Synchronous Model

- Directed Graph $G=(V,E)$
  - $V$: Set of processes (nodes)
  - $E$: Set of communication links
    - $(u,v) \in E \Rightarrow u$ can send a message to $v$ and $v$ is a neighbor of $u$. Is bidirectional unless said otherwise
  - $\text{In-nbrs}_i$, $\text{out-nbrs}_i$ for each process $i$
  - $M$: Set of messages
At each node/process $i$

- $\text{States}_i$: set of states
- $\text{Start}_i$: start/initial state (may be many)
- $\text{FinalState}_i$: final state (may be many)
- $\text{Msgs}_i$: function: $\text{states}_i \times \text{out-nbrs}_i \rightarrow \mathbb{M} \cup \{\text{null}\}$, message generation function
- $\text{Trans}_i$: state transition function
  - $\text{States}_i \times \{\mathbb{M}\}^k \rightarrow \text{states}_i$
Each process does the following in lock step synchrony (with other processes):

- Apply message generation function to current state, generate messages and send to recipients.
- Apply state transition function to current state and incoming messages and go to new state; Remove all messages from channel.
Synchronous Algorithm

- All processes start from an initial state
- All communication links are empty
- Execute one round after another
- Terminate when you reach a final state
Complexity

- **Time complexity:**
  - Number of rounds until all required outputs are produced or until all processes halt.
  - Processes start at different times?

- **Message complexity**
  - Total number of non-null messages
  - Number of bits