Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - principle of operation
    - broadcast marker
    - upon receipt of marker record own state, and record any incoming message from another process until that process has recorded its state (these messages then belong to the channel between the processes)
    - processes may record their state at different points in time, but the differential is always accounted for by the state of the channel in between

![Diagram showing processes Pi and Pj with markers a and b]
Snapshots

♦ Chandy-Lamport Algorithm for the determination of consistent global states
  ‣ principle of operation
    – broadcast marker
    – upon receipt of marker record own state, and record any incoming message from another process until that process has recorded its state (these messages then belong to the channel between the processes)
    – processes may record their state at different points in time, but the differential is always accounted for by the state of the channel in between
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - any number of processes may at any time concurrently initiate snapshot-taking
  - a process initiating snapshot-taking follows the marker receiving rule (see below)
  - marker **sending** rule
    - a) record own state
    - b) broadcast marker
    - a) and b) must precede any other local actions or message send/receive events
  - marker **receiving** rule
    - if $P_i$ has not yet recorded own state (first marker is being received)
      - record own state
      - start recording all messages received on all incoming channels
    - if $P_i$ has already recorded own state
      - record state of channel on which marker was received
      - stop recording that channel
Chandy-Lamport Algorithm for the determination of consistent global states

Chandy and Lamport’s ‘snapshot’ algorithm

Marker receiving rule for process \( p_i \)

On \( p_i \)'s receipt of a marker message over channel \( c \):

\[ \text{if (} p_i \text{ has not yet recorded its state) it} \]
\[ \text{records its process state now;} \]
\[ \text{records the state of } c \text{ as the empty set;} \]
\[ \text{turns on recording of messages arriving over other incoming channels;} \]

\[ \text{else} \]
\[ p_i \text{ records the state of } c \text{ as the set of messages it has received over } c \]
\[ \text{since it saved its state.} \]
\[ \text{end if} \]

Marker sending rule for process \( p_i \)

After \( p_i \) has recorded its state, for each outgoing channel \( c \):

\[ p_i \text{ sends one marker message over } c \]
\[ (\text{before it sends any other message over } c). \]
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - example:

  
  \[
  \begin{array}{c|c|c|c|}
  & \mathbf{P_i} & \mathbf{P_j} \\
  \hline
  \text{recorded state} & <$1000, 0$> & <$50, 2000$> \\
  \mathbf{P_i} & <$1000, 0$> & \text{<>} & \text{<>} & \text{<>} & \text{<>} \\
  \mathbf{P_j} & \text{<>} & \text{<>} & \text{<>} & \text{<>} & \text{<>} \\
  \end{array}
  \]

  - recorded state
    
    \[
    \begin{align*}
    \mathbf{P_i} & : <$1000, 0$>, \mathbf{P_j} : <$50, 1995$>, c_{ij} : \text{<>}, c_{ji} : <$5 \text{ widgets}$>
    \end{align*}
    \]
Snapshots

♦ Chandy-Lamport Algorithm for the determination of consistent global states
  ‣ example:

  - recorded state
    \[ P_i: <$1000, 0>, P_j: <$50, 1995>, c_{ij}: <> \], \[ c_{ji}: <(5 \text{ widgets})> \]
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - Theorem: The Chandy-Lamport Algorithm terminates
    - Proof sketch:
      - Assumption: a process receiving a marker message will record its state and send marker messages via each outgoing channel in finite period of time.
      - If there is a communication path from \( p_i \) to \( p_k \), then \( p_k \) will record its state a finite period of time after \( p_i \).
      - Since the communication graph is strongly connected, all processes in the graph will have terminated recording their state and the state of incoming channels a finite time after some process initiated snapshot taking.
Snapshots

- Chandy-Lamport Algorithm for the determination of consistent global states
  - Theorem: Snapshots taken by the Chandy-Lamport Algorithm correspond to consistent global states
  - Proof:
    Let $e_i$ and $e_k$ be events at $P_i$ and $P_k$, and let $e_i \rightarrow e_k$.
    Then, if $e_k$ is in the cut, so is $e_i$.
    That means, if $e_k$ occurred before $P_k$ recorded its state, then $e_i$ must have occurred before $P_i$ recorded its state
    - $k = i$: obvious.
    - $k \neq i$: assume $P_i$ recorded its state before $e_i$ occurred
      - as $k \neq i$ there must be a finite sequence of messages $m_1, \ldots, m_n$ that induced $e_i \rightarrow e_k$
      - then, before any of the $m_1, \ldots, m_n$ had arrived, a marker must have arrived at $P_k$, and $P_k$ must have recorded it’s state before $e_k$ occurred, hence a \textit{contradiction} to the above assumption
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - Observation: Chandy-Lamport algorithm records a possible global system state, but the actual execution of the system that initiated the snapshot taking may never have reached this global system state.
  - Example:
Snapshots

♦ **Chandy-Lamport Algorithm for the determination of consistent global states**
  ‣ Observation: Chandy-Lamport algorithm records a possible global system state, but the actual execution of the system that initiated the snapshot taking may never have reached this global system state.
  ‣ Example:

![Diagram of BGP states](image-url)

- Observed snapshot state
- Actual computation
- State transitions
- Marking events
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - Observation: Chandy-Lamport algorithm records a possible global system state, but the actual execution of the system that initiated the snapshot taking may never have reached this global system state.
  - Example:
Snapshots

- **Chandy-Lamport Algorithm for the determination of consistent global states**
  - Observation: Chandy-Lamport algorithm records a possible global system state, but the actual execution of the system that initiated the snapshot taking may never have reached this global system state.
  - Example:
Snapshots

- Chandy-Lamport Algorithm for the determination of consistent global states
  - Reachability Theorem: Let $\text{Sys} = e_0, e_1, ..$ the linearization of a system execution. Let
    - $S_{\text{init}}$ the initial global state of the system immediately before Chandy-Lamport snapshot-taking was initiated by the first process,
    - $S_{\text{snap}}$ the recorded snapshot state, and
    - $S_{\text{final}}$ the global system state after the algorithm terminated.
  Then there is a permutation $\text{Sys}' = e'_0, e'_1, ..$ of $\text{Sys}$ such that
    - $S_{\text{init}}, S_{\text{snap}}$ and $S_{\text{final}}$ occur in $\text{Sys}'$ and
    - $S_{\text{snap}}$ is reachable from $S_{\text{init}}$, and
    - $S_{\text{final}}$ is reachable from $S_{\text{snap}}$.

© Pearson Education 2001