محمدجواد مهدی‌تبار

98522049

سوال اول)

ابتدا به documentation سایت انتلر رفتم (https://www.antlr.org/) . سپس مراحل گفته‌شده در این مستند را انجام دادم.

- فایل jar انتلر را دانلود کردم. در پی‌سی خود ذخیره کردم.
- Path فایل دانلود شده مرحله قبل را در environment variables اضافه کردم.
- سپس دو فایل bat در همان مکان قبلی درست کرده که به ترتیب دو فایل antlr4.bat , grun.bat بوده‌است .

Windows

1. Download https://www.antlr.org/download/antlr-4.9.2-complete.jar.
2. Add antlr4-complete.jar to CLASSPATH, either:
    1. Permanently: Using System Properties dialog > Environment variables > Create or append to CLASSPATH variable
    2. Temporarily, at command line:
       ```
       SET CLASSPATH=.;C:\Javalib\antlr4-complete.jar;%CLASSPATH%
       ```
3. Create batch commands for ANTLR Tool, TestRig in dir in PATH
   ```
   antlr4.bat: java org.antlr.v4.Tool %*
   grun.bat:   java org.antlr.v4.gui.TestRig %*
   ```

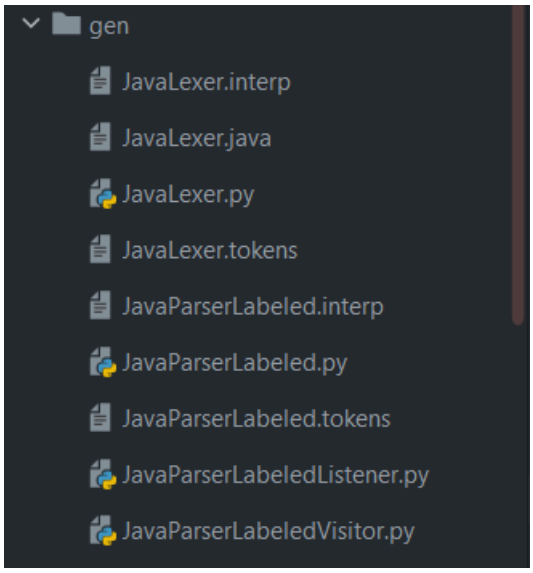
```
C:\Users\mjava>antlr4

C:\Users\mjava>java org.antlr.v4.Tool
ANTLR Parser Generator  Version 4.8
 -o ___              specify output directory where all output is generated
 -lib ___            specify location of grammars, tokens files
 -atn                generate rule augmented transition network diagrams
 -encoding ___       specify grammar file encoding; e.g., euc-jp
 -message-format ___ specify output style for messages in antlr, gnu, vs2005
 -long-messages      show exception details when available for errors and warnings
 -listener           generate parse tree listener (default)
 -no-listener        don't generate parse tree listener
 -visitor            generate parse tree visitor
 -no-visitor         don't generate parse tree visitor (default)
 -package ___        specify a package/namespace for the generated code
 -depend             generate file dependencies
 -D<option>=value    set/override a grammar-level option
 -Werror             treat warnings as errors
 -XdbgST             launch StringTemplate visualizer on generated code
 -XdbgSTWait         wait for STViz to close before continuing
 -Xforce-atn         use the ATN simulator for all predictions
 -Xlog               dump lots of logging info to antlr-timestamp.log
```

antlr4 successful output

```
C:\Users\mjava>grun

C:\Users\mjava>java org.antlr.v4.gui.TestRig
java org.antlr.v4.gui.TestRig GrammarName startRuleName
  [-tokens] [-tree] [-gui] [-ps file.ps] [-encoding encodingname]
  [-trace] [-diagnostics] [-SLL]
  [input-filename(s)]
Use startRuleName='tokens' if GrammarName is a lexer grammar.
Omitting input-filename makes rig read from stdin.
```

grun successful output

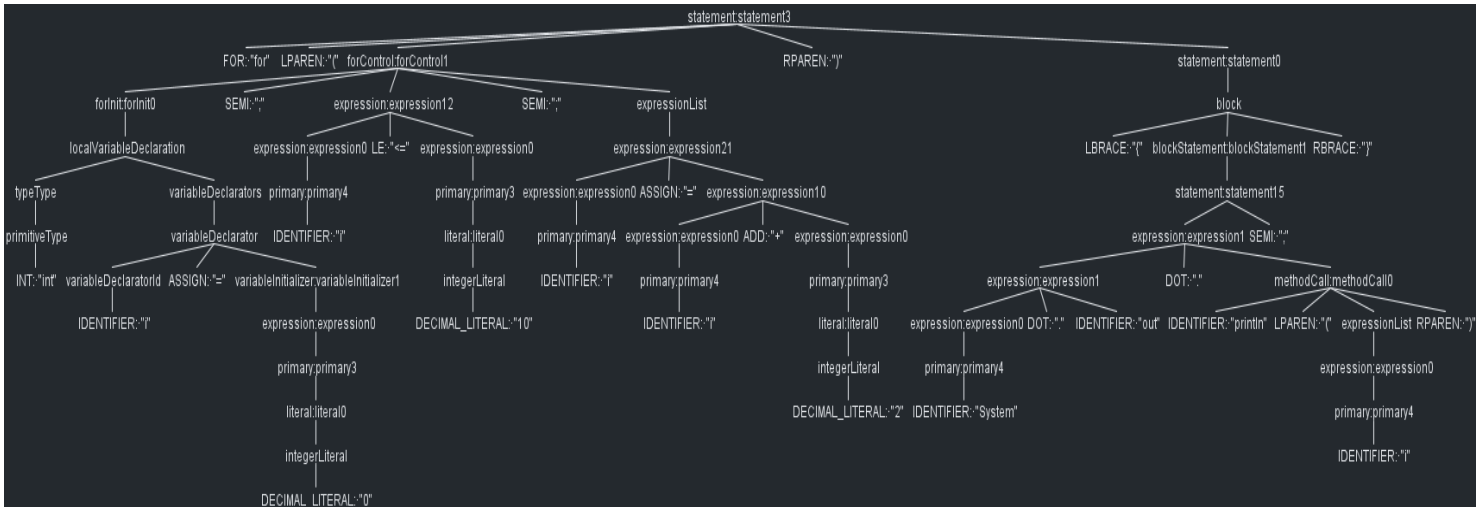
سپس فایل های Lexer , Parser , Listener , Visitor را از طریق Pycharm IDE به اینگونه که روی فایل g4 گرامر مورد نظر کلیک راست کرده و generate antlr recognizer را می‌زنیم.



سوال دوم)

به فایل javaParserLabeld.g4 مراجعه کرده و روی قانون statement ، تست انجام می‌دهیم.

و خروجی تولیدشده به شکل زیر می‌باشد



سوال سوم)

ابتدا به بررسی فایل جاوا می‌پردازیم.

```java
class Main {
    private int a1;
    public int a2;
    private int a3;
    public void hello1 () {
        system.out.println("hello");
    }
    public void hello2 () {
        system.out.println("hello");
    }
    public void hello3 () {
        system.out.println("hello");
    }
}
```

```java
class Student {
    private String id;
    private double grade;
    public void hello4 () {
        system.out.println("hello");
    }
    public void hello5 () {
        system.out.println("hello");
    }
    public void hello6 () {
        system.out.println("hello");
    }
}
```

برای فهمیدن متدها باید هر موقع که متد تعریف شد (enterMethodDeclaration) اسم تابع را به لیست تابع‌هایمان اضافه کنیم.

```python
def enterMethodDeclaration(self, ctx: JavaParserLabeled.MethodDeclarationContext):
    self.method_names.append(ctx.IDENTIFIER().getText())
```

method_names.json

```
{
  "package_name": "A.java",
  "methods": ["hello1", "hello2", "hello3", "hello4", "hello5", "hello6"]
}
```

سوال چهارم)

برای فهمیدن اینکه در هر کلاس چند attribute تعریف شده‌اند باید تابع enterFieldDeclaration را override کنیم. و برای فهمیدن اینکه در کدام کلاس تعریف شده‌اند باید آن node را traverse کنیم تا وقتی که به قسمت ClassDeclarationContext برسیم تا اسم class را بدست بیاریم.

```
def enterFieldDeclaration(self, ctx: JavaParserLabeled.FieldDeclarationContext):
    while type(ctx) != JavaParserLabeled.ClassDeclarationContext:
        ctx = ctx.parentCtx
    class_name = ctx.IDENTIFIER().getText()
    key_exists = self.class_attributes.get(class_name, "exists")
    if key_exists != "exists":
        self.class_attributes[class_name] += 1
    else:
        self.class_attributes[class_name] = 1
```

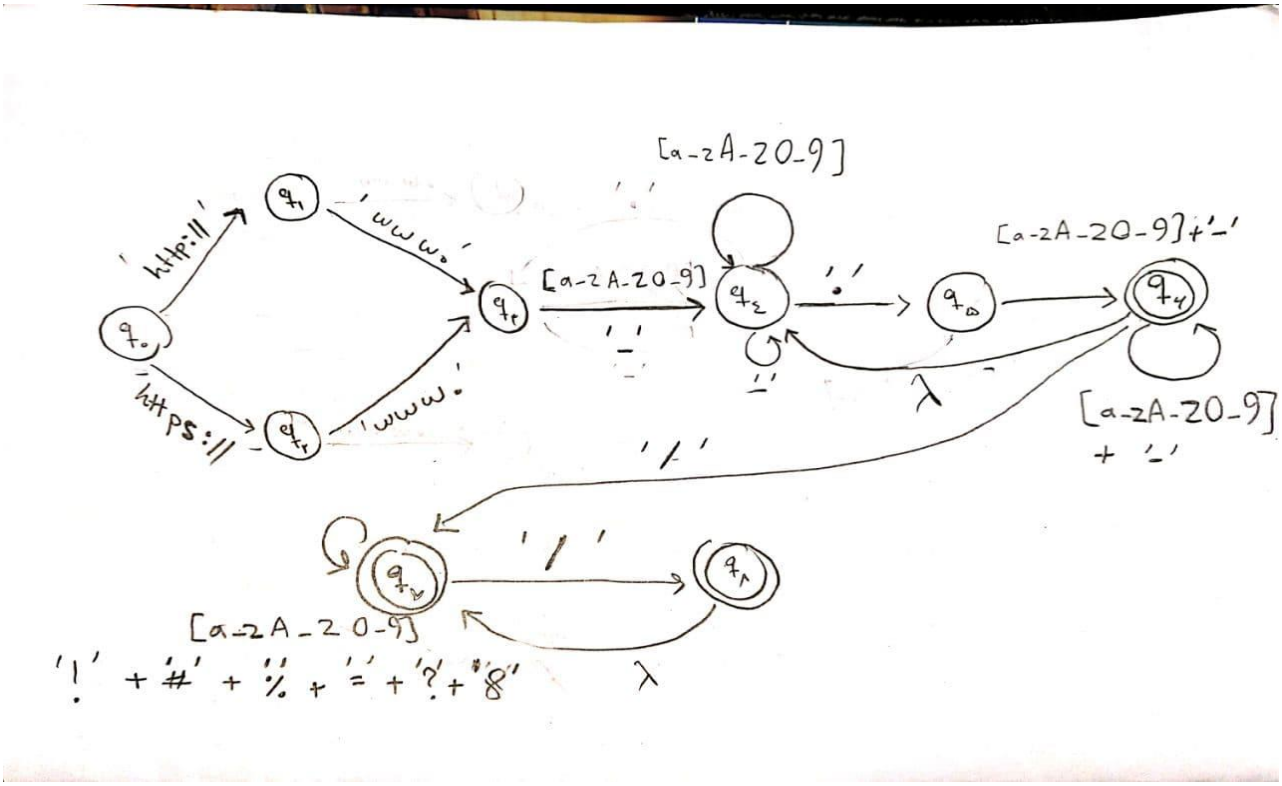
خروجی این سوال برابر عکس زیر است :

classAttributeCount.json

```
{
  "package_name": "A.java",
  "classAttributeCount": [("Main", 3), ("Student", 2)]
}
```

سوال پنجم)

اول NFA آن را می‌کشیم

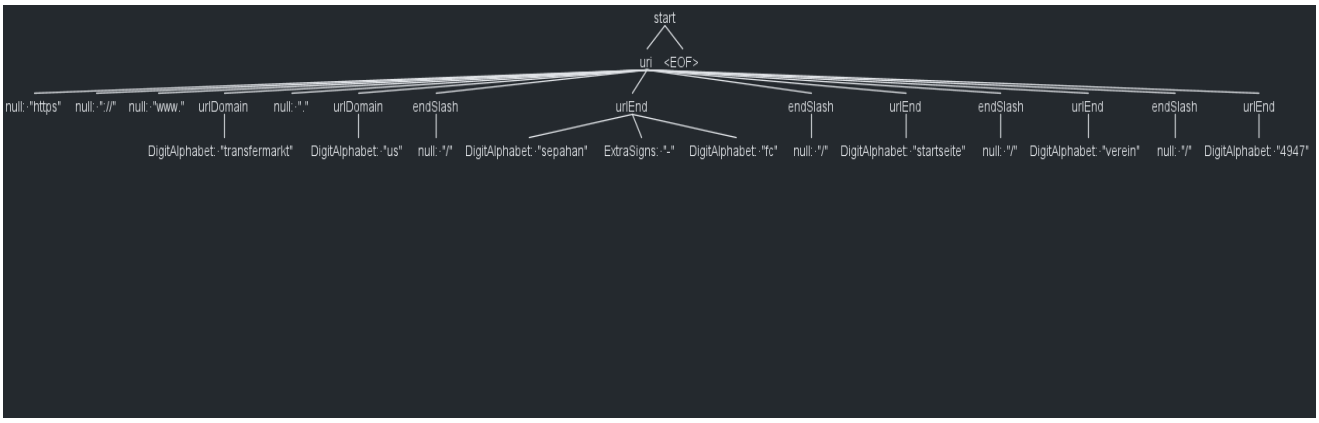



```
grammar urlGrammar;

start
   : uri EOF
   ;

uri
   :  (('http' | 'https') '://') ? 'www.'? urlDomain '.' urlDomain ('.' urlDomain)* (endSlash urlEnd)* '/' ?
   ;

urlDomain
    :
    DigitAlphabet
    ;
```

```
17    urlEnd
18        : (DigitAlphabet | ExtraSigns | endDot)+
19        ;
20
21    DigitAlphabet
22        : [a-zA-Z0-9]*
23        ;
24
25    ExtraSigns
26        :
27         ('#' | '$' | '%' | '@' | '&' | '=' | '!' | '-' | '?')*
28        ;
29
30    endDot :
31        '.'+
32        ;
33
34    endSlash
35        : '/';
```

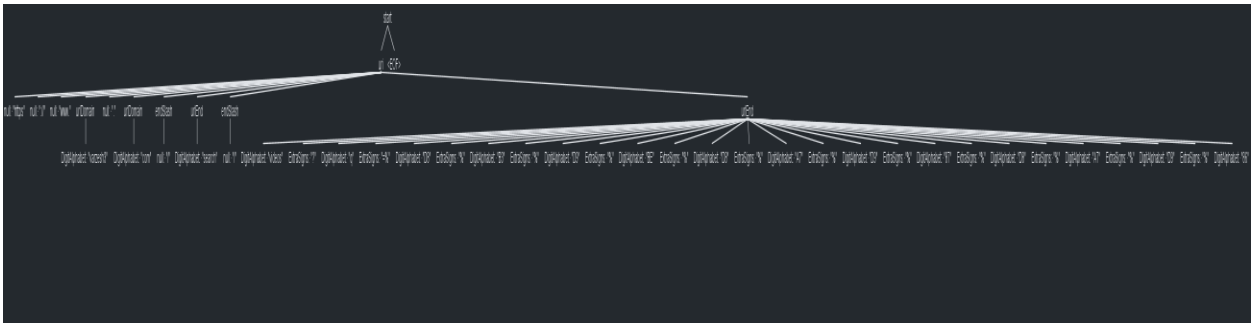
- ورودی اول

```
https://www.transfermarkt.us/sepahan-fc/startseite/verein/4947
```



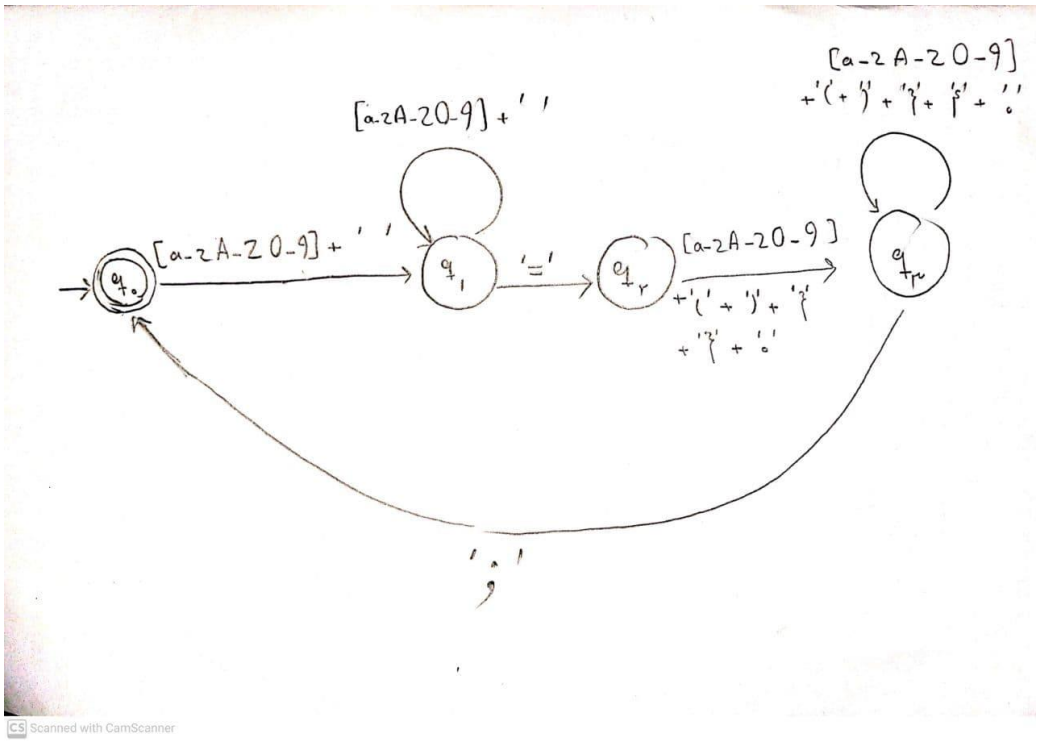
- ورودی دوم

```
https://www.varzesh3.com/search/videos?q=%D8%B3%D9%BE%D8%A7%D9%87%D8%A7%D9%86
```



1

سوال ششم)



[1] کیفیت عکس ها زیاد واضح نیس عکس ها را در فایل زیپ ضمیمه کرده‌ام

```
Signs :
    [(){}.]*
    ;

Alphabet :
    [a-zA-Z0-9]*
    ;

space :
    ' '
    ;

equalSign :
    '='
    ;

semiColon :
    ';'
```

```
grammar connectionString;

start :
    (prefix equalSign value semiColon)* EOF
    ;

prefix :
    (Alphabet | space)+
    ;

value :
    (Alphabet | Signs)+
    ;
```

ورودی امتحان شده :

```
Data Source=MSSQL1;Initial Catalog=AdventureWorks;Integrated Security=true;
```

خروجی تولید شده: