```
procedure followSet( A∈N, set T )
A is a non-terminal of the grammar (A∈N) and T is
an empty set on the first call of procedure.
returns the set of symbols {t ∈ Σ$ | S ⇒* αAtβ},
  and an updated set T

Recall notationally: Σ$ = Σ + {$}; 𝒴 ∈ N ⋃ Σ$;
C∈N; α,β ∈ (N⋃Σ$)*; P is the set of grammar production rules;
S is the goal or starting symbol of the grammar;
∅ is the Empty Set.

if( A∈T ) then (
  return ∅, T
)

add A to T
# F for Follow set
let F be a set
foreach ( p∈P with A in RHS of p ) do (
  foreach ( 𝒴 at the instances of A in the RHS of p ) do (
    let π be the sequence of all symbols in N⋃Σ$ following 𝒴
    # |π| = 0 if A is last symbol of rule
    if ( |π| > 0 ) then (
      # I for ignorable
      G, I ← firstSet(π, ∅)
      F ← F ⋃ G
    )

    if( |π| is 0 OR (
          π ∩ Σ$ = ∅ AND
          derivesToLambda(δ) is true for all δ∈π ) ) then (
      G, I ← followSet(LHS of p, T)
      F ← F ⋃ G
    )
  )
)
return F, T
```