```
procedure followSet ( A \in N, set T )
A is a non-terminal of the grammar (A \in N) and T is
an empty set on the first call of procedure.
returns the set of symbols \{t \in \Sigma_{\$} | S \Rightarrow^* \alpha A t \beta\},
   and an updated set T
Recall notationally: \Sigma_{\$} = \Sigma + \{\$\}; \quad \mathcal{Y} \in N \bigcup \Sigma_{\$};
C \in N; \alpha, \beta \in (N \cup \Sigma_{\$})*; P is the set of grammar production rules;
S is the goal or starting symbol of the grammar;
\emptyset is the Empty Set.
if ( A \in T ) then (
   return \emptyset, T
)
add A to T
\# F for Follow set
let F be a set
foreach ( p \in P with A in RHS of p ) do (
   foreach ( \mathcal{Y} at the instances of A in the RHS of p ) do (
      let \pi be the sequence of all symbols in N \cup \Sigma_{\$} following \mathscr{Y}
      \# |\pi| = 0 if A is last symbol of rule
     if ( |\pi| > 0 ) then (
        # I for ignorable
        G, I \leftarrow \text{firstSet}(\pi, \emptyset)
        F \leftarrow F \bigcup G
     )
     if ( |\pi| is 0 OR (
              \pi\cap\Sigma_{\$}=\emptyset and
              derivesToLambda (C) is true for all \mathcal{C} \in \pi ) ) then (
        G, I \leftarrow \text{followSet(LHS of } p, T)
        F \leftarrow F \bigcup G
     )
  )
)
return F,T
```