**分布式**算**法作业**

周锋 SA14011062

**2.1 分析在同步和异步模型下，convergecast算法的时间复杂性。**

解：（1） 同步模型：最坏情况下，算法执行的每一轮中只有一个msg传递，而此时生成树汇聚最大值的算法最多执行n-1轮，也就是说同步情况下的时间复杂度为O(n-1)。

（2） 异步模型：在异步模型的汇集算法的每个容许执行中，树中每个距离pr为t的处理器至多在时刻t接收消息M，因此对于每个节点而言，它到它所有子节点中t最大的路径决定了它本身时间花费。因此在最坏情况下，仍应该是同步模型下的最坏情况，即生成树中除了末端节点每一个节点只有一个子节点，此时时间复杂度仍为O(n-1)。

**2.2 证明在引理2.6中，一个处理器在图G中是从Pr可达的，当且仅当它的parent变量曾被赋过值。**

证明：必要性：因为图G是由parent和children确定的静态图，任一节点在收到M后才会加入到图中。即可达节点收到过M，执行了算法2.2的第五行。由于是容许执行的，所以第7行（parent:=j）也会执行。

充分性：若算法2.2的第7行执行过了，因为是容许执行，则必然有第5行也执行过了。即节点收到过M。而M又是从pr发出的，所以该节点是从pr可达的。

**2.3 证明Alg2.3构造一棵以Pr为根的DFS树。**

证明：连通性：假设构造的图G存在邻居节点Pj和Pi。Pj从Pr可达，但Pi从Pr是不可达的。则Pi的parent为nil或者Pi不为Pj的child。由于G里一结点从pr可达当且仅当它曾设置过自己的parent变量。所以：

1） Pj的parent必然设置过了；

2） Pi的parent为nil或者Pi属于Pj的unexplored集合。

而算法的第11和14行决定了Pj会向Pi发送M,使得Pi的parent成为Pj,Pi成为Pj的child。

这与假设的结果矛盾。故Pi必然也是从Pr可达的。

无环： 假设G中存在一个环，P1,P2,….,Pi,P1。令P1是该环中最早接收到M的节点。则Pi是从P1可达的，且P1的parent是Pi，P1是Pi的child。而Pi在收到M后，向P1发送M。因为P1的parent已经不为空，所以P1收到来自Pi的M时，根据第16行代码，P1会向Pi放回一个<reject>信息，不会将Pi设为parent。而Pi未收到P1返回的<parent>信息，也不会将P1设为child。与前面的出的结果矛盾。故G是无环的。

图G是一棵DFS树：只需证明在有子结点与兄弟结点未访问时，子结点总是先加入树中。

设有节点P1，P2和P3。P2和P3是P1的直接相邻节点。P1在第12~14行中先选择向P2发送M，则P1当且仅当P2向其返回一个<parent>（第17行，第22行）时才有可能向P3发送M。而P2仅在其向所有的相邻节点发送过M后才会向P1返回<parent>。所以P2的子节点是永远先于P3加入树中的，即G是DFS树。

**2.4 证明Alg2.3的时间复杂性为O(m)。**

证明：

同步模型：每一轮中，根据算法，有且只有一个消息(M or Parent or Reject)在传输，从算法的第6 、14、16、20、25行发送消息的语句中可以发现：消息只发往一个处理器结点，除根结点外，所有的处理器都是收到消息后才被激活，所以，不存在多个处理器在同一轮发送消息的情况，所以时间复杂度与消息复杂度一致。

异步模型：在一个时刻内至多有一个消息在传输，因此，时间复杂度也与消息复杂度一致。消息复杂度：对任一边，可能传输的消息最多有4个，即2个M ，2个相应M 的消息（Parent or Reject），所以消息复杂度为O(m)

综上，该算法的时间复杂度为O(m)。

**2.5 修改Alg2.3获得一新算法，使构造DFS树的时间复杂性为O(n)。**

解：

（1）在每个处理器中维护一个本地变量，同时添加一个消息类型，在处理器Pi转发M时，发送消息N通知其余的未访问过的邻居，这样其邻居在转发M时便不会向Pi转发。

（2）在消息M和<parent>中维护一个发送数组，记录已经转发过M的处理器名称。

两种方式都是避免向已转发过M的处理器发送消息M，这样DFS树外的边不再耗时，时间复杂度也降为O(n)。

**3.1 证明同步环系统中不存在匿名的、一致性的领导者选举算法。**

证明：在匿名系统中，每个处理器在系统中具有相同的状态机。由Lemma3.1可知，设算法A是使环上某个处理器为leader的算法。因为环是同步的，且只有一种初始配置。在每轮里，各处理器均发出同样的message，所以在各轮里各个处理器接收到相同的message，则状态改变也相同。所以所有处理要么同为leader，要么同时不为leader。故同步环系统中匿名的、一致性的领导者选举算法的算法是不存在的。

**3.2 证明异步环系统中不存在匿名的领导者选举算法。**

证明：每个处理器的初始状态和状态机相同，除了接收消息的时间可能不同外，接收到的消息序列也相同。所以最终处理器的状态也是一致的。由于处理器处理一条消息至多需要1单位时间，若某时刻某个处理器宣布自己是leader，则在有限时间内，其它处理器也会宣布自己是leader。故异步环系统中匿名的领导者选举算法是不存在的。

**3.9 若将环Rrev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。**

证明：对一个整数P(0≤P≤n−1),可以表示为：

其中m=lg n,则有rev(P)=。

设P、Q在同一个片段上，P1、Q1在同一片段上，且设这两个片段时相邻的，由模运算的加法可得：P1=P+ l；Q1=Q + l。l表示片段的长度，l=2k。

又因为：

且P、Q在同一个片段上，有

|P-Q|<l=2k

所以存在r(0≤r≤k),满足 ar ≠ br。否则，|P−Q|≥l。这与P、Q在同一个片段上矛盾。

设s=min⁡{r},则根据rev(P),rev(Q)的表示方法可得：

sign(rev(P)-rev(Q))=sign (as-bs)

而

显然，P与P1的前k位相同，Q与Q1的前k位相同。由0≤s≤k得

sign(rev(P1)-rev(Q1))=sign (as-bs)

这两个相邻片段是序等价的，根据等价的传递关系，可得所有的片段都是次序等价。

**附1：“表面上，1-time复杂性至少等于时间复杂性，因为T2假定下的最坏时间不会高于O2假定下的时间。但事实并非如此，而往往O1和O2假定之下的1-time复杂性是前一种时间复杂性的一个下界。”为什么one-time复杂性是时间复杂性的下界呢？**

解：考虑运行在环上的分布式算法的1-time时间复杂性和时间复杂性。

<1> 1-time时间复杂性：

满足条件O2：发送和接收一个msg之间的时间恰好是一个时间单位，每个阶段节点转发消息都是同步进行，从而1-time时间复杂度仅与环直径相关，为O(D)。

<2> 时间复杂度：

满足条件T2：一个msg的发送和接收之间的时间至多为一个时间单位，即为O(1)。节点转发消息并非同步进行，消息转发轨迹可能呈链状结构，时间复杂性与环节点个数相关，为O(n)。

例如：echo协议，即应答协议，主要用于调试和检测中，是路由也是网络中最常用的数据包，可以通过发送echo包知道当前的连接节点有哪些些路径，并且通过往返时间能得出路径长度。echo算法的实现，如果转发消息同步进行，则对应1-time时间复杂性，为O(D)；如果不同步转发消息，网络路径可能呈链状结构，即对应时间复杂度O(N)。Note：考虑时间复杂度，任一节点可以在O(d)时间内将询问包发送到网络上的其它节点，但却可能需要O(N)的时间接收其它节点发来的响应包。

**附2：算法3.2（同步Leader选举算法）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？**

解：

<1> 降低消息复杂度（Id最小的节点被选举为Leader，Leader节点消息的转发速度最快）。

<2> 方案1：添加Relay变量，保证消息在转发节点不延迟，时间复杂度由O(n\*2^i)降为O(N\*2^i+n-N)，N为自发唤醒的节点数。

方案2：原算法延迟函数为f(id)=2^id，时间复杂度为O（n\*2^i）。通过重新定义延迟函数来降低时间复杂度，如f(id)=c\*id等。消息复杂度提高？

Note：思考方案2中消息复杂度和时间复杂度的关系！！！