```
procedure LLTabularParsing( ts, LLT, P )
ts is a token stream with peek and pop operations, LLT is an LL(1)
parsing table as described by the text.
returns a parse tree of the sentence in ts,
  or FAILS when ts does not emit a sentence belonging to the grammar
  deriving LLT.

Recall notationally: S is the starting symbol (goal) of the grammar,
Σ$ = Σ + {$} are the terminals of the language augmented by $,
N are the non-terminals and P is the production
rule list of the grammar (indexed by entries of LLT).


let T be a tree with an initial root node Root
let Current be a reference to a node of T
let K be a stack
Current ← Root
push S onto K
while ( |K| > 0 ) do (
  x ← K.pop()
  if ( x ∈ N ) then (
    # Next token may not predict a p ∈ P
    p ← P[LLT[x][ts.peek()]] or FAIL
    push MARKER onto K
    R ← RHS of p
    in reverse order, push the elements of R onto K
    let n ← new tree node for non-terminal x
    append n as the rightmost child of Current
    Current ← rightmost child of Current
  ) else if ( x ∈ Σ$ OR x is λ ) then (
    if ( x ∈ Σ$ ) then (
      # Next token must be what is expected
      if ( x ≠ ts.peek() ) then FAIL
      # Update x with the token type and source value from ts
      x ← ts.pop()
    )
    append x as the rightmost child of Current
  ) else if ( x is a MARKER ) then (
    Current ← Current.parent
  )
)

# Current now points to Root which has a singular child S
return Root's only child
```