

## پروژه گرامر LL1

این پروژه نشان می‌دهد که چطور می‌توان برای یک گرامر داده‌شده، جدول تجزیه LL1 را ساخت.

فرآیند پروژه در سه مرحله اصلی انجام می‌شود:

### 1. اصلاح گرامر (`grammarEdit.py`)

این ماژول از پروژه، ابتدا در تابع `read_grammar` گرامر را از فایل می‌خواند و سپس به جداسازی قواعد گرامر می‌پردازد و در این فرایند خطاهای احتمالی در تایپ گرامر مثل نداشتن `=>` را پوشش می‌دهد و سپس قواعد مختلف را براساس | جدا می‌کند و در یک `set` ذخیره می‌کند.

سپس در تابع `remove_left_recursion` چپ گردی مستقیم گرامر را برطرف می‌کند  
این تابع دقیقاً بر همان اساس جدا سازی بخش `alphas = [ ] # parts after the leading A in Aα`

`(betas = [ ] # β (those not starting with A`  
عمل می‌کند

برای رفع فاکتور چپ ابتدا با تابع `longest_common_prefix` طولانی‌ترین بخش مشترک هر قانون را پیدا می‌کند و بعد در تابع `left_factoring` بر این اساس به شکلی ساده فاکتور چپ را دقیقاً بر همان اساس استفاده از یک نانترمینال اضافه رفع می‌کند.  
و در تابع `edit_grammar` در جمع‌بندی از تمام توابع قبلی استفاده می‌کند و نتیجه که گرامر تصحیح شده است را در صورت تمایل در فایل جدیدی نوشته و همینطور برای استفاده توابع دیگر به صورت دیکشنری نیز باز می‌گرداند.

### 2. FIRST و FOLLOW (`first\_follow.py`)

در ماژول `first_follow` فالو و فرست های گرامر ادیت شده را بدست می‌آورد در تابع `compute_first_follow` گرامر را اکه در ارگومان های ورودی تابع بوده را خوانده و براساس ترمینال ها و نان ترمینال ها جدا می‌کند سپس دقیقاً به همان روش کتاب فرست و فالو را محاسبه می‌کند برای فرست از چپ، سمت راست قانون را خوانده اگر ترمینال باشد خودش و اگر نانترمینال باشد به صورت بازگشتی فرستش را در مجموعه فرست ها می‌گذارد.

و در انتها مجموعه های محاسبه شده را در خروجی پرینت کرده و همینطور برای استفاده ماژول های دیگر باز می‌گرداند

### 3. جدول LL(1) (`ll1Com.py`)

در ماژول `ll1Com` ابتدا از اینکه گرامر `ll1` هست یا نه چک می‌شود اگر بود جدول `ll1` برای گرامر داده شده رسم می‌شود و سپس برای هر متغیر براساس فرست ها قوانین قرار داده می‌شود و در صورت اشتقاق `ε` که در ابتدای کد به شکل `"EPSILON = ε"` به صورت متغیر گلوبال تعریف شده است برای فالو های متغیر نیز قانون قرار داده می‌شود و سپس جدول ساخته شده را پرینت می‌کند. و در این فرایند هر جا به کانفلیکت در جدول بخورد پس گرامر بعد از اصلاح هم هنوز `ll1` نیست و خطا بر می‌گرداند.

تابع `main` هم همه این ماژول ها را به ترتیب صدا زده و خروجی هر مرحله را به ورودی مرحله بعد می‌رساند.