Raymond’s Algorithm
Kerry Raymond, “A Tree-Based Algorithm for Distributed Mutual Exclusion”
ACM Transactions on Computer Systems, 1989

Processes are arranged in an undirected logical tree.

**Process State:**
- usingResource: true if process is accessing resource; false otherwise
- holder: either self or neighbor, indicates direction of token
- reqQ: FIFO queue, each element is self or neighbor, initially empty
- asked: true if request has been sent to current holder; false otherwise

```plaintext
function assignToken()
    if (holder== self) and (¬usingResource) and (reqQ ≠ empty)
        holder = dequeue(reqQ)
        asked = false
    if holder == self
        usingResource = true
    else
        send token to holder

function makeRequest()
    if (holder ≠ self) and  (reqQ ≠ empty) and (¬ asked)
        send request to holder
        asked = true

To request resource:
    Add self to reqQ
    assignToken()
    makeRequest()

On receipt of request from neighbor X:
    Add X to reqQ
    assignToken()
    makeRequest()

On receipt of token:
    holder = self
    assignToken()
    makeRequest()

To release resource:
    usingResource = false
    assignToken()
    makeRequest()
```