

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Inginerie Software și Automatică**

**Tabanschi Nichita FAF-222**

**Report**

*Laboratory work n.5*

***of Limbaje Formale și Automate***

Checked by:

**Dumitru Cretu**, *university assistant*

DISA, FCIM, UTM

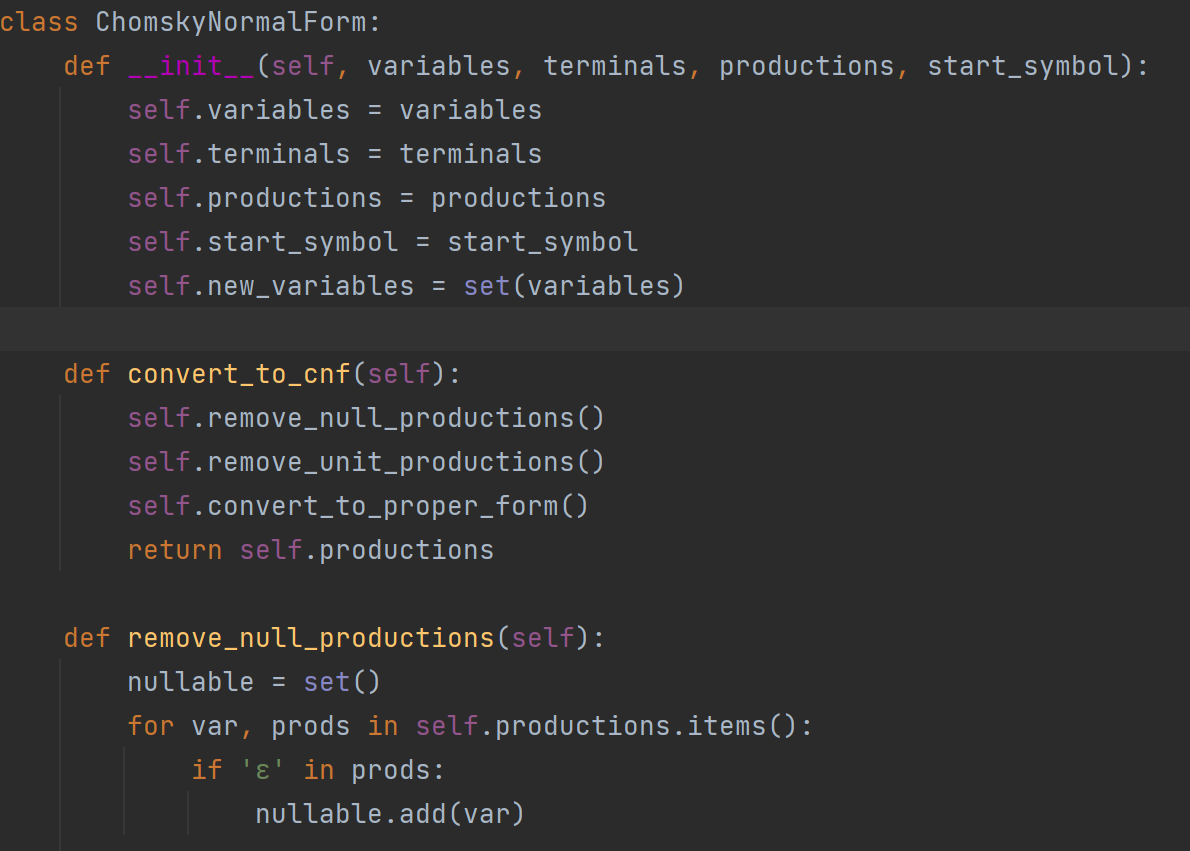
**Chișinău – 2024**

### Course: Formal Languages & Finite Automata

## Objectives:

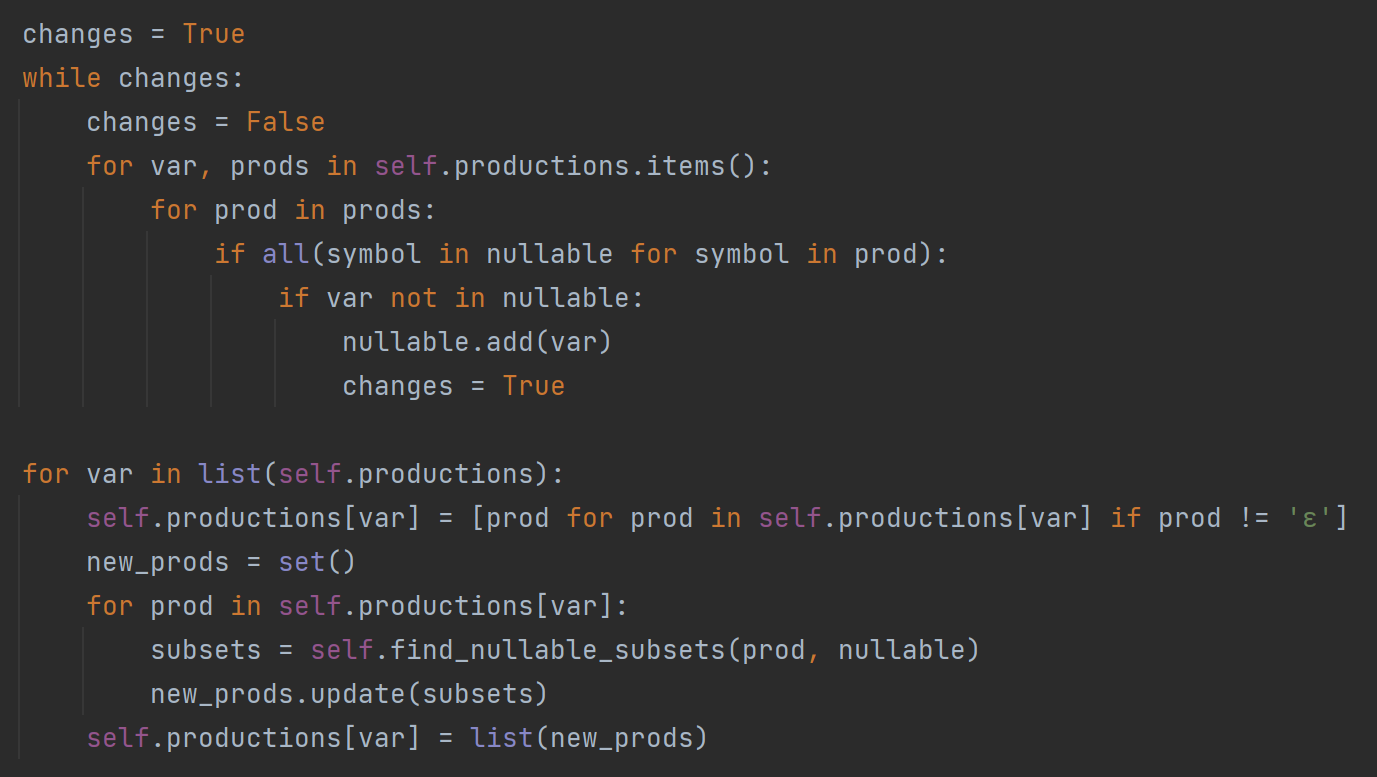
1. Learn about Chomsky Normal Form (CNF) [1].
2. Get familiar with the approaches of normalizing a grammar.
3. Implement a method for normalizing an input grammar by the rules of CNF.
   1. The implementation needs to be encapsulated in a method with an appropriate signature (also ideally in an appropriate class/type).
   2. The implemented functionality needs executed and tested.
   3. A **BONUS point** will be given for the student who will have unit tests that validate the functionality of the project.
   4. Also, another **BONUS point** would be given if the student will make the aforementioned function to accept any grammar, not only the one from the student's variant.

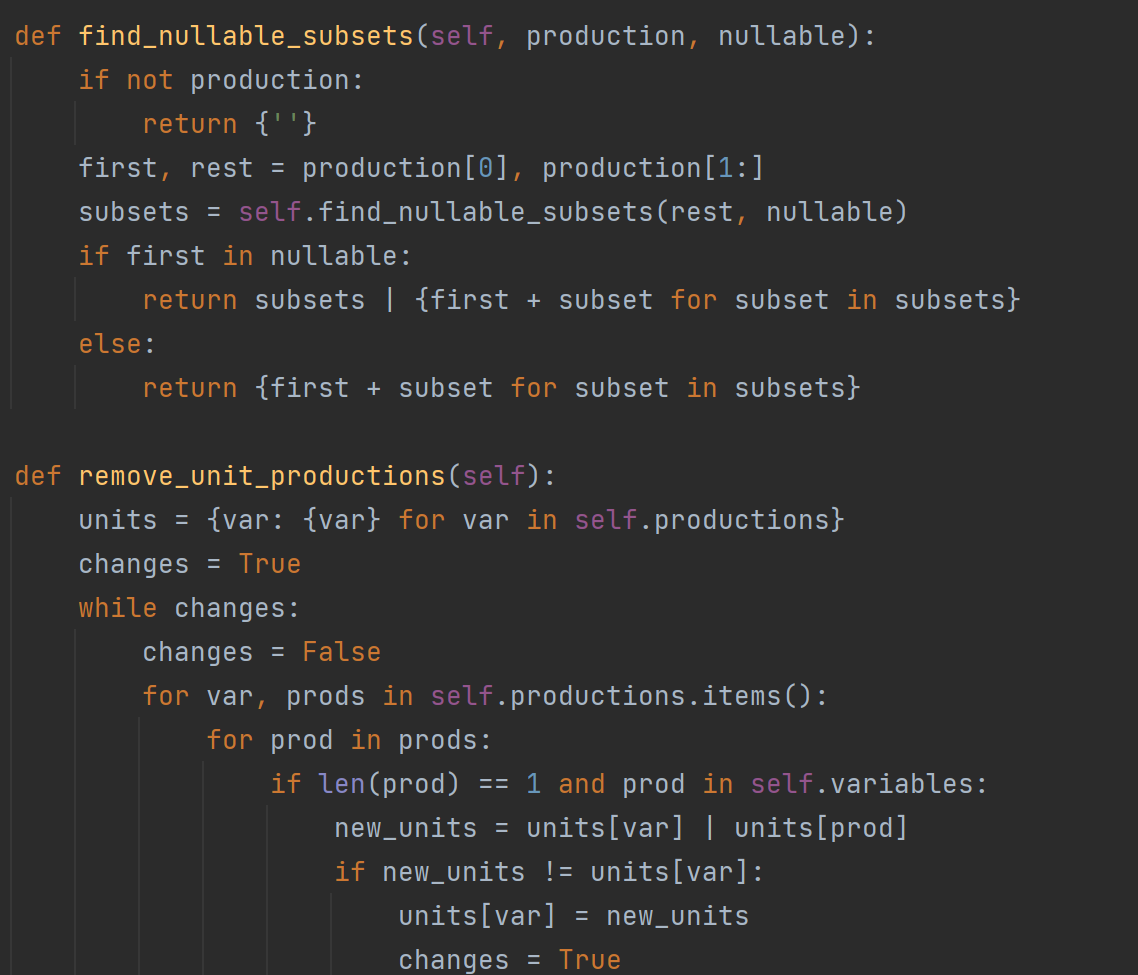
Coverter.py:

****

1. Class Definition:
   * The code defines a Python class named ChomskyNormalForm.
   * The purpose of this class is to convert grammar productions into Chomsky Normal Form (CNF).
2. Constructor (\_\_init\_\_ method):
   * The constructor initializes the class with the following parameters:
     + variables: A set of non-terminal variables (e.g., ‘S’, ‘A’, ‘B’, etc.).
     + terminals: A set of terminal symbols (e.g., ‘a’, ‘b’, etc.).
     + productions: A dictionary representing the production rules of the grammar.
     + start\_symbol: The start symbol of the grammar (e.g., ‘S’).
   * It also initializes a new set called new\_variables with the same values as variables.
3. Method: convert\_to\_cnf
   * This method is responsible for converting the given grammar to CNF.
   * It calls three other methods (remove\_null\_productions, remove\_unit\_productions, and convert\_to\_proper\_form) to perform the necessary transformations.
   * Finally, it returns the modified productions.
4. Method: remove\_null\_productions
   * This method identifies nullable variables (variables that can produce the empty string ‘ε’) in the grammar.
   * It iterates through the production rules and checks if ‘ε’ is present in any of the right-hand sides.
5. Incomplete Third Method:
   * The third method is not fully implemented in the provided code snippet.
   * It is expected to perform additional transformations to ensure the grammar is in CNF.

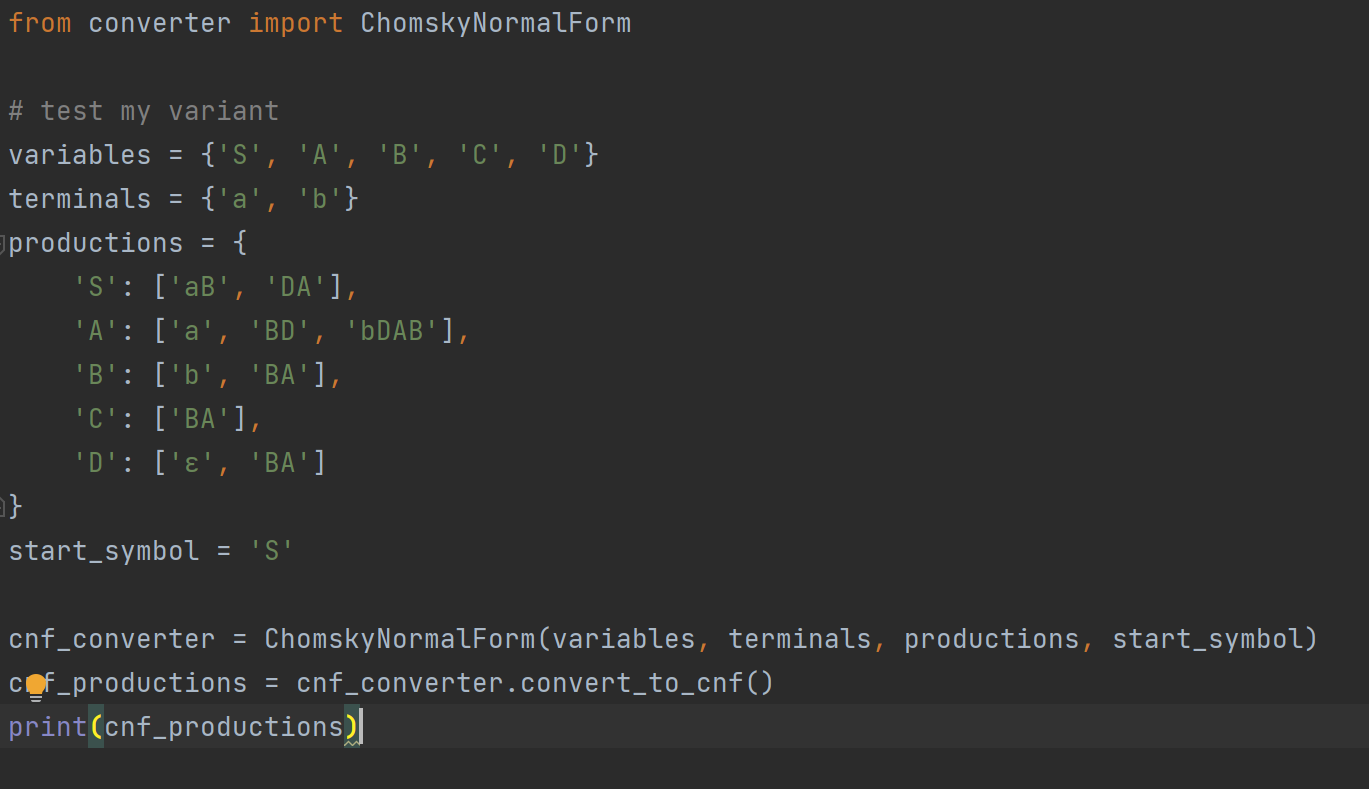
Overall, this code snippet provides the foundation for normalizing a given grammar into Chomsky Normal Form. The actual implementation of the missing methods would involve more complex logic to handle various cases and ensure the grammar meets CNF requirements.

****

****

****

Main.py:



Output:

