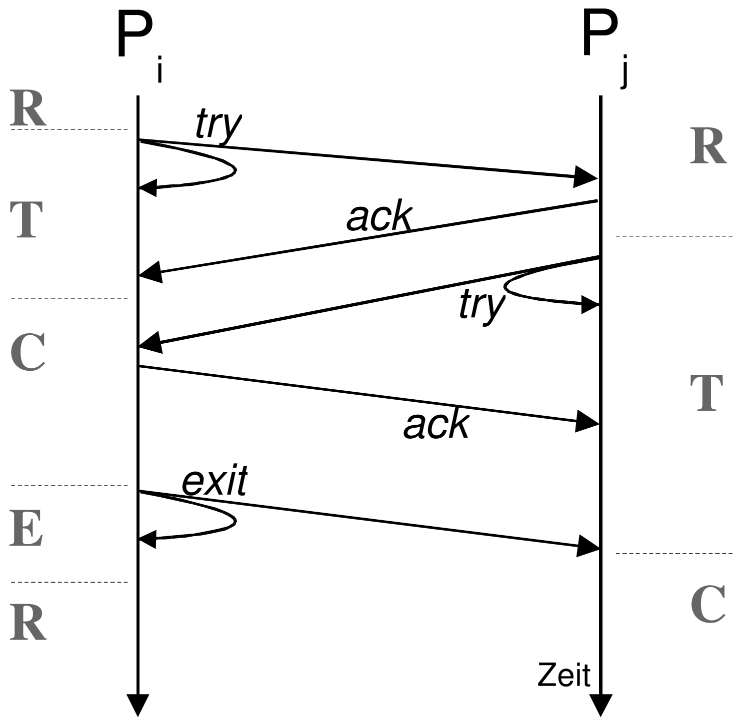
Mutual Exclusion
LogicalTimeME (I)
LamportTime Transformation

- jeder Prozess verwaltet lokale *clock*-Variable c, initial 0
- *clock* wird bei jedem Ereignis, das am Prozess auftritt, erhöht:
  - **send event**: *clock* wird um 1 erhöht, *clock* wird an Nachricht gehängt
  - **receive event**: *clock* wird erhöht, so dass sie größer ist als vorher und größer als Zeitstempel der Nachricht
- Logische Zeit: (c,i) → totale Reihenfolge

Mutual Exclusion
LogicalTimeME (II)

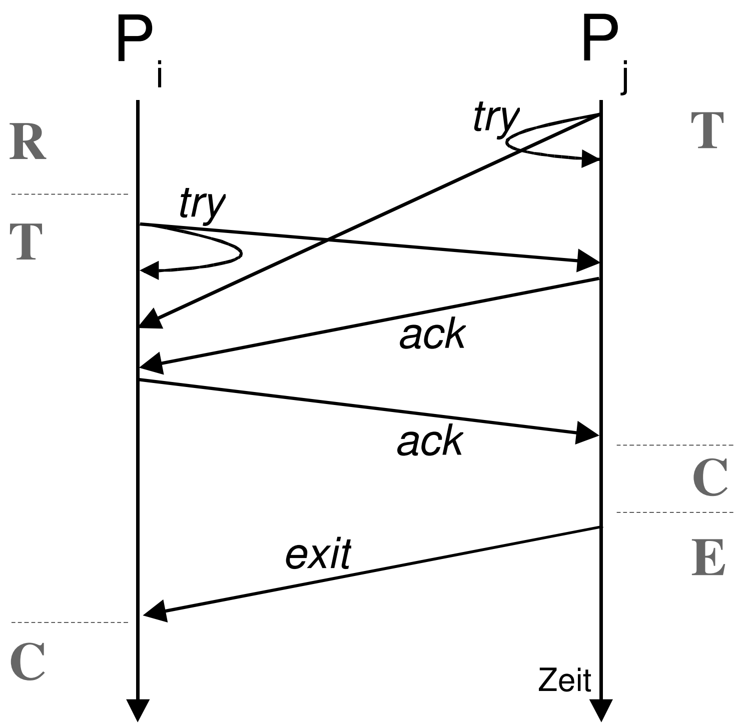
- P_i verwaltet $\mathit{history(j)}_i$: Nachrichten von P_j und dessen Logische Zeiten t_j
- *try* wird mit *ack* beantwortet
- P_i kann crit_i ausführen, wenn:
  - letztes *try* hat $\mathit{history(i)}_i$ erreicht
  - jede Anfrage mit $t_j < t_i$ von der P_i gehört hat wurde befriedigt
  - von jedem P_j eine Nachricht mit $t_j > t_i$ empfangen

Mutual Exclusion
LogicalTimeME (III)
Beispiel 1



P_j muss auf *exit* von P_i warten

Mutual Exclusion
LogicalTimeME (IV)
Beispiel 2



$t_i > t_j$

$\Rightarrow P_i$ muss auf *exit* von P_j warten

$t_j < t_i$

$\Rightarrow P_j$ darf **C** betreten

t_i: Logische Zeit des *try*-Broadcast-Ereignisses von P_i