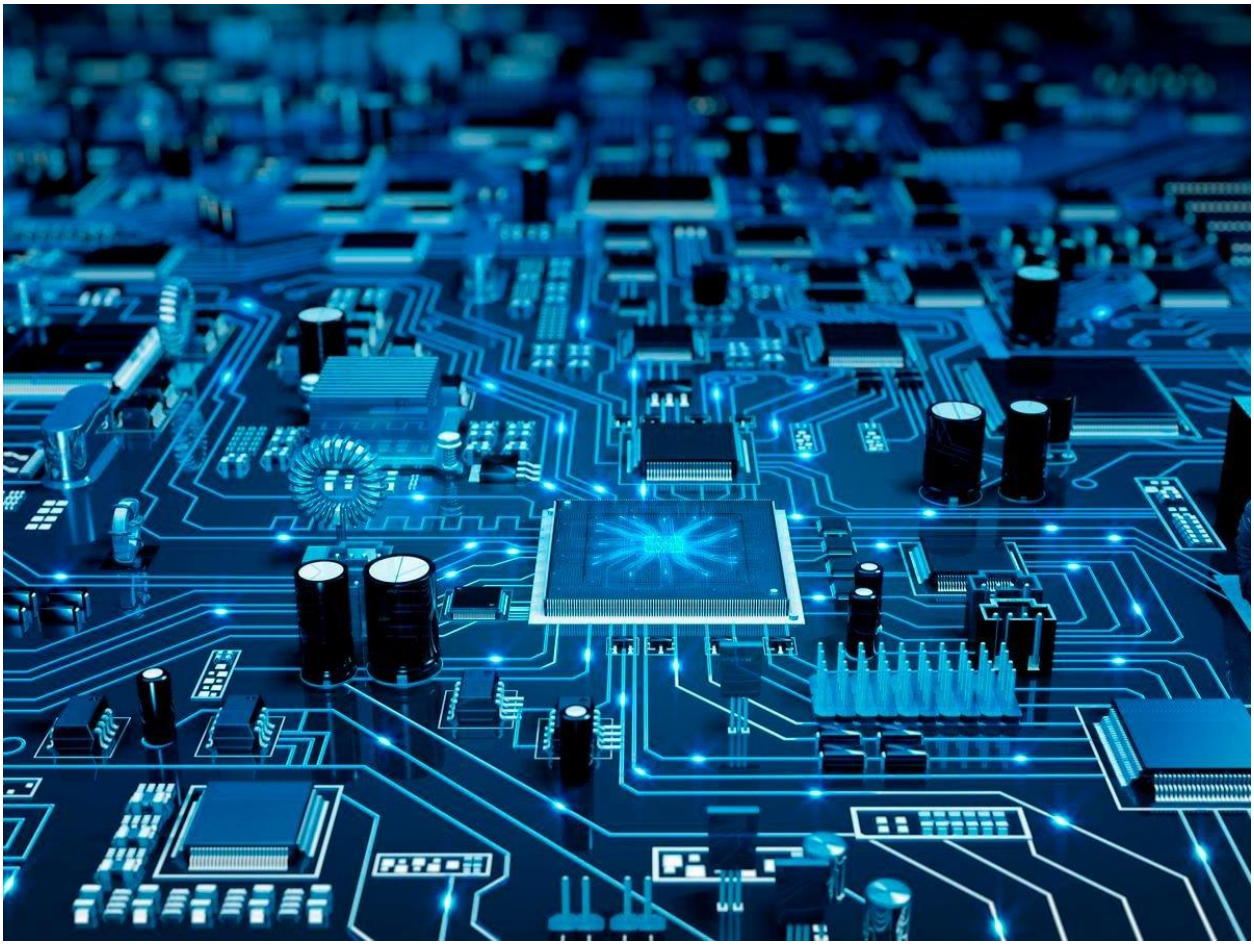

بسم الله الرحمن الرحيم

گزارش پروژه عملی دوم



نام و نام خانوادگی: روزبه غزوی

شماره دانشجویی: 98522274

گزارش سوال Grammar :

همانطور که در داک پروژه نیز ذکر شده است در این سوال باید برنامه ای نوشته شود که تشخیص دهد آیا رشته ورودی، توسط گرامر ورودی تولید می شود یا خیر.

برای این کار باید از الگوریتم Cocke–Younger–Kasami یا به اختصار CYK بهره بگیریم.

الگوریتم فوق تنها بر روی گرامر های به فرم چامسکی جواب می دهد پس ابتدا باید گرامر ورودی را به فرم CNF درآورده و سپس ازین الگوریتم بهره بگیریم.

تبدیل CFG به CNF : برای تبدیل گرامر CFG به فرم چامسکی باید مراحل زیر را طی کنیم:

Step 1. Eliminate start symbol from RHS.
If start symbol S is at the RHS of any production in the grammar, create a new production as:
S0->S
where S0 is the new start symbol.

Step 2. Eliminate null, unit and useless productions.
If CFG contains null, unit or useless production rules, eliminate them.

Step 3. Eliminate terminals from RHS if they exist with other terminals or non-terminals.
e.g,; production rule X->xY can be decomposed as:
X->ZY
Z->x

Step 4. Eliminate RHS with more than two non-terminals.
e.g,; production rule X->XYZ can be decomposed as:
X->PZ
P->XY

مراحل فوق در برنامه به وسیله توابع مختلف پیاده سازی شده اند:

```
#convert cfg to cnf
def convert_to_cnf():


#Add S0->S rule----------------------------------------------------------------------
def START(productions, variables):
```

```
#Remove rules containing both terms and variables, like A->Bc, replacing by A->BZ and Z->c----------TERM
def TERM(productions, variables):

#create a dictionary for all base production, like A->a
dictionary = setupDict(productions, variables, terms=K)

#Eliminate non unitry rules
def BIN(productions, variables):

#Delete non terminal rules
def DEL(productions):
```

گزارش سوال Pda Calculator :

```
#Create Stack
class Stack:

#constructor of Class
def __init__(self):

# If there is no top node, then that means there are nothing in the stack.
def isEmpty(self):

def peek(self):

#gharar dadan yek element dar stack
def push(self, value):

#bardashtan element rooye stack
def pop(self):

#tashkhise adad dar reshte vorodi
def isNumber(txt):
```

```
#peyda kardane amalgar ha be tartib
def findNextOpr(txt):

#peida kardane adade badi dar reshte vorodi
def getNextNumber(expr, pos):

#anjam 4 amale asli + exeption
def exeOpr(num1, opr, num2)

#tatbighe tedade parantez ha va mohasebe tamame maghadir dar stack
def _calculator(expr):
```

پایان