Lemma 1 in every configuration, the following is true:

(a) at most one token has $\text{hasToken} = 1$
(b) if no process has $\text{hasToken} = 1$, then there is exactly one token message in transit
(c) if any process has $\text{hasToken} = 1$, then there is no message in transit
(d) if $\text{status}_i = \text{CS}$ then $\text{hasToken}_i = 1$
(e) if $\text{status}_i = \text{ES}$ then $i$ appears exactly once in $RQ_i$
(f) if $\text{status}_i \neq \text{ES}$ then $i$ is not in $RQ_i$

$C_i \xrightarrow{c_{i+1}} C_{i+1}$

Theorem 2 The LRME algorithm satisfies ME

Proof: Since if a process $i$ is in $\text{CS}$, then $\text{hasToken} = 1$ and by Lemma 1 no two processes can simultaneously hold $\text{hasToken} = 1$ and Thm follows.

To prove the fairness property, we consider the chain of requests initiated by a node in the ES and show the token travels around the network causing requests to move to the heads of their queues, so eventually any node can enter CS.

Lemma 3 Let $i$ and $j$ be neighboring nodes.

(a) if neither $i$ nor $j$ are token-holders, then $i$ and $j$ are link-consistent;

(b) if $i$ is the token-holder and has completed MakeOG/AckOG handshake since then, then $i$ and $j$ are link-consistent.

Definition 1 A node $i$ is token-holder if either $\text{hasToken}_i = 1$ or a Token message is in transit from $i$ to a neighbor of $i$. 
Definition 2  Nodes $i$ and $j$ are link-consistent if either $i \in OG_j$ and $j \in IC_i$, OR $i \in IC_j$ and $j \in OG_i$.

Lemma 4  For every configuration $C_t$ of the execution, $\vec{G}_t$ is a token-oriented DAG.

Definition 3  The directed link $(i, j)$ is in $\vec{G}_t$ iff $(i, j) \in E$ and either: (1) $j$ is token-holder or (2) neither $i$ nor $j$ are token-holders and $j \in OG_i$ (By lemma 3, $i \in IC_j$).

Proof: (of lemma 4) Proof is by induction on sequence of configurations. Basis is obvious For inductive step, assume $\vec{G}_t$ is token-oriented. We prove $\vec{G}_{t+1}$ is token-oriented.

Suppose in the event between $C_t$ and $C_{t+1}$ node $j$ receives a Token message from node $i$. Then token holder changes from $i$ in $C_t$ to $j$ in $C_{t+1}$. By Lemma 3, $i$ is link-consistent with its other neighbors. Thus, for every neighbor $k \neq j$, the link $(i, k)$ does not change direction between $C_t$ and $C_{t+1}$. All links incident on $j$ become incoming to $j$ in $\vec{G}_{t+1}$, so $j$ becomes a sink. Since $i$ was previously the only sink, and it’s no longer a sink, therefore $j$ is the only sink.

This doesn’t create a cycle since only the links incident on $j$ change the direction, so $j$ will be a part of any new cycle. But $j$ is a sink, so it cannot be a part of a cycle.

Lemma 5  The following are true in every configuration $C_t$:

(a) if $i \in RQ_j$ or a Request message is in transit from $i$ to $j$, then $(i, j) \in \vec{G}_t$

(b) Suppose $|RQ_i| > 0$ and $i$ is not a token-holder. Then, there is a neighbor $j$ of $i$ s.t. exactly one of these hold:

(i) exactly one Request message is in transit from $i$ to $j$ and $i$ is not in $RQ_j$ OR
(ii) no Request message is in transit from $i$ to $j$, and exactly one copy of $i$ is in $RQ_j$.

Also, for every neighbor $k \neq j$, there is no Request message in transit from $i$ to $k$, and $i$ is not in $RQ_k$.

(c) If $|RQ_i| = 0$ then there is no Request message in transit from $i$ and $i$ is not in $RQ_k$ for any neighbor $k$.

(d) If $i$ is the token-holder, then $i$ is not on $RQ_k$ for any neighbor $k$, and no Request message is in transit from $i$. Also, if $\text{hasToken}_i = 1$, then $i$ is not in $RQ_i$.

Given a configuration a request chain for any node $i$ is a maximum-length sequence of node identifiers $<p_1, p_2, \ldots, p_m>$ s.t. $i = p_1$, and $p_l \in RQ_{p_{l+1}}$ for each $l, 1 \leq l \leq m$.

Lemma 6  In every configuration $C_t$ of the execution, there is exactly one request chain for each node $i$ and it contains no repeated id’s.