



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ ФИЛИАЛ  
ПЛОВДИВ

---

**КУРСОВА РАБОТА**  
**по ПСХА**

**НА**

**Веселин Руменов Станчев**

**Фак. № 614872**

**Специалност:** Компютърни системи и технологии  
**Образователно-квалификационна степен:** Магистър

**Изготвил:**  
**Веселин Руменов Станчев**

Пловдив 2022

## Съдържание

Лексер с Tensorflow.....	3
Граматика за краен автомат.....	4
Детерминиран краен автомат.....	4
Контекстно-свободна граматика.....	5
Синтактично дърво за statement на езика.....	5
Tensorflow code.....	6

## Лексер с Tensorflow

Моделът на проектиране на курсовата работа е open-source. Избрания език за курсовата работа е Python. Представява лексер, написан чрез модула за Python Tensorflow. Целта е да се покаже приложението на Python за целите на дизайн на компилатор за формален език за програмиране.

Като алтернативни модули на Tensorflow могат да бъдат посочени:

- ➔ Pytorch
- ➔ Keras

С помощта на знанията по дискретна математика са изградени:

- ➔ граматика за краен автомат
- ➔ детерминиран краен автомат който разпознава ключовите думи от езика
- ➔ контекстно-свободна граматика за езика
- ➔ синтактично дърво за statement на езика

Крайния автомат, граматиките и синтактичното дърво ще бъдат разгледани по-долу.

За изграденото синтактично дърво е създаден лексер.

Лексерът е реализиран чрез :

- описание на граф

### Граматика за краен автомат

FreedomLang={cl.wr, cl.show,int, double,  
do,while,bool,char,if,else,for,func, a-zA-Z}

S -> wr A

A -> show B

B -> int C

C -> double D

D -> do E

E -> while F

F -> bool T

T -> char R

R -> if Y

Y -> else L

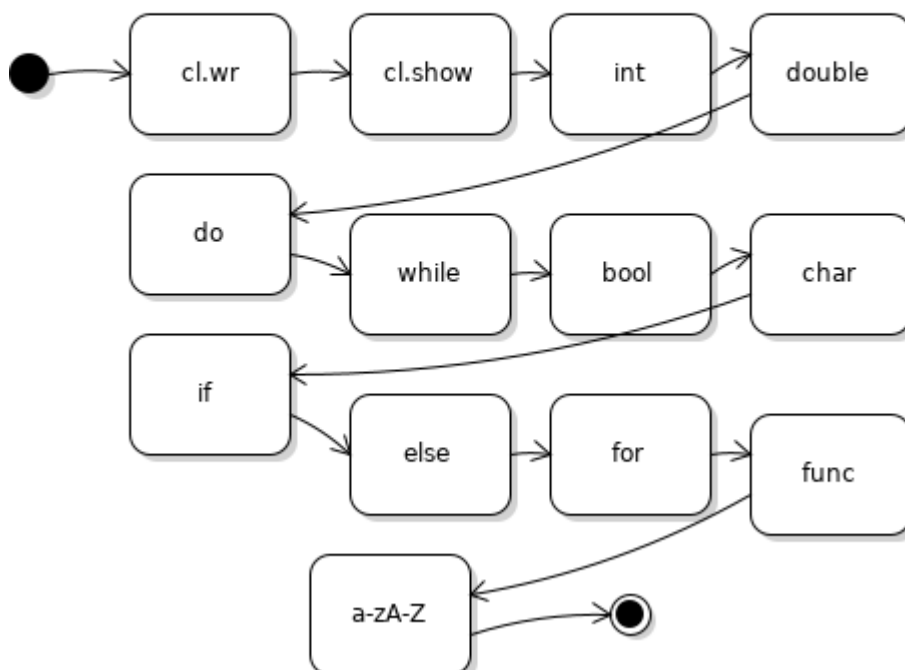
L -> for K

K -> func U

U -> a-zA-Z

### Детерминиран краен автомат

FreedomLang={cl.wr, cl.show,int, double,  
do,while,bool,char,if,else,for,func, a-zA-Z}



## Контекстно-свободна граматика

$::= \langle \text{identifier} | \text{type} | \text{variable} | \text{keywords} | \text{statements} \rangle$

$::= \langle 'a' \dots 'z' | 'A' \dots 'Z' \rangle$

$::= \langle \text{int} | \text{bool} | \text{char} | \text{double} | \text{func} \rangle$

$::= \langle \text{true} | \text{false} \rangle$

$::= \langle \text{type} | \text{identifier} \rangle$

$::= \langle \text{show} | \text{wr} | \text{if} | \text{else} | \text{do} | \text{while} \rangle$

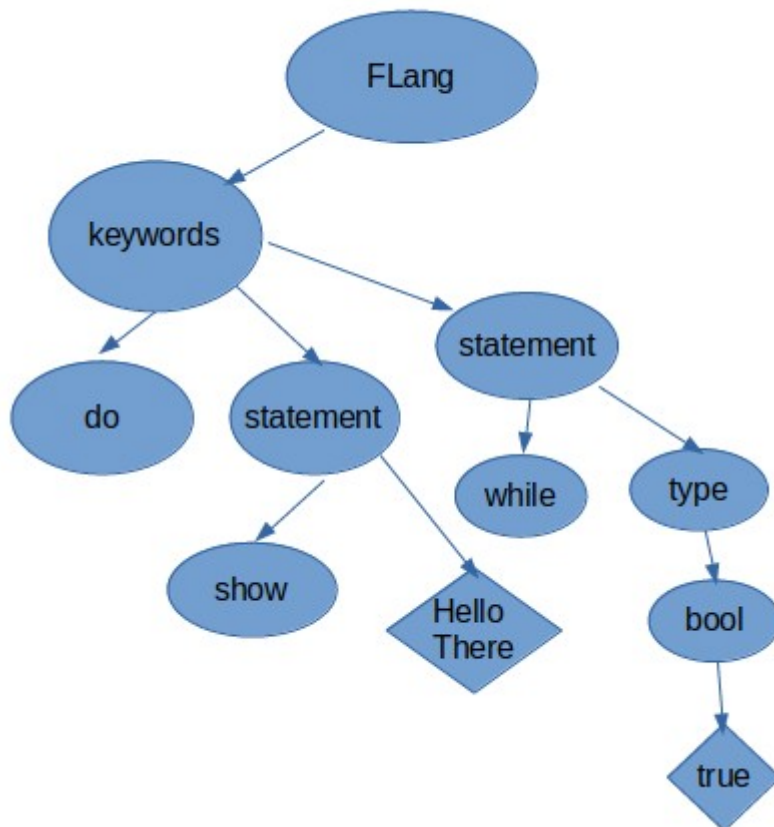
$::= \langle \text{keywords} | () \rangle$

statement на езика

`do { cl.show("Hello There") } while( true );`

### Синтактично дърво за statement на езика

`do { show("Hello There") } while( true );`



## Tensorflow code

```
import tensorflow as tf

tf.compat.v1.disable_eager_execution()
with tf.Graph().as_default():
    FLang=tf.constant(1,dtype=None,shape=None,name="FLang")
    kwords=tf.constant(2,dtype=None,shape=None,name='Keywords')
    do=tf.add(kwords,1,name="do")
    statement=tf.add(kwords,1,name="statement")
    statement1=tf.add(kwords,1,name="statement1")
    show=tf.add(statement,1,name="show")
    stringvalue=tf.add(statement,1, name="HelloThere")
    while1=tf.add(statement1,statement1,name="while")
    type1=tf.add(statement1,statement1,name="type")
    bool1=tf.add(type1,type1,name="bool")
    true1=tf.add(bool1,bool1,name="true")
    sess = tf.compat.v1.Session()
    outs = sess.run(true1.name)
    sess.close()
    print(true1.name)
```

