Link for github repo:

<https://github.com/AndreiFoidas/LFTC>

<https://github.com/AndreiFoidas/LFTC/tree/main/src/main/java>

# Parser

Is a class containing 3 fields: grammar (the Grammar class), a table (LR0Table class) and canonicalCollection (a List of List of LR0Items). A LR0Item is a class that represents the Item from the theoretical definition: [A → α.β]. It has the fields: nonTerminal (here is A), dotPosition (position of the dot on the right hand side, here 1) and content (here αβ).

## Operations

* goto(state, symbol): for every item in the state that has the symbol “symbol” where the dot indicates the position, it returns the closure of that item but with the dot position shifted to the right
* closure(items): for every item in the list of items, if the dot position shows a nonTermial, we search and add all the productions of that nonTerminal to the closure set
* canonicalCollection(): we will keep adding to the canonicalCollection until there are no new elements we can add by the following rule: for every state in the collection, we compute the goto of that state and every symbol of the Grammar, if the result is not empty or the collection doesn’t have the result already, we add it

## Tests

