

from collections import defaultdict

*def* find(*grammar*):

    first = defaultdict(set)

*def* first\_rec(*sym*):

        if *sym* in first and first[*sym*]:

            return first[*sym*]

        if *sym* not in *grammar*:

            return {*sym*}

        result = set()

        for prods in *grammar*[*sym*]:

            if prods == "ε":

                result.add("ε")

            else:

                empty = True

                for i, part in enumerate(prods.split()):

                    part\_first = first\_rec(part)

                    result.update(part\_first - {"ε"})

                    if "ε" not in part\_first:

                        empty = False

                        break

                if empty:

                    result.add("ε")

        first[*sym*] = result

        return result

    for non\_terminal in *grammar*:

        first\_rec(non\_terminal)

    return {k: first[k] for k in *grammar*}

grammar = {

        "S": [" A "],

        "A": [" B C "],

        "B": [" x "," ε "],

        "C": [" y "," ε "],

    }

for x,y in grammar.items():

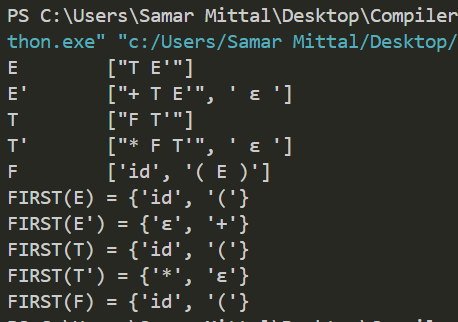
    print(x,"\t",y)

first\_sets = find(grammar)

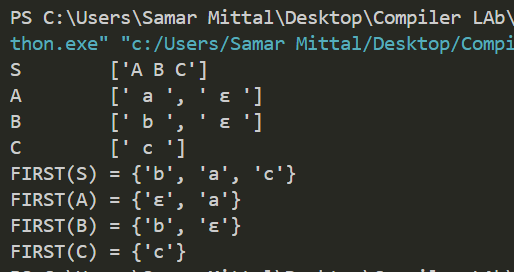
for variable, first\_set in first\_sets.items():

    print(*f*"FIRST({variable}) = {first\_set}")

**OUTPUT**



TEST CASE 1



TEST CASE 2

TEST CASE 3

TEST CASE 4

