

Language Processing

-a37446 (Manisha Gautam)

-a37448 (Mausam Tiwari)

GeoJSON Parser

Regular Expressions:

* [1-9][0-9]* | 0 | [0-9]+\.[0-9] - Identifies the coordinates with decimal values

[\n\t\r] ;

[,\[\]\{\};():=] {printf("%s ",yytext); return yytext[0];}

Grammar :

suru :

| next

;

next : '{TYPE ':' FEATURECOLLECTION ',' FEATURES ':' '['features ']' }' ; {printf("\n");}

features:

| '{featurelist}' comma features {printf("\n");}

;

comma:

| ','

;

featurelist:

| arg {printf("\n");}

;

arg:TYPE ':' FEATURE comma featurelist

| GEOMETRY ':' geometry comma featurelist

| PROPERTIES ':' '{ property }' comma featurelist

;

property:

proparg ;

proparg:

| PROP0 ':' VALUE0 comma property

| PROP1 ':' prop1arg comma property

;

prop1arg: '{prop1 ':' prop1 '}'

| NUM ;

prop1: THIS

| THAT ;

geometry :point

| linestring

| polygon

;

polygon: '{ TYPE ':' POLYGON ', COORDINATES ':' lstt'}'; {printf("\n");}

linestring: '{ TYPE ':' LINESTRING ', COORDINATES ':' lstt'}'; {printf("\n");}

point : '{ TYPE ':' POINT ', COORDINATES ':' lst '}' ; {printf("\n");}

lst: '[' NUM ', NUM '];

```
lstt: '[' lstelem '];
```

```
lstelem: lst
```

```
    | lst ',' lstelem
```

```
    ;
```

```
%%
```

With the above regular expression and the grammar we achieved to validate the GeoJSON code in the text.txt file .