```
procedure derivesToLambda( L∈N a non-terminal of grammar,
  T a stack )
returns true if ∃{p_i}⊆P permitting L⇒* λ.

Given P the production rules of a reduced grammar and the stack T
is empty on the first call of procedure.

foreach ( p∈P with LHS L ) do (
  if( p in T ) then (
    continue loop
  )
  if ( p is L→λ ) then (
    return true
  )
  if ( RHS of p contains a terminal ) then (
    continue loop
  )
  let allderivelambda ← true
  foreach ( X_i∈N in RHS of p ) do (
    push p onto T
    allderivelambda ← derivesToLambda(X_i,T)
    pop T
    # it takes just one X_i ⇏* λ to mean LHS of p ⇏* λ
    if( allderivelambda is false ) then (
      break loop
    )
  )
  if( allderivelambda is true ) then (
    return true
  )
)
return false
```