

ابتدا به کمک ابزار PGen گراف های ساختار زبان مدنظر را رسم کردیم. سپس جداول prettytable.prt و table.npt و csvtable.csv را بدست آورده. سپس با کمک گزینه export full parser، فایل های مدنظر یعنی Parser.java و Lexical.java و CodeGenerator.java را بدست آوردیم. جداول خروجی و فایل های جاوا را به علاوه ی فایل گراف هایمان (parser.pgs) در پوشه ی src کنار اسکریپت و پوشه ی تست قرار دادیم. برای این فاز نیازی به تغییرات کوچکی در فایل flex بودیم. (تغییرات داده شده در فایل با کامنت مشخص شده است) مهم ترین تغییرات، افزودن تابع nextToken() و import کردن Lexical است. اینکار را به این دلیل کردیم که با parser منطبق شود. در نهایت فایل Main.java که از ابتدا نوشته شده بود و قرار است توسط اسکریپت اجرا شود را در پوشه src اضافه میکنیم.

#### • تابع nextToken():

در این تابع پس از خواندن هر symbol یا token، عبارت مناسب را باتوجه به یال های گراف در خروجی به ما میدهد. مثلا در ابزار PGen وقتی میخواستیم در یالی از (،) استفاده کنیم، به ما ارور میداد و برای همین به اسم comma آن را قرار دادیم. در این تابع میگوییم اگر تایپ توکن دیده شده مثلا، بود و جزو operators بود، به ما comma برگردان. یا مثلا اگر کامنت یا فضای خالی (whitespace) بود، ارزش عبور کند (continue). اگر به آخر فایل رسیده باشیم، علامت \$ را خروجی میدهد.

```
public String nextToken() { //changed
    try {
        while (true) {
            Symbol current_symbol = next();

            if (current_symbol == null)
                return "$";

            if (current_symbol.getType().equals("ReservedKeywords"))
                return String.valueOf(current_symbol.getVal());

            if (current_symbol.getType().equals("WhiteSpace"))
                continue;

            if (current_symbol.getType().equals("Operators") && current_symbol.getVal().equals(","))
                return "comma";

            if (current_symbol.getType().equals("Operators") && !current_symbol.getVal().equals(","))
                return String.valueOf(current_symbol.getVal());

            if (current_symbol.getType().equals("Comment"))
                continue;

            return current_symbol.getType();
        }
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```