

هدف: طراحی تحلیلگر معنایی (Syntax Analyzer) از نوع پارسر LL(1) برای یک زبان برنامه نویسی شبه C

فاز دوم پروژه شامل پیاده سازی واحد تحلیلگر معنایی است که در اینجا پارسر نامیده می شود. وظیفه اصلی واحد پارسر تعیین صحت کد ورودی به زبان شبه C می باشد. این برنامه با دریافت یک فایل ورودی به زبان شبه C باید تعیین نماید که آیا کد وارد شده صحیح می باشد یا دارای خطاست. این فاز از پروژه شامل بخش های زیر می باشد.

- واحد تولید کننده جدول پارس (Parse Table Generator)
- پارسر LL(1) (LL(1) Parser)

جزئیات هر یک از دو قسمت فوق در ادامه توضیح داده می شود.

۱- واحد تولید کننده جدول پارس

این بخش از فاز دوم پروژه به منظور تولید جدول پارس با دریافت هر گرامر می باشد. این واحد باید شامل قسمت های زیر باشد که می توانند هر یک در قالب یک تابع پیاده سازی گردند:

- امکانی برای دریافت گرامر به عنوان ورودی. این امکان به دو صورت قابل پیاده سازی است:
 - از طریق یک فایل متنی که با توجه به قراردادهای شما برای نمایش متغیرها، ترمینال ها و قواعد گرامر تهیه شده است.
 - از طریق یک واسط گرافیکی (UI) یا واسط متنی (Command Prompt) برای دریافت قواعد گرامر با توجه به فرمت مورد قرارداد شما.
- ساختمان داده ای برای نگهداری قواعد گرامر پس از دریافت از بخش قبل. این ساختمان داده می تواند همان Right Hand Side Table (RHST) بیان شده در کلاس باشد. علاوه بر این می توانید برای ذخیره متغیرها و ترمینال ها از کلاس Token در فاز اول پروژه بهره بگیرید.
- تابع First() که مجموعه تمام first های تمام قواعد را محاسبه و در خروجی نمایش می دهد.
- تابع Follow() که مجموعه تمام follow های تمام متغیرها را محاسبه و در خروجی نمایش می دهد.
- تابع Predict() که مجموعه تمام predict های تمام قواعد را محاسبه و در خروجی نمایش می دهد.
- ساختمان داده ای برای جدول پارس که با توجه به مجموعه قواعد دریافتی تشکیل می شود. تایپ هر عنصر این جدول دو بعدی می تواند از نوع کلاس Token در فاز اول پروژه باشد.
- تابع ParseTableGenerator() که با استفاده از توابع فوق جدول پارس را پر کرده و در صورت LL(1) بودن پیغام مناسبی در خروجی نمایش می دهد. در صورت LL(1) نبودن نیز خانه هایی که دارای conflict می باشند در خروجی نمایش داده می شوند.
- تابعی برای نمایش محتویات جدول پارس به صورت مناسب. خواه گرامر LL(1) باشد خواه نباشد.

۲- پارسر LL(1)

این بخش از فاز دوم پروژه با دریافت یک گرامر توصیف کننده زبان شبه C و با استفاده از تابع `ParseTableGenerator()` اقدام به پر کردن جدول پارس می نماید. در این قسمت باید تابع پارسر بیان شده در کلاس درس پیاده سازی شده و همچنین با دریافت یک فایل ورودی و یا از طریق یک واسط کاربری کد ورودی را دریافت کرده و Syntax آن را چک می کند و در انتها پیغام مناسب را در خروجی چاپ می نماید. گرامر توصیف کننده زبان علاوه بر امکان تولید تابع `main()` باید شامل دستورالعمل های زیر باشد:

- دستورالعمل تعریف متغیر با امکان تعریف چند متغیر به صورت ساده، آرایه، اشاره گر تک پوینتری.
- دستورالعمل های `if`، `while`، `do while`، `Assignment`.

توجه ۱: دقت کنید که گرامر عبارات ریاضی (E) باید شامل عملگر تقسیم، تفریق و منهای تک عملوندی باشد. علاوه بر این باید شامل اعداد (توکن `num`) به صورت صحیح و اعشاری باشد.

توجه ۲: دقت کنید که انواع تایپ های موجود شامل `int`، `char`، `float` و `bool` می باشد.

توجه ۳: دقت کنید که گرامر مربوط به عبارات بولین تنها شامل عملگرهای `and` و `or` می باشد و عملگرهای مقایسه همانند کلاس باید در گرامر مربوط به دستور انتساب (`Assignment`) قرار داده شود.

نمره اضافی: امکان تعریف تابع و فراخوانی تابع

مهلت تحویل پروژه هفته اول آذر