Input: table L and sorted array H of SPM-relevant suffixes
with common prefix u of length k
generic function *process* to postprocess an *SPM*
Output: All suffix-prefix matches $\langle r, s, \ell \rangle$
such that $\ell \geq \ell_{min}$ and u is a prefix of s

$T :=$ [ ] ▷ empty list
$W :=$ [ ] ▷ empty list
with each lcp-interval associate an integer *itv.firstinW*
run Algorithm 1 with the following functions:
function *process_leafedge*(*firstedge*, *itv*, (p, q)) ▷ p is read number and q is offset
    if $itv.lcp \geq \ell_{min}$ **then**
        if *firstedge* **then**
            $itv.firstinW := |W| + 1$
        if $q = 0$ **then** ▷ (p, q) refers to whole read
            append p to W
        if $q + itv.lcp = |r_p|$ **then**
            append p to T
    else
        W := [ ]
function *process_branchedge*(*firstedge*,*itv*,*itv*′)
    if $itv.lcp \geq \ell_{min}$ **then**
        if *firstedge* **then**
            $itv.firstinW := itv'.firstinW$
    else
        W := [ ]
function *process_lcpinterval*(*itv*)
    if $itv.lcp \geq \ell_{min}$ **then**
        for all $p \in T$ **do**
            for all $j \in W[itv.firstinW \ldots |W|]$ **do**
                process $\langle r_p, r_j, itv.lcp \rangle$ ▷ call generic function to process *SPM*s
        T := [ ]