

مستندات پروژه نهایی درس اصول طراحی کامپایلر

اعضای گروه:

فاطمه صفاری

زینب همایونی

1397

لینک گیت هاب پروژه:

https://github.com/fatemesaffari8/compiler

روش انجام:

میخواهیم برای گرامر زیر کامپایلر طراحی کنیم.



روش کلی تحلیل نحوی و تولید کد، روش بالا به پایین است. می دانیم گرامری در تحلیل نحوی بالا به پایین قابل استفاده است که LL(1) باشد. لذا ابتدا باید گرامر را بررسی کنیم و در صورت نیاز به شکل LL(1) تبدیل کنیم.

با استفاده از فاکتورگیری چپ و حذف خود بازگشتی چپ گرامر را به LL(1) تبدیل می کنیم.

سپس با استفاده از نرم افزار antlr به تحلیل لغوی و تحلیل نحوی و تولید درخت میپردازیم.

Antlr از یک گرامر کد تولید می کند. نشان میدهد که چه چیز هایی برای یک زبان قابل قبول است و ساختار زبان را نیز نشان میدهد.

ابتدا گرامر را در یک فایل MiniJava.g4 با استایل مربوطه اضافه می کنیم. Antlr این گرامر را به دو بخش تقسیم میکند.

بخش اول lexer که شامل token هاست و بخش دوم parser که هرtoken را با قواعد گرامر مرتبط میکند.

با اجرای دستور زیر سه فایل MiniJavaLexer.py و MiniJavaParser.py و MiniJavaListener.py خواهیم داشت.

java -Xmx500M -cp <path to ANTLR complete JAR> org.antlr.v4.Tool -Dlanguage=Python2 MiniJava.g4

ما از کلاس MiniJavalistener.py ، extend میکنیم.

کلاس MiniJavaPrintListenerیک تابع تعریف میکند که اگر در پیمایش درخت به قاعده Goal رسیدیم فراخوانی میشود. متغیر ctx ، current contex است. که متد MainClass را داراست. و به همین روش تا انتها محاسبه می شود.

در تابع main ابتدا یک lexer درست کرده ایم. میتوان از FileStream  نیز استفاده کرد. وظیفه یlexer تولیدtoken است. خروجی این را به parser میدهیم. متد Goal را فراخوانی کرده ایم که سر ترم گرامر است. Parser یک درخت تولید میکند.

این درخت ورودی ast.py است که درخت خلاصه ی نحوی میدهد.