第二次作业

SA23011253 任永文

1. 板书 2 的思考题：是否对于每一执行的每一断言都成立的断言/性质就是不变式？

   (a) 不是，不变式除了满足该断言在每次执行的每个断言都成立外还要满足每次转移过程中该断言都成立。

2. 板书 2 的 Ex：给出一个转移系统 S 和断言 P，满足 P 在 S 中总为真，但 P 不是 S 不变式的。

假设有一个转移系统 $S: y = \ln x + 1 (x > 0)$，断言 $P1: x > 0, P2: x > -1$。在该系统中，P1 是不变式而 P2 不是，因为转移前 P2 成立不能保证 P2 成立。

3. Ex 证明图 G 里一处理器从 Pr 可达当且仅当它曾设置过自己的 parent 变量。

   (a) 若处理器是 Pr 可达的，则说明收到过 m，由于是容许执行因此 parent 被赋值过。

   (b) 若处理器被赋值过，由于是容许执行因此收到过 m，因为 m 是从 Pr 发出的，所以处理器是 Pr 可达的。

4. EX2.1 分析在同步和异步模型下汇聚算法的时间复杂性。

   (a) 同步模型
   引理：在汇聚算法的每个容许执行中，树中每个高为 t 的子树根节点在第 t 轮里收到所有孩子节点的 msg。
   （1）第一轮时，高为 1 的子树根节点收到来自叶子节点的 msg。
   （2）假设第 t 轮时，高为 t 的子树根节点收到所有孩子节点的 msg, 则第 t-1 轮开始时，这些节点向父节点发送 msg。所以第 t+1 轮时高为 t+1 的子树根节点收到所有孩子的 msg。
   定理：所以 d 为树高时，存在一个时间复杂度为 O(d) 的同步汇聚算法。

   (b) 异步模型
   引理：在汇聚算法的每个容许执行中，树中每个高为 t 的子树根节点在时刻 t 收到所有孩子节点的 msg。
   （1）当 t=1 时，各叶子节点的 msg 从叶子节点传输到父节点，由异步模型的时间复杂性知，各叶子结点至多在时刻 1 收到所有子节点的 msg。
   （2）设 Pi 是树高为 t 的子树根节点，Pj 是 Pi 的子节点，因此 Pj 至多是树高为 t-1 的子树根节点，由归纳假设知，Pj 至多在 t-1 时刻收到 Pj 子节点的所有 msg，所以，Pi 至多在时刻 t 收到所有孩子的 msg。
   定理：所以 d 为树高时，存在一个时间复杂度为 O(d) 的异步汇聚算法。

5. EX2.3 证明 Alg2.3 构造一棵以 Pr 为根的 DFS 树。

   (a) 证明连通。假设某节点在 G 中从 Pr 不可达，因网络是联通的，若存在两个相邻节点 Pi 和 Pj，使得 Pj 在 G 中是从 Pr 可达的，而 Pi 是不可达，因为 G 里面一结点从 Pr 可达当且仅当它曾经设置过自己的 parent 变量，所以 pi 的 parent 变量在整个执行中仍为 nil，而 pj 在某点上已设置过自己的 parent 变量，由算法可知 Pi 仍保留在 Pj 的 unexplored 数组中，因此 Pi

被搜索到之前算法不会终止，因为是容许执行，所以 Pi 最终会被搜索到，并被 parent 赋值，因此得出矛盾，故图中所有节点从 Pr 均可到达。

(b) 证明无环。假设有一环，Pi1Pi2...Pi1，若 Pi 是 Pj 的孩子，则 Pi 在 Pj 第一次收到 M 之后第一次收到 M，因为每个处理器在该环上是下一个处理器的双亲，这就意味这 Pi1 在 Pi1 在第一次收到 M 之后第一次收到 M，矛盾，所以无环。

(c) 由算法可知，只有当前结点的邻居都已被搜索之后，结点才向上一层发送 parent 消息，因此算法构造的是一棵深度优先树。

6. EX2.4 证明 Alg2.3 的时间复杂性为 O(m)。

(a) 在同步模式下，由算法可知，算法中当且仅当有一个消息（Parent 或 Reject）在传输，且只发往一个处理器结点，除根节点外，所有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，因此消息复杂度和时间复杂度一致。

(b) 在异步模式下，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。

(c) 对任一边可能传输的消息最多有 4 个，即 2 个 M 和 2 个回应 M 的消息（Parent 或 Reject），所以时间复杂度为 O(m)。

(d) 综上，时间复杂度为 O(m)。

7. EX2.5 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。

```
code for processor Pi,0<=i<=n-1
Wr parent:init nil;
children:init
unexplored:init all the neighbors of Pi;

upon receiving no msg:
if(i=r) and (parent=nil) then
{
    parent:=i;
    send <already> to all processors in unexplored;
    for all Pj in unexplored;
    remove Pj from unexplored;
    send M to Pj;
}
upon receiving <already> from Pj;
{
    remove Pj from unexplored;
}
upon receiving <M> from Pj;
{
```

```
        parent :=j;
        remove Pj from unexplored;
        if (unexplored !=empty) then
        {
            for all Pk in unexplored;
            remove Pk from unexplored;
            send M to Pk;
            send <already> to all prcessors in unexplored;
        }else send <parent> to parent;
    }else send <reject> to Pj;
    upon receiving <parent> or <reject> from neigh our Pj;
    if receiving <parent> then add j to children;
    if unexplored is empty then
    {
        if parent !=i then send <parent> to parent;
        terminate;
    }else
    {
        for all Pk in unexplored;
        remove Pk from unexplored;
        send M to PK;
    }
```

证：该算法避免了向已转发过 M 的处理器发送消息 M，这样 DFS 树外的边不再耗时，时间复杂度降为 O(n)。

8. 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。

证：设一个非均匀大小为 n 的环 R，匿名算法 A 选中某处理器为 leader，由于环 R 是同步的且只有一种初始配置，所以 R 上 A 只有唯一的合法执行，最终每个处理器状态相同，都宣布自己是 leader，与选举算法矛盾。

9. Ex3.2 证明异步环系统中不存在匿名的领导者选举算法。

证：由于同步算法是异步算法的一个特例，即每轮时间相同传输距离相同，同步系统中不存在匿名选举算法故异步系统也不存在，最后每个处理器都会宣布自己是 leader，与选举算法矛盾。

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

证：任意整数 P 可表示为 $P = \sum_1^m a_i 2^{i-1}$，则 $rev(P) = \sum_1^m a_i 2^{+2^k-i}$。

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

已知 $P=\sum_1^m a_i 2^{i-1}, Q=\sum_1^m b_i 2^{i-1}, P1=\sum_1^m a_i 2^{i-1}+2^k, Q1=\sum_1^m b_i 2^{i-1}+2^k$，显然 P 与 P1 的前 k 位相同，Q 与 Q1 的前 k 位相同。

由于 $|P-Q|<l=2^k$，因此所以存在 $r(0\leq r\leq k), a_r\neq b_r$。设 $s=min\{r\}$, 则根据 rev(P),rev(Q) 的表示方法可得：$sign(rev(P)-rev(Q))=sign(a_s-b_s)$。由 $(0\leq s\leq k)$ 得 $sign(rev(P1)-rev(Q1))=sign(a_s-b_s)$，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的。