Documentation:

The Grammar class is designed to represent a formal grammar and provide functionality for reading thegrammar from a file, checking if it is context-free, and retrieving information about its structure.

The Grammar class has the following attributes:

filename: the name of the file from which the grammar is

readterminals: a list of terminals

nonterminals: a list of non-terminals

productions: a dictionary of

productionsstart_symbol: the start

symbol

EPSILON: the epsilon symbol string representation

The Grammar class has the following methods:

__init__(self, filename): the constructor of the Grammar class

read_grammar(self): reads the grammar from the file. It also checks if the symbols from theproductions are already defined terminals/non-terminals, depending on the context.

The file must have the following

structure:#Nonterminals

<nonterminal1> <nonterminal2>

...#Terminals

<terminal1> <terminal2>

...#Productions

<left1> ::= <right1> | <right2> | ...

<left2> ::= <right3> | <right4> | ...

...

#StartSymbol

<start_symbol>

terminal_representation(self): returns a string representation of the terminals

nonterminal_representation(self): returns a string representation of the non-

terminalsproductions_representation(self): returns a string representation of the

productions start_symbol_representation(self): returns a string representation of the

start symbol cfg_check(self): checks if the grammar is context-free

productions_for_nonterminal(self, nonterminal): returns a string representation of the productions for a given non-terminal

__str__(self): returns a string representation of the entire grammar

g1.txt:

GRAMMARS

---

1. Given the grammar $G = (N, \Sigma, P, S)$

$N = \{S, C\}$

$\Sigma = \{a, b\}$

$P: S \rightarrow ab \mid aCSb$

$C \rightarrow S \mid bSb$

$CS \rightarrow b,$

prove that $w = ab(ab^2)^2 \in L(G)$.

It has been extracted from a seminar. Another one will be used in the latter laboratories.