块 1 处于可能出错状态时, 如果执行自愈操作, 则变迁 λ_4 触发, 模块到达 Prej1 状态, 随后变迁 λ_5 触发, 模块重新回到 Pup1。同时变迁 $\lambda_1, \lambda_2, \lambda_6$ 禁止, 表示当执行自愈操作时, 模块 1 中其它活动全部停止; 当变迁 λ_3 触发, 模块 1 进入 Phf1 状态, 表示模块 1 出现硬件故障, 这时变迁 λ_7 触发, 模块 2 替代模块 1 完成相应的任务, 这时变迁 λ_1, λ_4 禁止, Pclock1 位置和 Pup1 位置中标记不发生转移, 位置 Pup2 表示模块 2 处于正常工作状态, 位置 Pdown2 表示模块 2 处于失效状态。初始时刻模块 2 处于 Pup2 状态, 当变迁 λ_8 触发, 模块 2 进入 Pmprob2 状态, 表示模块 2 处于可能出错状态。当变迁 λ_9 触发, 模块 2 进入 Pdown2 状态, 表示模块 2 失效, 这时变迁 λ_{12} 触发, 模块重新回到 Pup2。同时变迁 λ_{11} 禁止, 表示当模块重新回到 Pup2 时, 不执行自愈操作。模块 2 处于可能出错状态时, 如果执行自愈操作, 则变迁 λ_{11} 触发, 模块到达 Prej2 状态, 随后变迁 λ_{13} 触发, 模块重新回到 Pup2。同时变迁 $\lambda_8, \lambda_9, \lambda_{12}$ 禁止, 表示当执行自愈操作时, 模块 2 中其它活动全部停止。当模块 2 也出现硬件故障时, 变迁 λ_{10} 触发, 模块到达 Phf2 状态, 表示整个容错系统完全失效, 变迁 λ_{14} 触发, 初始化容错系统, 重新回到正常状态。

4.3 容错系统的时间颜色 Petri 网模型分析

一个具有有界标记颜色集合的 timed-CPN 系统与一个

随机 Petri 网系统有等价关系^[6]。

表 1 变迁平均实施速率参数选择

变迁	变迁平均 实施速率	数值
λ_1	x_1	1/240
λ_2	x_2	1/2100
λ_3	x_3	1/5000
λ_5	x_5	6
λ_6	x_6	2
λ_7	x_7	5
λ_8	x_8	1/240
λ_9	x_9	1/2100
λ_{10}	x_{10}	1/5000
λ_{12}	x_{12}	2
λ_{13}	x_{13}	6
λ_{14}	x_{14}	2

在 timed-CPN 模型中采用以下的简化规则得到相应的随机 Petri 网。

1. 简化颜色标记为一个不带任何信息的普通标记。
2. 不考虑模型中弧的变量。
3. 不考虑模型中变迁的颜色函数。

使用以上简化规则,从而得到和 timed-CPN 模型相对应的随机 Petri 网模型,这样就可以使用随机 Petri 网的分析方法来分析容错系统的性能。一个 11 元组 $E=(p_{up1}, p_{mprobl}, p_{hf1}, p_{down1}, p_{up2}, p_{mprobl2}, p_{hf2}, p_{down2}, p_{clock1}, p_{rej1}, p_{rej2})$, 其中位置元素的取值为 0 或 1, 如果一个标记在位置 p_i 中, $p_i=1$, 否则为 0。从系统的 timed-CPN 模型中, 得出系统具有 10 种可能的变化状态。状态变迁图如图 5。

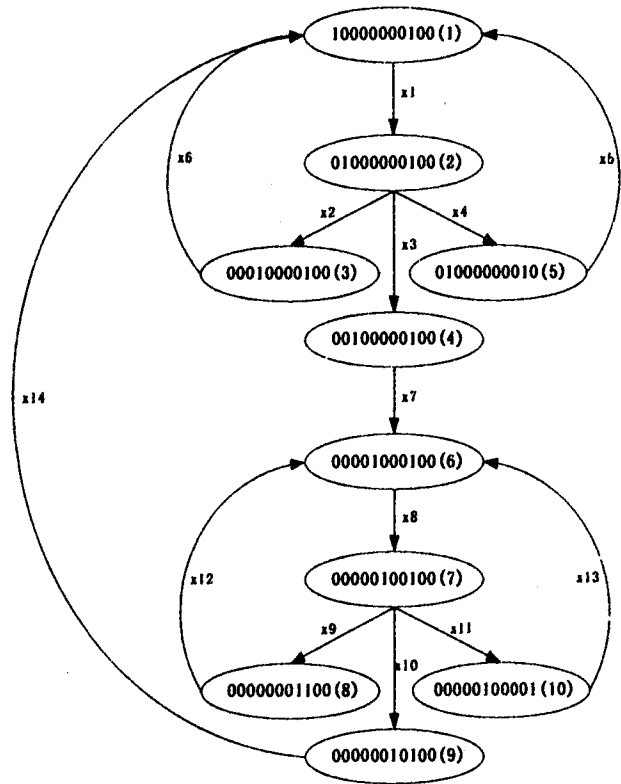


图 5 系统的状态变迁图

从图中可以看出系统共包含 10 个状态,椭圆形表示系统

的可达状态,从状态 i 到状态 j 的弧表示状态 i 到状态 j 可能的变迁。 $x_1, x_2, \dots, x_{14}$ 表示 $\lambda_1, \lambda_2, \dots, \lambda_{14}$ 的变迁平均实施速率。

timed-CPN 模型的平衡状态方程:

$$\begin{cases} x_1 m_1 - x_6 m_3 - x_5 m_5 - x_{14} m_9 = 0 \\ x_2 m_2 + x_3 m_2 + x_4 m_2 - x_1 m_1 = 0 \\ x_6 m_3 - x_2 m_2 = 0 \\ x_4 m_2 - x_5 m_5 = 0 \\ x_7 m_4 - x_3 m_2 = 0 \\ x_8 m_6 - x_7 m_4 - x_{12} m_8 - x_{13} m_{10} = 0 \\ x_9 m_7 + x_{10} m_7 + x_{11} m_7 - x_8 m_6 = 0 \\ x_9 m_7 - x_{12} m_8 = 0 \\ x_{11} m_7 - x_{13} m_{10} = 0 \\ x_{10} m_7 - x_{14} m_9 = 0 \\ m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} = 1 \end{cases}$$

其中 m_i 表示稳定概率, x_i 是马尔可夫随机过程的转移矩阵。当 $\lambda_{11}=\lambda_4$ 时,解得

$$\begin{aligned} m_1 &= \frac{2400(71+105000\lambda_4)}{131(18637+3850000\lambda_4)} \\ m_2 &= m_7 = \frac{1050000}{131(18637+3850000\lambda_4)} \\ m_3 &= m_8 = \frac{250}{131(18637+3850000\lambda_4)} \\ m_4 &= \frac{42}{131(18637+3850000\lambda_4)} \\ m_9 &= \frac{105}{131(18637+3850000\lambda_4)} \end{aligned}$$

得到系统的非稳定概率 $U=1-m_1-m_6$ 。

4.4 数值分析

图 4~5 中对应的变迁实施速率如表 1 所示。

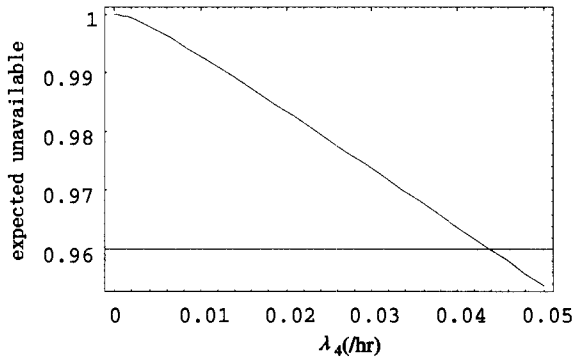


图 6 系统稳定性和自愈率 λ_4 的关系

图 6 表明了 $\lambda_{11}=\lambda_4$ 时,系统的稳定性随着自愈率增加即自愈时间间隔减少而增加。

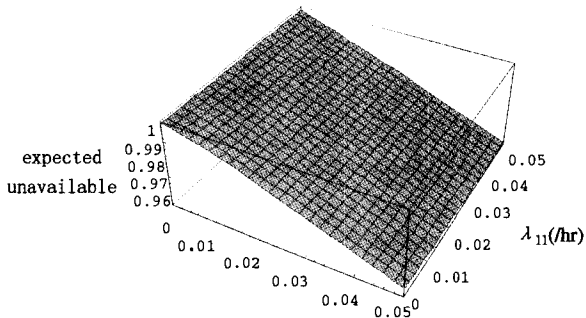


图 7 系统稳定性和 λ_4, λ_{11} 的关系

图7表明了系统的非稳定性对于自愈率 λ_4 和故障恢复率 λ_{11} 的依赖性。

结论 本文综合考虑了容错系统的硬件故障和老化现象,提出了将传统的冗余策略和软件抗衰策略相结合的技术应用在容错系统中,并分析了该系统的状态变化,给出了系统的时间颜色 Petri 网(timed-CPN)模型。在结合数据分析容错系统的行为后,得出了系统非稳定概率和模块自愈时间间隔及故障恢复率之间的关系。

参考文献

- 1 Castelli V, Harper R E, Heidelberger P, et al. Proactive management of software aging. IBM J. Research & Development, 2001, 45:311~332

- 2 Marshall E. Fatal error: how Patriot overlooked a scud. Science, 1992, 255:1347
- 3 Huang Y, Kintala C, Kolettin N, et al. Software rejuvenation: analysis, module and applications. In: Proc. 25th Int'l Symp. on Fault Tolerant Computing, IEEE CS Press, 1995. 381~390
- 4 Shereshevsky M, Cukic B, Crowel J, et al. Software aging and multifractality of memory resources. In: Proc. Int'l Conf. on Dependable Systems and Networks, IEEE CS Press, 2003. 721~730
- 5 Garg S, Telek M, Puliafito A. Analysis of software rejuvenation using Markov regenerative stochastic Petri net. In: Proc. 6th Int'l Symp. on Software Reliab. Eng. IEEE CS Press, 1995. 24~27
- 6 Genrich H J. Equivalence transformation of prt-net. In: Proc. of 9th European workshop on Application and Theory of Petri Net, Venice, Italy, 1988

(上接第273页)

分析:除以4的余数只能为0、1、2和3,使用4个状态分别代表已经读入的数字的和除以4的不同的余数的等价类:

q_0 :已经读入的数字的和除以4,余数为0的输入串的等价类;

q_1 :已经读入的数字的和除以4,余数为1的输入串的等价类;

q_2 :已经读入的数字的和除以4,余数为2的输入串的等价类;

q_3 :已经读入的数字的和除以4,余数为3的输入串的等价类;

因为不能接收空串,所以还需要一个开始状态 q_s 。

设 w 是当前读入的输入串;

q_s :在开始状态读入1时,有 $w=1$,进入状态 q_1 ;读入1时,有 $w=2$,进入状态 q_2 ;读入3时,有 $w=3$,进入状态 q_3 ;

q_0 :能引导自动机到达次状态的 w 除以4余数为0,因此, $w=4n+0$;

在此状态读入1,引导自动机到达下一状态的输入串为 $w1$,则 $w1=10(4n+0)+1=4\times 10n+1$,表明 $w1$ 属于 q_1 对应的等价类,所以自动机在 q_0 状态读入1,应该到达 q_1 状态;

在此状态读入2,引导自动机到达下一状态的输入串为 $w2$,则 $w2=10(4n+0)+2=4\times 10n+2$,表明 $w2$ 属于 q_2 对应的等价类,所以自动机在 q_0 状态读入2,应该到达 q_2 状态;

在此状态读入3,引导自动机到达下一状态的输入串为 $w3$,则 $w3=10(4n+0)+3=4\times 10n+3$,表明 $w3$ 属于 q_3 对应的等价类,所以自动机在 q_0 状态读入3,应该到达 q_3 状态;

q_1 :能引导自动机到达次状态的 w 除以4余数为1,因此, $w=4n+1$;

在此状态读入1,引导自动机到达下一状态的输入串为 $w1$,则 $w1=10(4n+1)+1=4\times 10n+11$,表明 $w1$ 属于 q_3 对应的等价类,所以自动机在 q_1 状态读入1,应该到达 q_3 状态;

在此状态读入2,引导自动机到达下一状态的输入串为 $w2$,则 $w2=10(4n+1)+2=4\times 10n+12$,表明 $w2$ 属于 q_0 对应的等价类,所以自动机在 q_1 状态读入2,应该到达 q_0 状态;

在此状态读入3,引导自动机到达下一状态的输入串为 $w3$,则 $w3=10(4n+1)+3=4\times 10n+13$,表明 $w3$ 属于 q_1 对应的等价类,所以自动机在 q_1 状态读入3,应该到达 q_1 状态;

q_2 :能引导自动机到达次状态的 w 除以4余数为2,因此, $w=4n+2$;

在此状态读入1,引导自动机到达下一状态的输入串为

$w1$,则 $w1=10(4n+2)+1=4\times 10n+21$,表明 $w1$ 属于 q_1 对应的等价类,所以自动机在 q_2 状态读入1,应该到达 q_1 状态;

在此状态读入2,引导自动机到达下一状态的输入串为 $w2$,则 $w2=10(4n+2)+2=4\times 10n+22$,表明 $w2$ 属于 q_2 对应的等价类,所以自动机在 q_2 状态读入2,应该到达 q_2 状态;

在此状态读入3,引导自动机到达下一状态的输入串为 $w3$,则 $w3=10(4n+2)+3=4\times 10n+23$,表明 $w3$ 属于 q_3 对应的等价类,所以自动机在 q_2 状态读入3,应该到达 q_3 状态;

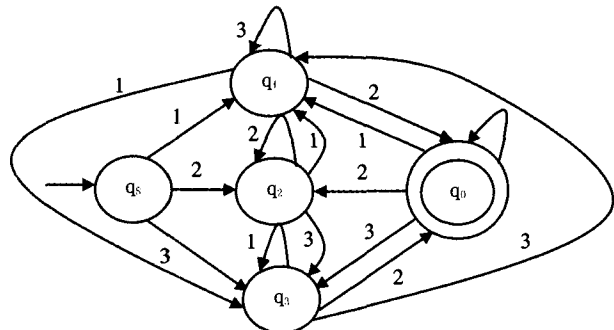
q_3 :能引导自动机到达次状态的 w 除以4余数为3,因此, $w=4n+3$;

在此状态读入1,引导自动机到达下一状态的输入串为 $w1$,则 $w1=10(4n+3)+1=4\times 10n+31$,表明 $w1$ 属于 q_3 对应的等价类,所以自动机在 q_3 状态读入1,应该到达 q_3 状态;

在此状态读入2,引导自动机到达下一状态的输入串为 $w2$,则 $w2=10(4n+3)+2=4\times 10n+32$,表明 $w2$ 属于 q_0 对应的等价类,所以自动机在 q_3 状态读入2,应该到达 q_0 状态;

在此状态读入3,引导自动机到达下一状态的输入串为 $w3$,则 $w3=10(4n+3)+3=4\times 10n+33$,表明 $w3$ 属于 q_1 对应的等价类,所以自动机在 q_3 状态读入3,应该到达 q_1 状态。

综上所述,得有限状态自动机为:



总结 本文提出了根据等价类构造一类有限状态自动机的方法。该方法可以针对所有 M 进制的数字串构成的语言,而且满足语言中的所有数字串能够整除任意正整数 N 。

参考文献

- 1 陈有祺. 形式语言与自动机[M]. 南京:南开大学出版社,1999
- 2 Sipser M 著. 张立昂,等译. 计算理论导论[M]. 北京:机械工业出版社,1996
- 3 蒋宗礼. 形式语言与自动机理论[M]. 北京:清华大学出版社,2003
- 4 阮传概. 近世代数及其应用[M]. 北京:邮电大学出版社,2002