第二部分 分布式算法

第五次课

中国科学技术大学计算机系

国家高性能计算中心（合肥）

§ 3.4 同步环

研究同步环上leader选举问题的上、下界。

- 上界： 提出两个msg复杂性为O(n)的算法，显然，这样的算法的msg复杂性是最优的。但运行时间并非只与环大小n有关，还与算法使用的非普通的id相关。(与id值相关)

- 下界： 讨论

1）只基于标识符之间比较的算法

2）时间受限（即若干轮内终止，轮数只依赖于环大小,不依赖于id值）的算法

当算法受限于上述两个条件时，至少需要Ω（$n\lg n$）个msgs.

§ 3.4.1 上界 O(n)

上节已证明在异步环上leader选举问题的msg复杂度下界为Ω $(n\lg n)$，其算法的关键是msg延迟是任意长的。

因为同步环上

1) msg延迟是确定的，故同步模型是否有较好的结果呢？

2) 获取info不仅来自于接收msg,在某轮里的内附件也能获取info

本节提出两个算法，msg复杂性上界为O(n),针对单向环，但是用于双向环

1) 非均匀的：要求环中所有的结点开始于同一轮，标准的同步模型 (需知道n)

2) 均匀的： 结点可开始于不同轮，弱同步模型 (无需知道n)

§3.4.1 上界O(n)

1. Non-uniform Alg

- **基本特征**

  选择环中最小id（各id互不相同）的结点作为leader，按Phase运行，每个阶段由n个轮组成。在Phase i (i ≥ 0)，若存在一个id为i的结点，则该结点为leader，并终止算法，因此，最小id的结点被选为leader

  显然，Phase数目取决于n个节点的标志符的取值。

- **具体实现**

  Phase i包括轮：n·i+1, n·i+2, …, n·i+n

  在第i阶段开始，若一个结点的id是i，且它尚未终止，则该节点绕环发送一个msg后作为一个leader终止；若一结点的id不是i，且它收到一个msg，则它转发此msg后作为non-leader终止。

§3.4.1 上界O(n)

- 分析

正确性：显然，只有最小的标志符被选中作为leader

Msg复杂性：恰有n个msg被发送，故复杂性为O(n)。注意这n个msg均是在找到leader的那个Phase里发送的。

时间复杂性

依赖于环大小和环上最小标志符，不妨设环大小为n，最小标识符为i，则算法执行轮数为：$n\cdot(i+1)$，不妨设$i\geq 0$ //运行时间与环大小及标识符取值相关

缺点

必须知道环大小n和同步开始，下面算法克服了这些限制

①为什么id为i的结点要在phase i发msg

∵各结点互不知道彼此的id值

∴只能在第i phase，结点(id=i)发自己的id

②为什么每个phase要n轮

§ 3.4.1 上界O(n)

2. Uniform Alg

- **特点：**①无须知道环大小，②弱同步模型

  一个处理器可以在任意轮里自发地唤醒自己，也可以是收到另一个处理器的msg后被唤醒
- **基本思想**

  ① 源于不同节点的msg以不同的速度转发

  源于id为i的节点的msg，在每一个接收该msg的节点沿顺时针转发到下一个处理器之前，被延迟2^i-1轮

  ② 为克服非同时启动，须加一个**基本的唤醒阶段**，其中每个自发唤醒的结点绕环发送一个唤醒msg，该msg转发时无延迟

  ③ 若一个结点在算法启动前收到一个唤醒msg，则该结点不参与算法，只是扮演一个relay(转发)角色：即转发或没收msg

§ 3.4.1 上界O(n)

- **要点：** 在基本阶段之后，选举leader是在参与结点集中进行的，即只有自发唤醒的结点才有可能当选为leader。
- **具体实现**

  ① **唤醒**：由一个结点发出的唤醒msg包含该结点的id，该msg以每轮一边的正常速率周游，那些接收到唤醒msg之前未启动的结点均被删除(不参与选举)

  ② **延迟**：当来自一个id为i的节点的msg到达一个醒着的节点时，该msg以2^i速率周游，即每个收到该msg的节点将其延迟2^i-1轮后再转发。

Note： 一个msg到达一个醒着的节点之后，它要到达的所有节点均是醒着的。一个msg在被一个醒着的节点接收之前是处在1st阶段 (唤醒msg，非延迟)，在到达一个醒着的节点之后，它就处于2nd阶段，并以2^i速率转发(非唤醒msg，延迟)

§3.4.1 上界O(n)

③ 没收规则

a) 一个参与的节点收到一个msg时，若该msg里的id大于当前已看到的最小(包括自己)的id，则没收该msg；

b) 一个转发的节点收到一个msg时，若该msg里的id大于当前已看到的最小(不包括自己)的id，则没收该msg。

§ 3.4.1 上界O(n)

■ 算法 Alg3.2 同步leader选举

```
var waiting : init  Φ;
     asleep  : init true;    //加上relay更好？    : init false;
1: 设R是计算事件中接收msg的集合
2: s:= Φ;// the msg to be sent
3: if asleep then {
4:     asleep:=false;
5:     if R = Φ then { // p_i未收到过msg，属于自发唤醒
6:          min:=id;   //参与选举
7:          s:=s+{<id>}; // 准备发送
       }else{ //已收到过msg，但此前未启动，被唤醒故P_i不参与
8:          min:=∞; //选举，置min为∞， 使其变为relay结点
            // relay:=true; ?
       }
  }
```

§ 3.4.1 上界O(n)

```
for each <m> in R do {// 处理完收到的m后相当于从R中删去
    if m < min then { // 收到的id较小时通过
        become not elected;  // P_i未被选中
        //可用relay控制使转发节点不延迟?
        将<m>加入waiting且记住m何时加入; //m加入延迟转发
        min:=m;
    } // if m > min then it is swallowed
    if m=id then become elected; // P_i被选中
} //endfor
for each <m> in waiting do
    if <m> 是在2^m-1轮之前接收的 then
        将<m>从waiting中删去并加入S
send S to left;
```

§ 3.4.1 上界O(n)

■ **分析**

下面证明，在第1个结点被唤醒之后的n轮，只剩下第二阶段的msg，只有参与的结点才有可能被选中。

(1) 正确性

对$\forall i\in[0,n-1]$，设id_i是结点p_i的标识符，$<id_i>$是源于p_i的msg

Lemma 3.9 在参与的节点中，只有最小id的节点才能收回自己的id。

pf：①选中：设p_i是参与者中具有最小id的结点(Note：至少有1个结点须参与算法)，显然没有节点(无论是否参与)能没收$\langle id_i\rangle$；另一方面，因为在每个节点上$\langle id_i\rangle$至多延迟2^{id_i}轮，故p_i最终收回自己的id；

②唯一：除p_i外，没有别的节点p_j ($j\neq i$) 也收回自己的$\langle id_j\rangle$。

若p_j收回自己的$\langle id_j\rangle$ ，则$\langle id_j\rangle$已通过p_i及其它所有结点，但$id_i< id_j$，因为p_i是一个参与者，它将不会转发id_j，矛盾！

该引理蕴含着：恰有一个结点收回自己的msg，故它是唯一声明自己是leader的结点，即算法正确。

§ 3.4.1 上界O(n)

(2) msg复杂性

在算法的一次容许执行里，发送的msg可分为三个类型：

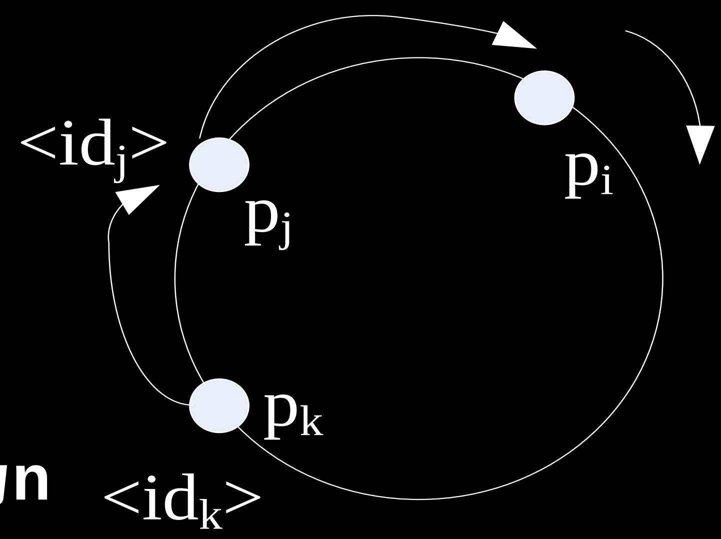
第一类：第一阶段的msg(唤醒msg)

第二类：最终leader的msg进入自己的第二阶段之前发送的第二阶段msg(其它结点发出的)

第三类：最终leader的msg进入自己的第二阶段之后发送的第二阶段msg(包括leader发出的)

Note：一个msg发送时，一开始是作为唤醒msg(非延迟)，当它到达的结点已唤醒时，msg就变为非唤醒msg(第二阶段，延迟msg)

§ 3.4.1 上界O(n)



① 第一类msg总数(第一阶段的msg)

Lemma 3.10 第一类msg总数至多为n

pf： 只要说明每个节点在第一阶段至多转发一个msg即可。

反证： 假设某结点p_i在其第一阶段转发两个msgs：一个来自p_j的$<id_j>$，一个来自p_k的$<id_k>$。不失一般性，p_j比p_k更靠近p_i(沿顺时针方向)。因此，$<id_k>$到达p_i之前先到达p_j。

若$<id_k>$是在p_j自发唤醒及发送$<id_j>$之后到达p_j，则$<id_k>$以2^{id_k}速度作为第二阶段msg继续前进；否则，p_j不是参与结点，不会发送$<id_j>$。

因此，或者$<id_k>$是作为第二阶段msg到达p_i，或者$<id_j>$未被发送，即：p_i最多收到一个第一阶段msg，矛盾！

§ 3.4.1 上界O(n)

② 第二类msg总数 (最终leader发出的msg进入自己的第二阶段之前发送的第二阶段msg)

为了求得第二类msg总数，首先说明第一个开始执行算法的结点启动之后的n轮，所有的msg均在自己的第二阶段中。

设p_i是**最早**开始执行算法的结点中的某一个，其**启动轮数为r**。

Lemma 3.11 若p_j距离p_i为k(顺时针)，则p_j 接收的第一阶段的msg不迟于第r+k轮。

§ 3.4.1 上界O(n)

pf：对距离k归纳

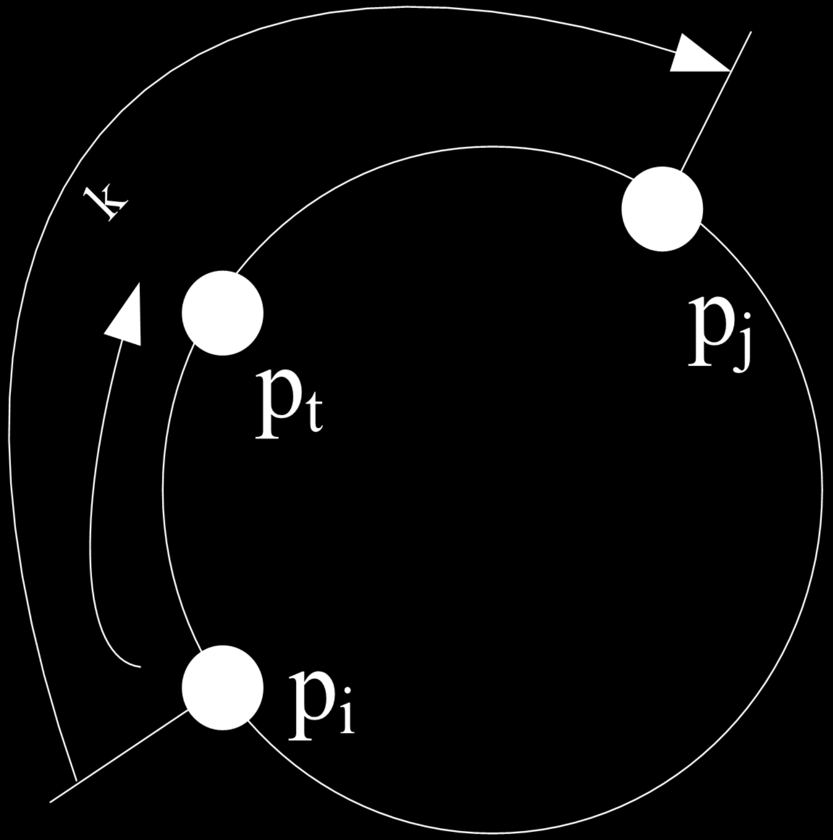
基础：k=1，因为p_i的左邻居在第r+1轮接收到p_i的msg，故引理成立。

假设：设距离p_i为k-1的结点接收第一阶段的msg不迟于r+k-1轮。

步骤

若距离p_i为k-1的结点p_t(顺时针)接收第一阶段msg时已被唤醒，则p_t已发送了第一阶段msg给邻居p_j；

否则，p_t至迟在第r+k轮转发第一阶段msg到p_j。



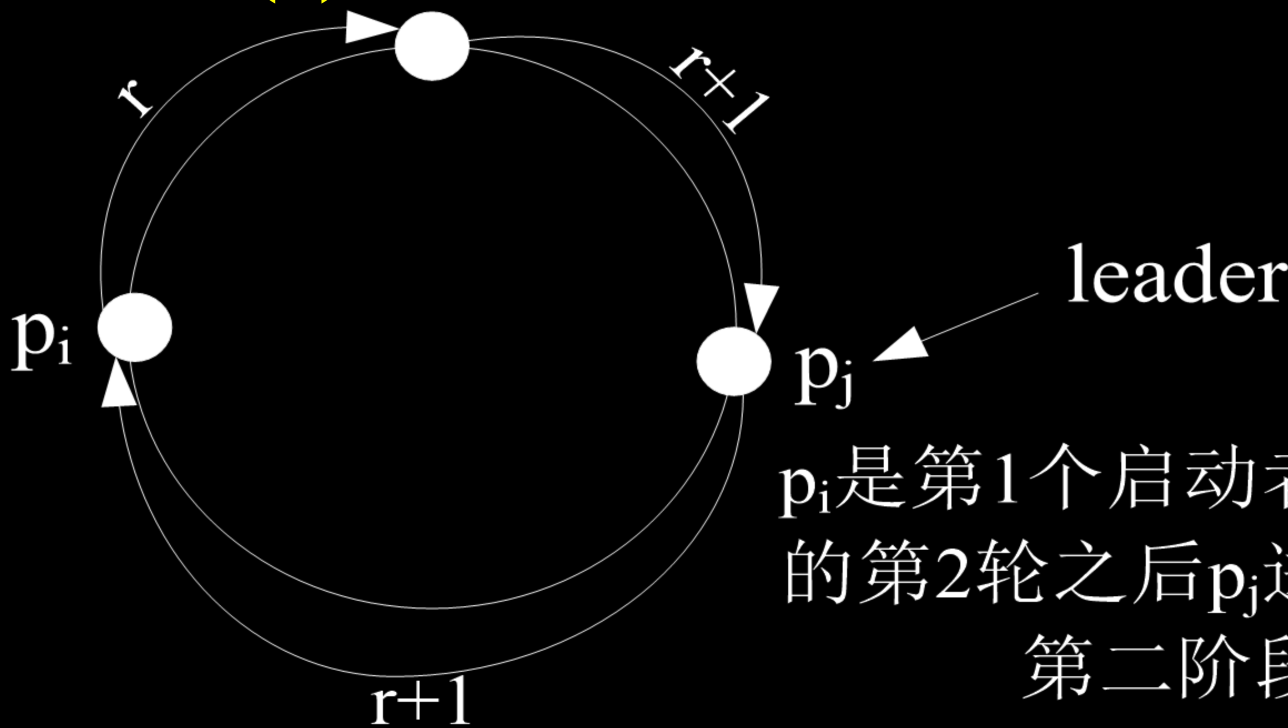
§3.4.1 上界O(n)

Lemma 3.12 第二类msg总数至多为n

pf：由引理3.10可知，在每边上至多只发送1个第一阶段的msg，又因为到第r+n轮，每边上已发送了一个第一阶段msg，故到第r+n轮之后，已无第一阶段的msg被发送。

Note： 第一阶段msg是唤醒msg，即若在p_i(第一个启动结点)发出唤醒msg绕环一周回到p_i之前已有某结点启动，则该启动结点的msg在未收到p_i的msg之前已将自己的唤醒msg向前转发。

§ 3.4.1 上界O(n)



p_i是第1个启动者，启动的第2轮之后p_j进入自己第二阶段

i) **第二类msg经历的总轮数：**

由引理3.11知，最终的leader(不一定是首个启动者)的msg进入自己的第二阶段的时刻是：算法的第1个msg被发送之后至多n轮(前n轮)，故第二类被发送的msg必是在首个启动结点的n轮之中。

ii) **在这n轮中，第二类msg数目。即第二类msg是算法启动的前n轮中非唤醒msg的总数：**

§ 3.4.1 上界O(n)

因为msg<i>在其第二阶段中，转发前须延迟2^i-1轮，所以若<i>是第二类msg，则它至多被发送$n/2^i$次。

因为较小id被转发的次数较多，故可这样构造以使msg数目最大：

所有结点均参与选举，标识符均尽可能小：0, 1, …, n-1（顺时针排列）。显然，因为id=0是leader，第二类msg中不包括leader的msg，故第二类msg总数至多是：

$$\sum_{i=1}^{n-1} \frac{n}{2^i} \le n$$

§3.4.1 上界O(n)

③第三类msg总数

(即：leader进入自己的第二阶段之后，所有非唤醒msg)

Lemma 3.13 在<id_i>返回p_i之后，没有msg被转发

pf：

设p_i是具有最小id的结点，当一结点转发<id_i>之后，该结点将不再会转发其它msg。

若<id_i>返回p_i，则所有结点均已转发过<id_i>，故再也没有其它msg被转发。

§ 3.4.1 上界O(n)

Lemma 3.14 第三类msg总数至多为2n

pf：设p_i是最终的leader，p_j是某个参与的结点，由引理3.9知，$id_i<id_j$，由引理3.13知，在p_i收回自己的id_i之后，环上不再有msg在传输。

i) 第三类msg经历的总轮数：因为在每个结点上，$<id_i>$至多延迟2^{id_i}轮(在唤醒结点上不延迟，故为至多)，所以$<id_i>$返回p_i至多经过$n\cdot 2^{id_i}$轮。

ii) 在这$n\cdot 2^{id_i}$轮中，第三类msg数目：第三类msg是在这$n\cdot 2^{id_i}$轮中发送的所有第二阶段msg(非唤醒msg)。在这$n\cdot 2^{id_i}$轮中，一个非唤醒的msg$<id_j>$被转发的次数至多为：

$$\frac{1}{2^{id_j}}\cdot n\cdot 2^{id_i}=n\cdot 2^{id_i-id_j}$$

§ 3.4.1 上界O(n)

故有：第三类msg总数至多为(包括leader)

$$\sum_{j=0}^{n-1} \frac{n}{2^{id_j - id_i}}$$

基于引理3.12的同样理由，当所有结点参与选举，及标识符为0, 1, …, n-1时有：

$$\sum_{j=0}^{n-1} \frac{n}{2^{id_j - id_i}} \leq \sum_{k=0}^{n-1} \frac{n}{2^k} \leq 2n$$

这里，$\forall\, id_i \in [0, n-1]$

Th3.15 存在一个同步的leader选举算法，其msg复杂度至多为4n

pf: 由引理3.10，3.12及3.14立即可得。

§ 3.4.1 上界O(n)

(3) 时间复杂度

由引理3.13知，当leader接收到自己的id时，计算终止。

这发生在第一个启动算法的节点之后的$O(n \cdot 2^i)$轮，其中i是leader的标识符。 // 当i=0时，为O(n)轮

//运行时间与环大小及标识符取值相关

(4) 思考

为何非唤醒msg要延迟2^i -1轮？

如何修改算法3.2来改善时间复杂性？

§ 3.4.2 有限制算法的下界Ω (n lgn)

上节中的两个算法在同步环上最坏的msg复杂度为O(n)，但两算法的缺陷为：

① 它们用一种非同寻常的方式使用id，即id决定msg延迟多长；

② 在每个容许的执行中，执行轮数依赖于id，而id相对于n而言可能是巨大的。(更主要的)

时间复杂度会表现很差

本节将说明：

① 这些性质对于基于消息的有效算法而言是固有的；

② 若一个算法中的标识符仅用于比较，则它需要Ω(nlgn)个msgs；

③ 若一个算法中，限制轮数的上界，但独立于id，则它的msg复杂度亦为Ω(nlgn)。

§ 3.4.2 有限制算法的下界Ω(nlgn)

- 同步的下界不可能从异步的下界导出

  因为上节中的算法表明：同步模型中的附加假定是必不可少的。

- 同步的下界对于非均匀和均匀算法均成立，但异步的下界只对均匀算法成立。

- 但是从同步导出的异步结果是正确的，并且提供了一个非均匀算法的异步下界。

异步通信模型中领导者选举问题所需的消息数下界为Ω(nlgn)且算法不依赖于比较的或者限时的

§3.4.2 有限制算法的下界Ω(n lgn)

1. 基于比较的算法的概念和定义

为了得到下界，假定所有处理器在同一轮开始执行

一个环是由结点表按顺时针方向指定的，结点表开始于最小标识符。

在同步模型里，算法的容许执行完全由初始配置定义，这是因为msg延迟及节点步骤的相对次序均无选择的可能。

系统的初始配置完全由环决定，即由节点标识符列表按上述规则来决定。当算法的选择可以从上下文判断清楚时，则将由环R确定的容许执行表示为exec(R).

- 匹配：若环R_1上的结点p_i和R_2上的p_j在各自的环里具有相同的位置，则称p_i和p_j是匹配的，即：匹配的结点在各自环上距其最小id的结点的距离相同。

§ 3.4.2 有限制算法的下界Ω(n lgn)

■ 基于比较的算法

直观上，若一个算法在两个相对次序相同的环上具有相同的行为，则该算法是基于比较的，形式定义：

1）序相同（order equivalent）

两个环$x_0, x_1,\dots, x_{n-1}$和$y_0, y_1,\dots,y_{n-1}$是(次)序等价的，若对每个i和j，$x_i<x_j$, 当且仅当$y_i<y_j$。

回忆一下环上的结点p_i的k-邻居是2k+1个结点的序列(下标是mod n)：

$$p_{i-k}, \dots, p_{i-1}, p_i, p_{i+1},\dots, p_{i+k}$$

序等价的概念很容易扩展到k-邻居集(所有索引按模n计算)

§3.4.2 有限制算法的下界Ω(n lgn)

2）何谓行为相同(similar)?

直观上：在序等价的环R_1和R_2上的容许执行里，发送同样的msg做同样的决策。但一般情况下，算法发送的msg包含结点的id，因此，R_1上发送的msg通常不同于R_2上发送的msg。然而，我们关注的是msg模式和决策。**所谓msg模式(pattern)是指：msg是何时何地发送的，而不是指其内容。**

- **节点在第k轮里行为相似：**考虑两个执行α_1，α_2和两个结点p_i，p_j，我们说p_i在α_1的第k轮里的行为相似于p_j在α_2的第k轮里的行为，若下述条件成立：

  ① p_i在α_1的**第k轮**里发送一个msg到其左(右)邻居当且仅当p_j在α_2的第k轮里发送一个msg到其左(右)邻居；

  ② p_i在α_1的**第k轮**里作为一个leader终止当且仅当p_j在α_2的第k轮里作为一个leader终止。

§ 3.4.2 有限制算法的下界Ω(nlgn)

- **节点的行为相似：** 若α_1里p_i和α_2里p_j在所有的$k \geq 0$轮里均相似，则称α_1里p_i和α_2里p_j的行为相似。
- **算法的行为相似：** 指每对匹配的结点行为相似

3）定义

Def 3.2 一个算法是基于比较的，若对于每对序等价的环R_1和R_2，每对匹配的节点在exec(R_1)和exec(R_2)里的行为均相似。

该定义说明，若一算法只与环上标识符之间的相对次序相关，而与具体id值无关，则该算法一定只是基于标识符的比较

§ 3.4.2 有限制算法的下界Ω (n lgn)

2. 基于比较算法的下界

设A是一个基于比较的leader选举算法，证明时考虑的环就序模式而言具有高度的对称性。即：环里存在很多次序等价的邻域。

直观上，只要两个节点有序等价的邻域，它们在A里就有同样的行为。我们将通过在一个高度对称的环里执行A来导出下界。讨论若一结点在某轮里发送一个msg，则具有序等价邻域的所有结点也在同一轮里发送一个msg。

证明的关键是：区别获得信息的轮及没有获得信息的轮.

Note：在同步环里，一个结点即使没有收到一个msg也会获得info，例如，在§3.4.1里的非均匀算法中，若第1到第n轮里未接收到msg，实际上蕴含着信息：环里没有标识符为0的结点。

§ 3.4.2 有限制算法的下界Ω (n lgn)

下面的证明关键是观察：

若某一轮r里不存在msg也对于结点p_i是有用的(指可获取info)，则仅当存在一个次序等价的不同环，在该环上的第r轮里已接收一个msg

例如，在非均匀算法中，若环上某一个结点的id为0，则在第1，2，…，n轮里均已收到msg。但对于一个次序等价的不同环(设最小id>0)，则它在前n轮里就没有任何msg存在。但同样认为前n轮对每个结点是有用的。

因此，若某一轮在任何次序等价的环上均无msg发送，则该轮是无用的，而有用的轮被称为是主动的(active)。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Def 3.3 在一个环R上的一个执行里，某轮r称为是active的，若该执行的第r轮里，某一个结点发送一个msg。当R是从上下文已知时，用r_k表示第k个active轮。

Note： 一旦环R确定，整个容许执行就确定(因为系统同步)

由于一个基于比较的算法在等价环上的行为相似，因此：

对于序等价的环R_1和R_2，一轮在exec(R_1)里是主动的当且仅当它在exec(R_2)里也是主动的。

因为消息中的信息在k个轮里只能在环上通过k个结点，所以一个结点在k轮之后的状态只依赖于它的k-邻居。

然而一个更强的性质是：一个结点在第k个主动轮之后的状态只依赖于它的k-邻居。这实际上告诉我们：信息只有在主动轮里才能获得。这一点在下面的引理中给出形式证明。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Note：该引理无需结点匹配(否则立即从定义3.2中可得结论)，但要求它们的邻居是相同的(identical)。该引理要求假设两个环是次序等价的，原因是要确保考虑中的两个执行有相同的主动轮集合，因此r_k是良定义的。

Lemma3.16 设R_1和R_2是次序等价的两个环，设P_i和P_j分别是R_1和R_2上具有相同k-邻居的两个节点，那么在exec(R_1)的第1至第r_k轮里P_i所经历的转换序列和在exec(R_2)的第1至第r_k轮里P_j所经历的转换序列相同。

//该引理蕴含：在k个主动轮结束时，P_i和P_j的状态相同

Pf：非形式地，该证明说明在k个主动轮之后，一个结点可能只知道距离自己至多为k的那些结点。形式证明对k进行归纳。

§ 3.4.2 有限制算法的下界Ω(nlgn)

① **归纳基础**：k=0，因为两个具有相同0-邻居的结点有同样的id，故它们的状态相同；

② **归纳假设**：假定每两个具有相同(k-1)-邻居的结点在(k-1)个主动轮之后有相同的状态；

③ **归纳步骤**：因为P_i和P_j有相同的k-邻居，故它们亦有相同的(k-1)-邻居。因此由归纳假设知，P_i和P_j在第(k-1)个主动轮之后处在相同的状态。又因P_i和P_j各自的邻居有同样的(k-1)-邻居，故由归纳假设知，它们各自的邻居在第(k-1)个主动轮之后也处在相同的状态。

两个主动轮之间可能有非主动轮。

因为在第(k-1)主动轮和第k主动轮之间的轮(若有的话)里，没有结点接收任何msg，故P_i和P_j及各自的邻居均处在相同的状态(Note：P_i在非主动轮里可能改变它的状态，但因为P_j有同样的转换函数，故它有同样的状态转换)。

§ 3.4.2 有限制算法的下界Ω(nlgn)

在第k个主动轮里：

i)若P_i和P_j均不接收msg，则它们在该轮结束时有相同的状态；

ii)因为P_i 和P_j的邻居状态相同，若P_i接收右(左)邻的一个msg，则P_j也接收右(左)邻同样的msg。因此，在第k个主动轮结束之后，P_i和P_j均处于相同的状态。□

下一引理将上述论断从具有相同k-邻居的结点扩展至具有次序等价的k-邻居的结点。这依赖于事实：A是基于比较的。

更进一步要求环R是有空隙的，即环R中，每两个id之间均有n个未使用的标识符。形式地，在大小为n的环上，若对于每一个标识符x，标识符x-1到x-n均不在环上，则该环称为有空隙的。

§ 3.4.2 有限制算法的下界Ω(nlgn)

引理3.16定义在两环上(P_i和P_j的k-邻居相同)，引理3.17是定义在同一个环上(P_i和P_j的k-邻居序等价)

Lemma3.17 设R是有空隙环，P_i和P_j是R上具有序等价k-邻居的两个结点。则P_i和P_j在exec(R)的第1到第r_k轮里有相似的行为。

Pf： 我们构造满足下述条件的另一个环R'：

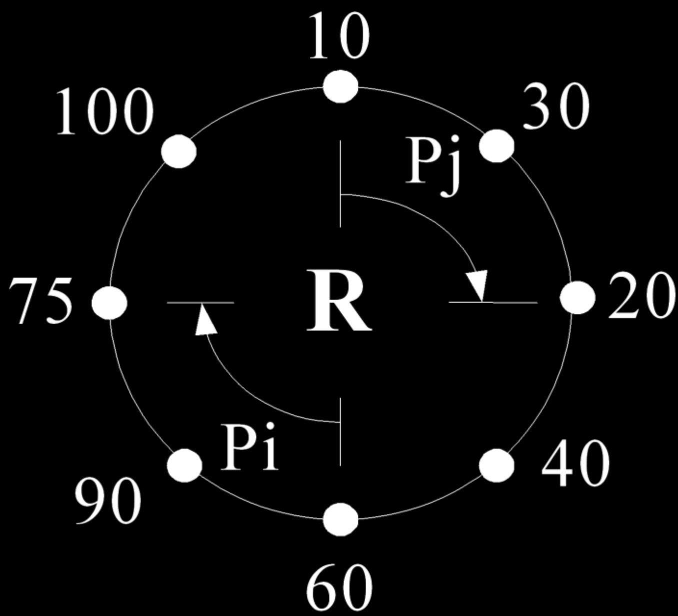
① R'中的P_j和R中P_i的有相同的k-邻居；

② R'中的标识符唯一

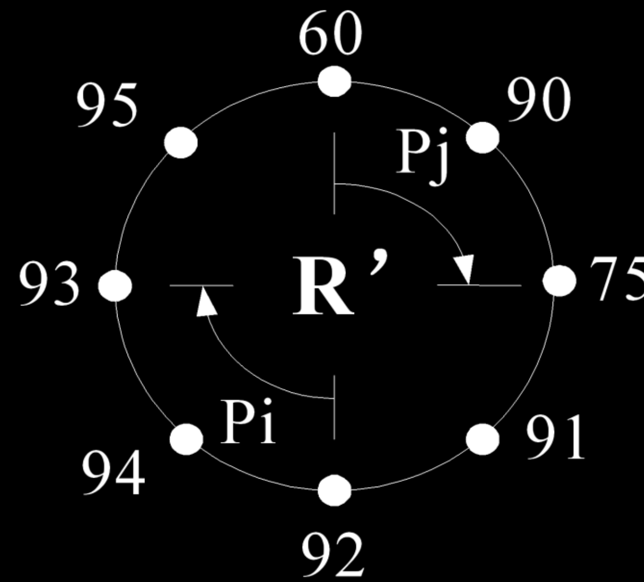
③ R'和R序等价，R'中的P_j匹配于R中的P_j。

因为R是有空隙环，故我们能够构造R'。例如，对于k=1和n=8有：

§ 3.4.2 有限制算法的下界Ω(nlgn)



R



R’

① P_i的1-邻居60，90，75　　P_j的1-邻居60，90，75

② R’中id唯一

③ R次序：　　R’次序：

10,**30**,20,40,60,90,75,100　　60,**90**,75,91,92,94,93,95

P_j与10距离为1,　　P_j与60距离为1

§ 3.4.2 有限制算法的下界Ω(nlgn)

(1)R上的P_i和R'上的P_j的前k个主动轮行为相似

因为R上P_i和R'上P_j的k-邻居相同，由引理3.16知，P_i和P_j在各自的exec(R)和exec(R')的1到r_k轮里所经历的转换序列相同，所以P_i和P_j在各自的执行exec(R)和exec(R')的1至r_k轮里的行为相似。 $P_i(R) \backsim P_j(R')$

(2)R上的P_j和R'上的P_j的前k个主动轮行为相似

因为算法是基于比较的，由定义3.2在两个序等价的环中，每对匹配的结点在各自执行中有相似的行为。而R里的P_j和R'里的P_j是匹配的，故R里的P_j在exec(R)的1至r_k轮里的行为相似于R'里的P_j在exec(R')的第1至r_k轮里的行为。 $P_j(R') \backsim P_j(R)$

(3)R上两节点的k-邻居序等价，则其行为在前k个主动轮里相似

由(1)和(2)得：R里的P_i和P_j在exec(R)的1至r_k轮里的行为相似。$P_i(R) \backsim P_j(R)$ □

§ 3.4.2 有限制算法的下界Ω(nlgn)

Th3.18 对于每个n≥8(n是2的幂)，存在一个大小为n的环S_n，使得对每个基于比较的同步leader选举算法A，在S_n上A的容许执行里发送msg的数目为Ω(nlgn)

Pf： 固定算法A。证明分2步：

(1) 构造1个高度对称(很多结点有很多序等价的邻居)的环S_n；

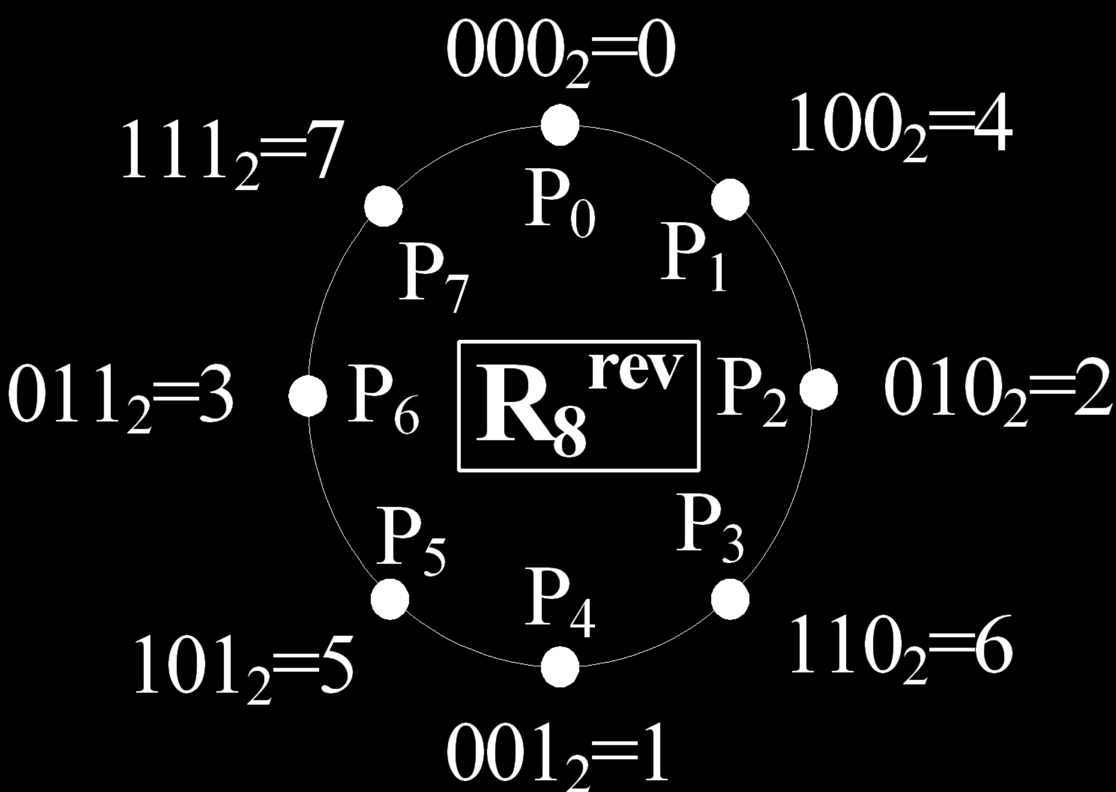
(2) S_n上发送的msg总数。

（1）构造S_n（分2步构造）

① 定义大小为n的环 R_n^{rev}：

对∀i∈[0，n-1]，设P_i的id为rev(i)，这里rev(i)是i的二进制表示的逆序列。

§ 3.4.2 有限制算法的下界Ω(nlgn)



例如，当n=8时有：

若将环 R_n^{rev} 划分为长度为j(j是2的方幂)的连续片断，则所有这些片断是序等价的(Ex3.9)。

片断数：n / j.

例如，4,2,6,1和5,3,7,0序等价

2,6,1,5和3,7,0,4序等价

§ 3.4.2 有限制算法的下界Ω(nlgn)

② 从 R_n^{rev} 构造S_n

将 R_n^{rev} 上每个id乘以(n+1)再加上n所获得的S_n是有空隙环。但这种变化不改变片断的序等价性。

(2) S_n上发送的msg总数（分3步）

①求S_n中序等价的邻居集数目（引理3.19）

②由①证明算法里主动轮数目下界（引理3.20）

③由①求每个主动轮里发送msg数目的下界(引理3.21)

由②和③的组合即可获得算法的msg复杂性下界。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Lemma3.19 对所有$k<n/8$以及每个S_n的k-邻居集N，在S_n中与N序等价的k-邻居集的个数(包括N本身)大于 $\frac{n}{2(2k+1)}$

Pf： N由2k+1个id构成，设j是大于2k+1的2的最小方幂。将S_n划分为n/j个连续片断，使某一片段包含N。

由S_n的构造可知，上述划分所得的所有片段均是序等价的。因此，至少有n/j个邻居集和N是序等价的。

设$j=2^i$，$\because 2^{i-1}<2k+1<2^i$，$\therefore j<2(2k+1)$

故与N序等价的邻居集数目> $\frac{n}{j} > \frac{n}{2(2k+1)}$

§ 3.4.2 有限制算法的下界Ω(nlgn)

Lemma3.20 在exec(S_n)里，主动轮的数目至少为n/8。

Pf：设主动轮数目T<n/8。//反证法

设P_i是exec(S_n)里被选为leader的结点，由引理3.19知，与P_i的T-邻居集序等价的T-邻居集个数大于

$$\frac{n}{2(2T+1)} > \frac{n}{2(2n/8+1)} = \frac{2n}{n+4}$$

$$\because\ n \geq 8, \quad \therefore\ \frac{2n}{n+4} > 1$$

因此，存在某个异于P_i的结点P_j，P_j的T-邻居集与P_i的T-邻居集是序等价的。由引理3.17知，P_j与P_i的行为相似，故P_j亦被选为leader，这与A是正确的算法矛盾！□

§ 3.4.2 有限制算法的下界Ω(nlgn)

Lemma3.21 对于∀k∈[1，n/8]，在exec(S_n)的第k个主动轮里，至少有 $n/2(2k+1)$ 个msg被发送。

Pf： 考虑第k个主动轮，因它是主动的，故该轮里至少有一结点发送一个msg，不妨设P_i发送一个msg。

由引理3.19知，与P_i的k-邻居集序等价的k-邻居集个数大于 $n/2(2k+1)$，因为每个k-邻居集中至少有一个结点(k≥1)，这也就是说至少有 $n/2(2k+1)$ 个结点，其k-邻居集与P_i的k-邻居集序等价。

因此，由引理3.17知，这些结点与P_i的行为相似，即它们在第k个主动轮中发送msg。

§ 3.4.2 有限制算法的下界Ω(nlgn)

由引理3.20和3.21知，在exec(S_n) 里发送msg的总数至少为：

$$\sum_{k=1}^{n/8} \frac{n}{2(2k+1)} \geq \frac{n}{6}\sum_{k=1}^{n/8}\frac{1}{k} > \frac{n}{6}\ln\frac{n}{8}$$

即Ω(nlgn)，Th3.18得证。□

注意：为了使上述定理成立，要求标识符是取自集合 {0,1,…,n^2+2n-1}。//该集合的势为n^2+2n。

原因是S_n中最小标识符为n，最大标识符为n^2+n-1 =(n+1)*rev(n-1)+n。

但是证明所用到的引理3.17要求算法在比S_n中最小标识符小n及最大标识符大n的所有标识符上是可以比较的。//有空隙环。

§ 3.4.2 有限制算法的下界Ω(nlgn)

3.时间受限算法的下界

下面的定义中，算法的时间不依赖于id，仅受限于环大小n，即使id可能是无界的(因为它们来自于自然数集合)。

Def3.4 若对任意的n，当标识符取自于自然数集合时，在所有大小为n的环上同步算法A的最坏时间是有界的，则称A为时间有界(或时间受限time-bounded)

证明时间受限的同步算法的msg复杂性的下界的基本思想是：

将时间受限算法映射为基于比较的算法。从而获得时间受限算法的msg复杂性下界Ω(nlgn)

§ 3.4.2 有限制算法的下界Ω(nlgn)

因为基于比较的算法的msg下界是针对n为2的方幂讨论的(虽然对所有n值成立)，故下面仍针对n为2的方幂情况来讨论。

为了将时间受限算法映射到基于比较的算法，需要定义在某一时间界限内算法的行为。

Def3.5 设R_1和R_2是任意两个大小为n的序等价的环，若每对匹配的结点在exec(R_1)和exec(R_2)的第1至t轮里有相似的行为，则同步算法A对于环大小为n的标识符集合S是基于t-比较的。

直观上，S上的一个基于t-比较的算法可看做这样一个算法：

它的行为与一个基于比较的算法的前t轮的行为相同，只要基于比较算法的标识符也选自于S；

若算法在t轮内终止，则它和S上基于比较的算法在所有轮上完全相同。

§ 3.4.2 有限制算法的下界Ω(nlgn)

首先要说明每个时间受限算法的行为和一个输入子集上的基于比较算法的行为相同(假设输入集合足够大)。为此须用Ramsey定理的有限版本。非形式地，Ramsey定理陈述：

设有一个集合S，若用t种颜色中的一种对每个大小为k的子集着色，则我们能够找到某个大小为ℓ的子集使得它的所有大小为k的子集有相同的颜色。若将着色看做等价类划分(若两个大小为k的子集着相同颜色，则它们属于同一等价类)，则该定理说明存在一个大小为ℓ的集合，其所有大小为k的子集是同一个等价类。

稍后，我们将对匹配结点行为相似的环着上相同颜色。

例：面试老师分为t组，每组有k个老师；面试学生集S中任何人可以到任何一组面试，则能找到某个L值，这ℓ个学生中所有k个人的子集是在同一房间面试的。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Ramsey's Theorem (finite version)

对所有正整数k,L和t，存在一个整数 $f(k,\ell,t)$使得对每一个大小至少为 $f(k,\ell,t)$的集合S，对S的k-元子集的每一个t-着色，S的某一个ℓ-元子集中所有k-元子集具有同样的颜色($\ell \geq k$) 。（着色⇔等价类划分）

在下面的引理中，用Ramsey定理将任何时间受限算法映射到基于比较的算法上。

Lemma3.22 设A是一个运行时间为r(n)的时间受限的同步算法，则对于每个n，存在一个具有n^2+2n个id的集合C_n，使得A是C_n上的一个基于r(n)-比较的算法，这里n是环大小。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Pf：固定n。设Y和Z是N(自然数集合)的任意两个n-元子集。

Y和Z称为**等价子集**，若对于每对序等价的环R_1和R_2(R_1和R_2中的标识符分别来自于Y和Z)，匹配结点在exec(R_1)和exec(R_2)的第1至r(n)轮里均有相似的行为。

该定义将N的n-元子集划分为有限多个等价类，因为行为相似仅指是否发送和接收msg，是否终止。我们对N的n-元子集着色使得两个n-元子集颜色相同当且仅当它们在同一等价类中。

由Ramsey定理，若设t是等价类(颜色)的数目，ℓ为n^2+2n，k为n，则因为N是无限集，存在一个势为n^2+2n的子集(N的子集)C_n，使得C_n的所有n-元子集属于同一个等价类。//N相当于定理中的S　$\{f(k,n^2+2n,t)\} \in N$

§ 3.4.2 有限制算法的下界Ω(nlgn)

考虑大小为n，标识符来自C_n的两个**序等价**的环R_1和R_2，设Y和Z分别是R_1和R_2的标识符集合，Y和Z显然均是C_n的n元子集，所以它们属于同一个等价类。

由等价子集定义知：匹配的结点在exec(R_1)和exec(R_2)的第1至r(n)轮里均有相似的行为。因此，A是C_n(环大小为n)上的一个基于r(n)-比较的算法(由def3.5)□

Note：因为上述引理中算法A中的标识符是特定的，故还不能直接用Th3.18导出其msg复杂性为Ω(nlgn)。因此，须使用A来构造A'，使A'的复杂性与A相同，且ids来自于集合{0,1,…,n^2+2n-1}。因为基于比较的算法的ids来自于该集合。

§ 3.4.2 有限制算法的下界Ω(nlgn)

Th3.23 对每个同步的时间有界的leader选举算法A，以及每个n>8，n为2的方幂，存在一个大小为n的环R，使得A在R上的容许执行里发送Ω(nlgn)个msgs。

Pf： 设A是运行时间为r(n)且满足定理假设的算法。固定n，设C_n是满足引理3.22的标识符集合，$c_0, c_1, \ldots c_{n^2+2n-1}$是$C_n$中按递增序排列的元素。

下面构造算法A'是基于比较的，它所执行的环大小为n，标识符集为$\{0,1,\ldots,n^2+2n-1\}$，它和A有同样的时间和msg复杂性。

§ 3.4.2 有限制算法的下界$\Omega(n\lg n)$

在算法A'中，一个标识符为i的结点执行算法就好像A在标识符C_i上执行一样。因为A在C_n上是基于r(n)-比较的且A在r(n)轮里终止，所以A'在大小为n的环上是基于比较的，且环上标识符来自于集合$\{0,1,\ldots,n^2+2n-1\}$。

由定理3.18，存在一个标识符来自于$\{0,1,\ldots,n^2+2n-1\}$，大小为n的环，算法A'在该环上发送的msg为$\Omega(n\lg n)$。

由A'的构造方法知，在大小为n、标识符来自于C_n的环上，存在A的一次执行，它发送的msg个数与A'相同。故定理得证。□

Ex3.9 若将环R_n^{rev}划分为长度为j(j是2的方幂)的连续片段，则所有这些片段是次序等价的。

下次继续！