分布式算法作业

学号：SA17011142　姓名：姜贵平

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

解：仿照 PPT 给出的广播算法时间复杂性的证明，给出 convergecast 在同步模型下时间复杂度的分析：在同步模型中，假设树的高度为 d，在 convergecast 算法的每个容许的执行中，高度为 t 的处理器在第（d - t）轮接受消息。

归纳基础：t = d – 1，每个叶节点的父节点在第一轮中接受来自于叶节点的消息 M。

归纳假设：假设树上每个高度为 t（t > 1）的处理器在第（d - t）轮接收到来自孩子结点的消息。

归纳步骤：假设根节点 Pr 的一个孩子节点为 Pi，则 Pi 高度为 1，由归纳假设可知 Pi 在第（d - 1）轮中接收到消息。那么由算法描述可知，根节点将在第 d 轮中接收到消息。

所以 convergecast 算法的时间复杂度为 O(d)，d 为生成树高度。

同样，异步模型中 convergecast 算法的时间复杂度为 O(d)，证明过程和上述过程类似，只是将处理器在第（d - t）轮接收消息修改为之多在（d - t）轮接收消息。

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

解：

充分性：如果一个处理器它的 parent 变量赋过值，那么根据算法的描述，只有当该处理器从其邻居结点第一次接收到消息 M 时，才会对 parent 变量赋值。而消息 M 最初由根节点 Pr 发送，所以该处理器是从 Pr 可达的。

必要性：如果处理器是从 Pr 可达的，那么从 Pr 发送的消息 M 能够到达该处理器。又因为当处理器第一次接收到消息 M 时，会设置自身的 parent 变量，所以处理器的 parent 变量被赋过值。

综上，命题得证。

2.3 证明 Alg2.3 构造一棵以 P_r 为根的 DFS 树。

解：

连通性：

假设图 G 中存在结点 Pi 从根节点 Pr 不可达，因为网络是连通的，假设存在结点 Pj，Pj 和 Pi 相邻，但是 Pj 由 Pr 可达。因为 G 中结点从 Pr 可达当且仅当它设置过自己的 parent 变量，所以 Pi 结点的 parent 变量在整个过程中仍未 nil，而 Pj 在某节点上已经设置过自己的 parent 变量，所以 Pj 会向结点 Pi 发送消息，这样结点 Pi 接收消息后会将自己的 parent 变量设置为 Pj，而这种执行结果与假设矛盾，所以 Pr 对每一点都是可达的。

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，根据算法描述，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。

算法构造的树为 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。假设 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)。

解：

同步模型：在算法执行的每一轮中，有且仅有一个消息（M，<Parent>或<Reject>）在网络中传输，同样在每一轮中仅有一个处理器会接收到这个消息，所以时间复杂度和消息复杂度一致，为 O(m)。

异步模型：在算法执行的时刻中，每个时刻至多有一个消息在网络中传输（异步系统中 msg 延迟至多为 1 个时间单位）。所以和同步系统类似，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)，并证明。

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)。

Ex3.1 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。

解：

在匿名算法中，环中每个处理器在系统中有相同的状态机，一致性算法则表明了算法不需要直到系统中的处理器数量 n。由引理 3.1 可知，匿名算法在同步系统中每一轮结束时状态是相同的。所以在选举算法中，一个处理器宣布自己是 leader，则其他处理器也会宣布自己是 leader，这和选举算法的假设矛盾。所以同步环系统中不存在匿名的、一致性的领导者选举算法。

Ex3.2 证明异步环系统中不存在匿名的领导者选举算法。

在异步系统中，每个消息的延迟至多为一个时间单位。

非均匀算法：在匿名算法中，每个处理器对于固定的处理器数量 n，有相同的状态机。每个处理器的初始状态也都相同。在异步系统中，处理器接收处理一个消息至多需要一个时间单位，如果某时刻其中一个处理器宣布自己是 leader，那么在其他所有的处理器接收到消息后，也会宣布自己是 leader。所以异步环系统中不存在匿名的、非均匀的 leader 选举算法。

均匀算法：均匀算法不需要知道处理器的数量 n。和非均匀算法类似，在匿名算法中，如果在某个时刻有一个处理器宣布自己是 leader，那么在有限的时间内其他处理器也会宣布自己是 leader。所以异步环系统中不存在匿名的、非均匀的 leader 选举算法。

综上，异步环系统中不存在匿名的领导者选举算法。

Ex3.9 若将环R^{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。

证明：对一个整数P($0 \le P \le n-1$),可以表示为：

$$P = \sum_{i=1}^{m} a_i \cdot 2^{i-1}$$

其中m=lg n,则有rev(P)=$\sum_{i=1}^{m} a_i \cdot 2^{m-1}$。
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，$l=2^k$。
又因为：

$$Q = \sum_{i=1}^{m} b_i \cdot 2^{i-1}$$

且P、Q在同一个片段上，有

$$|P-Q|<l=2^k$$

所以存在$r(0\le r\le k)$,满足$a_r \neq b_r$。否则，$|P-Q|\ge l$。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：

$$\mathrm{sign}(\mathrm{rev}(P)-\mathrm{rev}(Q))=\mathrm{sign}(a_s-b_s)$$

而

$$P1 = P + 1 = \sum_{i=1}^{m} a_i \cdot 2^{i-1} + 2^k$$

$$Q1 = Q + 1 = \sum_{i=1}^{m} b_i \cdot 2^{i-1} + 2^k$$

显然，P与P1的前k位相同，Q与Q1的前k位相同。由$0\le s\le k$得

$$\mathrm{sign}(\mathrm{rev}(P1)-\mathrm{rev}(Q1))=\mathrm{sign}(a_s-b_s)$$

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。