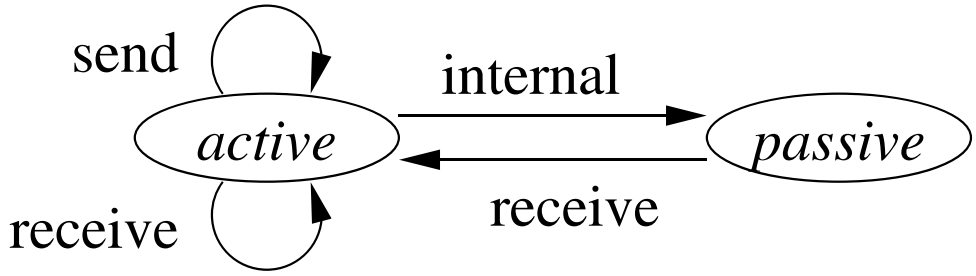
Termination detection

The *basic* algorithm is terminated if (1) each process is passive, and (2) no basic messages are in transit.



The *control* algorithm concerns termination detection and announcement.

Announcement is simple; we focus on detection.

Termination detection shouldn't influence basic computations.

Dijkstra-Scholten algorithm

Requires a centralized basic algorithm, and an undirected network.

A *tree* T is maintained, which has the initiator p_0 as the root, and includes all active processes. *Initially*, T consists of p_0.

cc_p estimates (from above) the number of children of process p in T.

- When p sends a basic message, $cc_p \leftarrow cc_p + 1$.
- Let this message be received by q.
  - If q isn't yet in T, it joins T with parent p and $cc_q \leftarrow 0$.
  - If q is already in T, it sends a control message to p that it isn't a new child of p. Upon receipt of this message, $cc_p \leftarrow cc_p - 1$.
- When a noninitiator p is passive and $cc_p = 0$, it quits T and informs its parent that it is no longer a child.
- When the initiator p_0 is passive and $cc_{p_0} = 0$, it calls *Announce*.