



ساخت تحلیل‌گر نحوی (پارسر یا تجزیه‌کننده)

هدف این است که به کمک روش‌های بیان شده در کلاس درس، تجزیه‌کننده‌هایی برای گرامر ورودی بسازید، سپس رشته‌ای از توکن‌ها^۱ را به عنوان ورودی دریافت کرده و سعی کنید آن را به کمک هر یک از پارسرها تجزیه نمایید. در آخر تجزیه‌کننده‌ها را از لحاظ اندازه جدول تجزیه و کارایی (اجرا و خطایابی) با یکدیگر مقایسه نمایید.

ورودی:

- (۱) Grammar_X.txt: فایل متنی حاوی یک گرامر که به فرم نرمال BNF نوشته شده است. اولین متغیر، نماد شروع گرامر است و نماد شروع افزوده ($S' \rightarrow S$) اضافه نشده است.
- (۲) Test_X.txt: فایل متنی حاوی رشته‌ای از توکن‌ها (ترمینال‌ها)ی گرامر است. فرض کنید این فایل از یک تحلیل‌گر لغوی فعال بدست آمده است.

خروجی:

تعیین این که کدام یک از تجزیه‌کننده‌ها را می‌توان برای گرامر ورودی ساخت. در صورتی که بتوانید تجزیه‌کننده‌ای را برای گرامر بسازید جدول تجزیه را به صورت یک ماتریس ساخته و نمایش دهید و در صورتی که ساخت تجزیه‌کننده‌ای ممکن نباشد علت را بیان کنید.

شرح پروژه:

کدی بنویسید که تلاش کند در صورت امکان، تجزیه‌کننده بالا به پایین (LL(1) را برای گرامر ورودی که در فایل Grammar_X.txt آمده ایجاد کند. سپس برای ساخت تجزیه‌کننده‌های پایین به بالا به ترتیب از ضعیف به قوی، LR(0)، SLR(1)، LALR(1) و LR(1) تلاش کند. هر کدام که امکان‌پذیر بود را ایجاد و برای هر کدام که ممکن نبود دلیل ناموفق بودن را مشخص نماید.

پس از ایجاد تجزیه‌کننده‌ها، رشته ورودی را از فایل Test_X.txt خوانده و تلاش کند آن را به کمک همه پارسرهای ایجاد شده، تجزیه کند. در این مرحله، باید برای هر تجزیه‌کننده لیست قوانین مورد استفاده را به ترتیب در خروجی نمایش دهد. توجه داشته باشید که رشته ورودی ممکن است توأم با خطا باشد، که در این صورت باید یکی یا ترکیبی از روش‌های برخورد با خطا را مورد آزمایش قرار دهید و خطای ورودی را رفع نمایید.

تجزیه‌کننده‌های ایجاد شده را از نظر اندازه جدول تجزیه، تعداد قوانین مورد استفاده برای تجزیه رشته ورودی و همچنین سرعت کشف خطا (در چندمین حرکت خطا یافت می‌شود) مقایسه نمایید.

^۱ از آنجا که شما تحلیل‌گر لغوی را ننوشته‌اید فرض می‌شود رشته‌ای که برای تحلیل نحوی به تحلیل‌گر وارد می‌شود خود حاوی توکن‌هاست

برای آزمودن پروژه می‌توانید یک یا چند فایل حاوی گرامر و رشته توکن‌ها را خودتان ایجاد کنید یا از مثال‌های کلاسی استفاده نمایید.

نمره اضافی:

در صورت انجام موارد فوق و نمایش نتایج به صورت واضح، نمره پروژه به شما تعلق خواهد گرفت. تعلق نمره اضافی در صورتی ممکن خواهد بود که:

- ۱) پس از کشف علت ناموفق بودن ساخت تجزیه‌کننده آن را رفع نمایید (۵۰ درصد)
 - ۲) تا حد امکان درصد کمتری از توکن‌ها را برای رفع خطا حذف کنید (۳۰ درصد)
 - ۳) در حین تست رشته ورودی برای تجزیه‌کننده‌ها تغییرات محتوای پشته را نشان دهید (۲۰ درصد)
- مقادیر درون پرانتز، نشان می‌دهد که هر مورد چند درصد از نمره اضافی را شامل می‌شود.

نکات مهم

۱. پروژه به صورت گروهی و در گروه‌های ۲ نفره انجام می‌شود. و در صورت افزایش تعداد نفرات، به ازای هر نفر اضافه، ۲۰ درصد از نمره کل اعضا کاهش می‌یابد. مثلاً نمره هر یک از افراد در یک گروه ۳ نفره ۰,۸ برابر نمره‌ای است که به پروژه گروه تعلق گرفته است.
۲. از زمان آپلود فایل، یک هفته فرصت دارید تا نام اعضای گروه را اعلام کنید بدیهی است که پس از اعلام، ترکیب گروه قابل تغییر نیست.
۳. در صورتی که پروژه برخی از گروه‌ها یا بخش‌هایی از آن مشابه باشند همه گروه‌ها نمره پروژه را از دست خواهند داد.
۴. زبان برنامه‌نویسی آزاد است و می‌توانید با هر زبان که مسلط هستید کار کنید.
۵. هر گروه یک ارائه شفاهی نیز با حضور تمام اعضا خواهد داشت که زمان آن اعلام خواهد شد. در زمان ارائه حضوری، همراه داشتن ماشین‌نی که پروژه در آن قابل اجرا باشد الزامی است. با وجود این که تقسیم کار صورت می‌گیرد ولی همه اعضای گروه باید بر کل پروژه مسلط باشند.
۶. لازم است همزمان با پیشرفت پروژه مستندی نیز برای آن آماده کنید. به گزارش خود مواردی مانند نقش افراد، نحوه اجرا، اصلاحات، تغییرات و همچنین مواردی که به صورت داوطلبانه به گرامر اضافه شده را بیفزایید (البته خیلی طولانی نباشد).
۷. تمام فایل‌های پروژه به همراه گزارش به صورت یک فایل فشرده، در زمان مشخص شده (یا قبل از ارائه حضوری) آپلود گردد. عنوان فایل باید به صورت (Compiler_3981_Lastname of group members) باشد.

موفق باشید.