

lock (1  
on lock? let thru 1, lock (empty queue)  
else acquire  
set in queue  
release

can promise available again  
wait for signal  
acquire  
send back signal  
return lock  
unlock (acquired\*)  
empty queue? let through 1, unlock  
else pop signal & notify  
release

tries to wait for available pop  
grow (n)  
acquire  
push available  
unlock n times

acquire (7  
mux = lock (1  
pop available  
if available not empty  
unlock (mux)

return popped  
release (i)  
acquire  
push available  
unlock (mux)  
guarantees handoff or now available