Petri 网死锁的分析与检测①

The Analysis and Test of Petri Net's Deadlock

唐培和　（柳州广西工学院计算机工程系　545006）

摘要：本文在讨论 Petri 网静态结构和动态运行特性的基础上，给出了死锁的定义，描述了死锁的物理意义，并且给出了死锁的分析方法和检测算法。本文的研究对 Petri 网的工程应用具有基础性的重要作用。

关键词：Petri 网　死锁　算法

1　引言

Petri 网是一种系统描述和分析的工具，尤其便于描述和模拟并发系统[1,2]。对于一个系统，若能构造出它的 Petri 网模型，并对之加以分析，就能揭示出被描述的系统在结构和动态行为方面的许多重要信息。这些信息可用于对系统的性能评估，或对系统提出改进设计的建议。

Petri 网具有许多特性，如活性、死锁、可达性、安全性等等。这些性质大致上可划分为两类，一类是和初始标识有关的，它们反映 Petri 网运行时的动态特性；一类是和初始标识无关的，它们反映 Petri 网的结构特性。

"死锁"(Deadlock)是人们不希望碰到但又不可回避的系统特性。在并发系统的运行过程中，为保证系统的运行安全可靠，必须有效地解决死锁问题。解决死锁的首要任务是发现死锁，然后才能设法排除它。因此，死锁分析与检测就显得很有意义和十分必要。它是许多计算机科学工作者感兴趣并研究过的问题，尽管如此，死锁检测还是一道难题，至今仍未获得彻底的突破。

2　结构死锁及其定义

2.1　定义 1

对于 Petri 网 $C=<P,T,I,O,u_0>$，死锁 D 是位置的非空子集，即：$D\neq\Phi$，且 $D***P$，并且满足[1]：

$$\Gamma^-(D)***\Gamma^+(D)$$

其中：$\Gamma^-(D)$是 D 输入转移的集合；$\Gamma^+(D)$是 D 输出转移的集合。

由死锁定义可知，对于死锁 D，如果存在一条类型为(t，p)的弧 a，其中 $p\in D$，$t\in T$，那么也肯定存在一条类型为(q，t)的弧 b，其中 $q\in D$。也就是说：

$$t\in\Gamma^-(D)\Rightarrow t\in\Gamma^+(D)$$

同时，不难理解，如果位置集合 D_1 和 D_2 都是死锁，$D_1\neq D_2$，且 D_1***P，D_2***P。则：$D_1\cup D_2$ 也是死锁。但是 $D_1\cap D_2$ 则不一定为死锁。

2.2　定义 2

如果 D 是最小死锁，当且仅当除了 D 本身和空集 Φ 以外，D 中不再包含其他任何死锁[3,4]，$|D|\geqslant 2$。

基于网结构所定义的最小死锁包含以下三种物理意义：

(1) 对于 Petri 网 $C=<P,T,I,O,u_0>$，存在 $D***P$，如果 D 使得：

$$\Gamma^-(D)=\Gamma^+(D)$$

成立，并且 $\sum u_0(p_i)\neq 0$，其中 $p_i\in D,k=|D|$。则 Petri 网可无休止地运行下去。

(2) 对于 Petri 网 $C=<P,T,I,O,u_0>$，存在 $D***P$，如果 D 使得：$\Gamma^-(D)**\Gamma^+(D)$成立，并且 $\sum u_0(p_i)\neq 0$，其中这 $p_i\in D,k=|D|$。则总可以使得每一个转移 $t'\in\Gamma^+(D)$都无法启动。换句话说，我们总可以调度 Petri 网 C，使它到达标识 $u\in R(C,u_0)$，在标识 u 中，转移 $t'\in\Gamma^+(D)$都不是使能的。

(3) 如果 D 是 Petri 网 $C=<P,T,I,O,u_0>$中的死锁，并且 $\sum u_0(p_i)=0$，其中 $p_i\in D,k=|D|$。则对于每一个转移 $t'\in\Gamma^+(D)$，都将因为无法获得资源而不能启动。换句话说，Petri 网 C 在初始标识 u_0 下，所有转移 $t'\in\Gamma^+(D)$都不是使能的。

2　最小结构死锁分析与检测

Barkaui，Bermond，Murata 等人对结构死锁做了深入、细致的分析和描述，得出了很多结论。在这些结论中，有两个定理对最小死锁的检测很有用，它们是定理 1 和定理 2[3,4,5]。

① 本文的研究获广西自然科学基金(桂科自 0066006)资助

2.1 定理 1

设 D 是 Petri 网 $C=<P,T,I,O,u_0>$ 中的死锁，G_D 是由 $D\cup\Gamma^-(D)$ 给出的图，$\Gamma^-(t)$ 为 G_D 中转移 t 的输入位置。如果死锁 D 满足 $|D|\geqslant 2$ 是一个最小死锁，当且仅当：在 G_D 中存在一个通过 D 中所有位置的环 λ 满足，对所有的 $t\in\lambda$：

要么 $|\Gamma^-(t)|=1$；要么 $|\Gamma^-(t)|\geqslant 2$

并且存在一个交替环(或由压缩后得到)通过 $\Gamma^-(t)$ 的所有位置。

2.2 定理 2

设 D 为 Petri 网 C 的位置子集，若 D 为最小死锁，当且仅当：由 $(D,\Gamma^-(D))$ 导出的子图可以压缩为一个单一的位置，该位置不与任何转移相连接。

定理 1 和定理 2 为最小死锁的检测提供了依据和方法。根据这两个定理，可构造出如下的最小死锁检测算法。

```
step1:  * * * * D, D * * * P;
        if Γ-(D) * * * Γ+(D)
          then goto step2; {D 为死锁}
          else goto step1; {D 不是死锁}
step2:  if * * * * p∈D, * * * * * t∈Γ+(D):Γ+(p)={t}
          then goto step3;
        else end; {D 不是最小死锁}
step3:  G_D := D∪Γ-(D);
        while (|D|>1) do
          if * * * * * ω∈G_D{ω 为交替环}
            then goto step4;
            else end; {D 不是最小死锁}
step4:  q * * * * * * ω; {压缩}
        D := D - Pω∪{q}
        Γ-(D) := Γ-(D)-{t|t∈ω,Γ+(t) * * Pω};
        G_D := D∪Γ-(D);
      end while. {D 为最小死锁}
{算法结束}
```

该算法比较复杂。设 $n=|D|$，并且 $m=|\Gamma^+(D)|$. 不难知道，step1 的复杂性是 $O(m^2)$；step2 的复杂性是 $O(m,n)$；step3 的复杂性是 $O(mn)$，step4 的复杂性是 $O(m+n)$。因此算法总的复杂性为：

$O(m^2+mn^2)$

对于最小死锁，有 $m\geqslant n$，所以该算法的复杂性最多为：$O(m^3)$。

3 动态死锁及其定义

结构死锁反映了 Petri 网模型的结构特性，对建模与分析具有一定的意义。但其作用有限，原因是：

(1) 结构死锁是潜在的，有时候它并不影响网的运行。

(2) 结构死锁和网的初始标识无关，而实际上初始标识对网的运行过程影响很大，因此结构死锁难以准确地反映网的动态运行特性。

(3) 对于稍微复杂一点的 Petri 网，满足结构死锁定义的位置集合较多。因此，依据结构死锁指导正确建模比较困难。

(4) 结构死锁无法反映死锁产生的原因(如资源竞争)以及死锁产生的过程。

实际应用中，我们希望了解 Petri 网动态运行时是否产生死锁，以及死锁产生的过程，我们称这样的死锁为"动态死锁"(Dynamic Deadlock)。如果能够分析并检测出网的动态死锁，无疑对 Petri 网建模与应用具有十分重要的意义。

3.1 定义 3

对于 Petri 网 $C=<P,T,I,O,u_0>$，如果存在可达标识 $u_d\in(C,u_0)$，并且任给一个转移 $t\in T$，状态函数 $\delta(u_d,t)$ 无定义，则称标识 u_d 为死锁标识。

3.2 定义 4

对于 Petri 网 $C=<P,T,I,O,u_0>$，如果可达集 $R(C,u_0)$ 中存在死锁标识，则称 Petri 网 C 有死锁。

显然，这里定义的死锁为动态死锁，它反映网的动态运行特性。如果 Petri 网有死锁，则其可达集中必然包含相应的死锁标识。因此，动态死锁的问题可归约为 Petri 网的可达集中是否包含死锁标识的问题。如果可达集中存在死锁标识，则 Petri 网有死锁；否则 Petri 网中无死锁。为此，M. Hack 提出了一种死锁检测的方法[2]：找出所有可使网处于死锁状态的标识，然后判断它们的可达性。如果都不可达，那么 Petri 网无死锁；否则网中肯定有死锁。该方法看起来好象可行，但很难实现。原因有两个：一是难以找出所有导致死锁的标识；二是可达性问题还是一个未解决的难题。

4 动态死锁的分析与检测

4.1 可达树及其讨论

可达树(Reachability tree)是 Petri 网性能分析的重要工具之一，利用它能解决许多问题。那么可达树在动态死锁分析与检测方面是否有意义呢？经过仔细分析，发现可以利用可达树来达到目的。因为可达树具有以下三个特点：

(1) 它覆盖了网的可达集。如果 Petri 网有死锁，其死锁标识必然被可达树所覆盖。

(2) 可达树对网的可达集进行了分类，不同的标识类用不同的广义标识(含 ω 符号)来表示。这样对查找并验证死锁标识十分有利。

(3) 根据可达树,有可能了解到导致死锁的运行过程。这一点也是很有意义的。

但是,可达树也有其自身的不足,这就是由于符号 ω 的引入而导致的信息丢失。因此,单纯地依赖可达树我们很难断定一个网是否有死锁。 特别是两个不同的网还有可能产生相同的可达树,这样就使问题更加复杂化了。所以说单纯地依靠可达树来检测 Petri 网是否有死锁是比较困难的,甚至是不可能的。不过,可达树虽然相同,但产生可达树的 Petri 网是不一样的。由此,我们自然想到利用产生可达树的网来弥补可达树的不足,经过仔细分析,证明该方法不失为一种可行的办法。

4.2 基于可达树的死锁分析

由于可达树覆盖了整个可达集,并且对可达集进行了分类。因此,只要对可达树中的每一个节点所关联的标识(广义标识)进行分析,看它是否包含死锁标识就可知道 Petri 网是否有死锁了。

在可达树中,叶节点的个数一般较多,且多为重复节点(Duplicate Node)。分析完叶节点后,与叶节点相同的节点也就无需再分析了。根节点自然不需要分析。

可达树中的叶节点可分为两类:一类是端点(Termind Node),也叫死点;另一类是重复节点。对此两类节点须作分别讨论。

(1) 端点的死锁分析。如果 Petri 网 $C=<P,T,I,O,u_0>$ 的可达树中存在叶节点 x,x 是端点,则 Petri 网 C 有死锁。死锁标识就是节点 x 所关联的标识,设为 u_x,因为在端点标识 u_x 下,没有任何转移是使能的。

如果 u_x 中不含 ω 分量,则 u_x 就是一个单独的死锁标识;如果 u_x 中包含 ω 分量,则 u_x 代表一类死锁标识的集合,在此集合中,所有可达的标识都是死锁标识。

导致死锁的运行过程可用一个转移的启动序列来表示。这个序列就是在可达树中,从根节点到端点 x 的有向路径,设为 σ,$\sigma\in T^*$。显然启动序列 σ 不一定是导致 Petri 网死锁的唯一路径。典型地,如果存在转移的启动序列 λ,$\lambda\in T^*$,使得:

$$\delta(u_0,\lambda)=u_0$$

则启动序列 $\lambda n\cdot\sigma$(其中 $n\geqslant 0$)都可能导致死锁。可见,这样的路径集有可能是无限的。

(2) 重复节点的死锁分析。重复节点就是可达树中重复出现的节点。根据重复节点的特殊性,我们又把它分成两类。一类是其所关联的标识中不含 ω 分量的重复节点;另一类是包含 ω 分量的广义标识的重复节点,也称为覆盖节点。

我们设 x 为重复节点,它对应的标识为 u_x。如果 u_x 中不含 ω 分量,则 u_x 肯定不是死锁标识。这个结论是可以证明的:设可达树中有节点 x′,x 是 x′的重复节点。如果 x′出现在根节点到重复节点 x 的有向路径上,则必然有:

$$\delta(ux',\sigma)=u_x,\ \sigma\in T^*\ \text{亦即:}\ u_x\in R(C,u_x')$$

又由于 $u_x=u_x'$,因此 u_x 不可能是死锁标识。如果 x′不在根节点到重复节点 x 的有向路径上,则 x′肯定有后继节点存在。因此 u_x(或 u_x')肯定不是死锁标识。

接下来我们讨论 u_x 中包含 ω 分量的情况。

为了分析问题的方便,首先消除 u_x 中不影响死锁特性的 ω 符号。

在 Petri 网的可达树中,若存在位置 $p\in P$,$u_x(p)=\omega$,并且 p 满足:$\Gamma^-(p)\neq\Phi$,并且 $\Gamma^+(p)\neq\Phi$,则 $u_x(p)$ 不论取什么值都不影响 Petri 网的运行。原因是位置 p 只有输入而没有输出,它不影响任何转移的启动。因此,如果 u_x 中有这样的 ω 分量存在,可予以消除,以降低问题分析的复杂性。为了方便,我们用零来替换 $u_x(p)$ 对应的 ω 符号。即:$u_x(p)=0$。消除不影响死锁特性的 ω 符号后,如果 u_x 中不再有 ω 符号,则可按无 ω 分量的重复节点处理。否则,如果存在位置 $p\in P$,$u_x(p)=\omega$,不妨令 $u_x(p)=0$,从而可得到 u'_x,即:

$u_x\ u_x(p)*******0\ u'_x$

u'_x 中不再包含 ω 分量。如果 u'_x 不是死锁标识,则 u_x 中肯定不含死锁标识。因为 u'_x 是被 u_x 所覆盖的最小的一个标识,u'_x 不是死锁标识,u_x 中所有大于 u'_x 的标识自然也就不是死锁标识了。但是,如果 u'_x 是死锁标识,则不能断定 Petri 网有死锁。因为 u'_x 不一定是可达的。对此还须进一步分析。

设节点 x′出现在根节点到叶节点 x 的有向路径上,x 是 x′的重复节点。并设根节点到 x′的路径为 $\sigma\in T^*$,x′到 x 的有向路径为 $\lambda\in T^*$,$|\lambda|\geqslant 1$。

根据路径 σ,我们总可以调度 Petri 网的运行,使它到达标识 u′,$u'\in u_x'$,并且使得:

$$\delta(u',\lambda)=u',\ u'\in u_x$$

成立。令 $\Delta u=u''-u'$,Δu 可看作是一个特殊的标识,它的各个分量既可以是正的,也可以是负的,当然也可以为零。

如果任给一个位置 $p\in P$,都使得 $\Delta u(p)\geqslant 0$,则 u_x 中肯定不包含死锁标识。因为 Petri 网从节点 x′运行到叶节点 x 的过程中,不存在资源(Token)损耗。从标识 u′到 u″,u″仍然是活的,起码下面的过程可持续下去:

$$\delta(u',\lambda)=u''$$

$$\delta(u'',\lambda)=u'''$$

.....

所以广义标识 u_x 中不存在死锁标识。

如果存在位置 $q\in P$,$ux'(q)=ux(q)=\omega$,使得 $\Delta u(q)<0$,则需计算被 u_x 所覆盖的最小的可达标识,设为 u''_d。如果

u'_d 为死锁标识，则 Petri 网肯定有死锁；否则 u_x 中不含死锁标识。

u' 和 u' 是两个可达的标识，由于 $q\in P$，使得 $\Delta u(q)<0$，因此对于状态转换过程：

$\delta(u', \lambda) = u'$

肯定存在着资源损耗。根据标识分量不可能是负值的原则，我们可以通过下式求得 u'_d：

$u'd = \delta(u', \lambda^k)$

其中：$\lambda^k=\lambda\cdot\lambda\cdot\lambda\cdot\lambda\cdot\cdot\cdot\cdot\cdot\lambda$，$k=0,1,2,3,\cdots\cdots$

* * * * * * * *

获得 u'_d 后，通过网来验证 $u'd$ 是否为死锁标识，即可了解 Petri 网是否有死锁了。

导致死锁的运行过程也可以给出。设产生标识 u' 的路径为 σ'，即：

$\delta(u_0, \sigma') = u'$

则产生死锁的路径为：$\sigma'\cdot\lambda^{k+1}$

4.3 死锁检测算法

根据上面的分析，可以构造出基于可达树的死锁检测算法。设 x 为 Petri 网可达树中的叶节点，它所关联的标识为 u_x，分析 u'_x 的算法如下：

step1：如果 x 是端点，则：

- Petri 网有死锁，死锁标识为 u_x；
- 导致死锁产生的运行过程为 σ，σ 是根节点到 x 的有向路径。

step2：如果 x 是重复节点，则：

step21：消除 u_x 中对死锁特性无影响的 ω 符号，方法如下：如果 $p\in P$，$u_x(p)=\omega$，并且使得：$\Gamma^-(p)\neq\Phi$，并且 $\Gamma^+(p)\neq\Phi$，则：$u_x(p)=0$。

step22：如果 u_x 中不含 ω 分量，则 u_x 不是死锁标识

step23：如果 $p\in P$，$u_x(p)=\omega$，可按如下方法求得被 u_x 所覆盖的最小标识 u'_x：

$u_x\ u_x(p)\ * * * * * * 0\ u'_x$

如果 u'_x 不是死锁标识，则广义标识 u_x 中不含可达的死锁标识。

step24：计算被 u_x 所覆盖的最小的可达标识。

(1) 产生 u' 和 u'，并计算 $\Delta u=u'-u'$

(2) 如果 $\Delta u\geqslant 0$，则 u_x 中不含可达的死锁标识。

(3) 如果 $q\in P$，$\Delta u(q)<0$，则：$u'_d=\delta(u', \lambda^k)$，其中 $k\in N$

如果 u'_d 为死锁标识，则 Petri 网有死锁；否则 u_x 中不含可达的死锁标识。

5 结论

基于 Petri 网结构的死锁检测，有一定的局限性。首先，基于网结构的死锁定义，并不包含资源竞争的语义；其次，基于网结构的死锁检测，检测出来的死锁很可能是潜在的，也就是说，静态结构死锁反映的是系统的结构特性，和 Petri 网的动态运行过程、初始标识没有紧密的关系。

基于可达树的动态死锁检测，与 Petri 网的静态结构、初始标识以及网的运行过程相关。它反映的是 Petri 网的动态运行特性。按照本文给出的方法，既可以发现死锁，也可以给出导致死锁产生的运行过程。由于可达树中引入了符号 ω，造成了一定程度的信息丢失，给死锁检测带来了不少困难。依靠可达树、结合网结构一起研究，无疑是一种可行的办法。但该算法的实现比较复杂，有比较大的难度。

参考文献

1 F. Commoner. Denklocks in Petri Nets. Report CA7216 - 2311, Massachusetts Computer Associates, wakefield, Massachusetts, 1972.7.

2 J. L. Peterson. Petri Net Theory and the Mokelling of system. Englewood Cliffs, New Iersey, Prentice Hall, Inc., 1981.

3 K. Baukaui, B. Lemaire. An Effective characterization of Minimal Deadlock and Traps in Petri Nets Based on Graph Theory. 10th International conference on Application and Theory of Petri Nets, 1989.

4 J. C. Bermond, G. Memmi. A Graph theoretical characterization of Minimal Deadlock in Petri Nets. 4th Europer Workshop on Applications and Theory of Petri Net, 1983.

5 Tadao Murata. Petri Net: Properties, Analysis and Applications. Reprinted from PROCEDINGS OF THE IEEE 77(4), 1989. p541 - 580.