



تکلیف کامپیوتری اصول طراحی کامپایلر ها

مهلت ارسال: 9 خرداد 1399

استاد: دکتر جابری پور

در این تکلیف قصد داریم بخش کوچکی از فعالیت های یک پارسر **top down** از نوع **LL(1)** را پیاده سازی کنیم.

بدین منظور برنامه ی شما باید یک فایل گرامر را به عنوان ورودی دریافت کند و از روی آن **گرامر** با محاسبه ی مجموعه های **first** و **follow** برای هر ناپایانه، با استفاده از جدول گرامر های ورودی **RHST** یا همان **Right Hand Side Table** و در نهایت با محاسبه ی **predict** ها برای هر ناپایانه و به ازای تمامی پایانه های موجود، **parse table** را به وجود آورد و آن را چاپ کند. توجه داشته باشید که روندی که این برنامه طی می کند باید عینا مشابه روند طی شده توسط یک پارسر **LL(1)** که در کلاس درس توضیحات آن داده شده است.

❖ نکات :

- برنامه ی شما باید شامل تابع هایی برای محاسبه ی مجموعه ی **first** به ازای یک هر ناپایانه ورودی باشد.
- برنامه ی شما همچنین باید شامل تابعی به منظور محاسبه ی مجموعه ی **follow** به ازای ورودی گرفتن یکی از ناپایانه ها باشد.
- دقت کنید شماره ی هر سطر از قواعد در گرامر از 0 شروع می شود.
- برنامه ی شما باید تابع **is_nullable(word w)** را داشته باشد که یک پایانه یا ناپایانه را به عنوان ورودی بگیرد و **nullable** بودن آن را تشخیص دهد.
- مجموعه های **first** و **follow** پس از محاسبه برای یک ناپایانه توسط تابع های بالا باید قابلیت چاپ کردن را نیز داشته باشد. (به عنوان مثال برنامه ی شما شامل تابع **print_follows()** باشد که در هر سطر از خروجی برای هر کدام از ناپایانه ها، مجموعه ی **follow** آن را چاپ کند)
- در گرامر نوشته شده از کاراکتر **#** به منظور نشان دادن مفهوم **nullable** بودن (نماد های افسیلون یا لامبدا) استفاده شود.
- در گرامر نوشته شده باید **ناپایانه ها** **تماما با حروف بزرگ انگلیسی** و **پایانه ها** **با حروف کوچک انگلیسی** مشخص شوند.
- پیشنهاد می شود برای کدنویسی تمیز تر از کلاس **word** برای متغیر های گرامر استفاده کنید که شامل ویژگی **wordType** باشد که این ویژگی می تواند یکی از انواع پایانه یا ناپایانه باشد.

- در نهایت برنامه ی شما باید یک تابع به نام run داشته باشد که با ورودی گرفتن آدرس فایل گرامر و با استفاده از توابع تعریف شده در بالا parseTable را حساب کند و آن را در خروجی چاپ کند.
- توجه کنید که متغیر آغازین گرامر شما باید حتما حرف S باشد و برای این متغیر در مجموعه ی follow آن کاراکتر \$ که به معنای پایان برنامه است قرار داده شود. (برای نشان دادن به پایان رسیدن یک کد برنامه که طبق گرامر زبان شما نوشته شده است باید یک کاراکتر به عنوان مشخص کننده ی پایان فایل باشد که در اینجا ما آن را \$ در نظر گرفته ایم)
- همانطور که در بالا گفته شد، رفتار این برنامه باید مطابق پارسر LL(1) باشد. به همین دلیل به عنوان مثال برای محاسبه ی predict های هر ترکیب ناپایانه و پایانه و تشکیل parseTable باید از توابع first و follow و is_nullable استفاده شود. به دلیل این پیوستگی بالا بین اجزای برنامه حتما هر تابع را به تنهایی به طور کامل مورد ارزیابی قرار دهید چرا که عملکرد نادرست یکی از تابع ها می تواند منجر به خروجی نادرست برنامه در خیلی از موارد شود.

در زیر نمونه ای از گرامر ورودی آورده شده است. دقت کنید که گرامر شما هم به همین فرمت نوشته و از فایل خوانده شود :

```
S : EXP
EXP : TERM EXP'
EXP' : + TERM EXP'
EXP' : - TERM EXP'
EXP' : #
TERM : FACTOR TERM'
TERM' : * FACTOR TERM'
TERM' : / FACTOR TERM'
TERM' : #
FACTOR : ID
ID : id ID'
ID' : ++
ID' : --
ID' : #
ID : -- id
ID : ++ id
FACTOR : num
FACTOR : ( EXP )
```

در ادامه نیز تصویر خروجی محاسبه ی follow های گرامر بالا در ترمینال و پس از اجرای برنامه به عنوان نمونه آمده است :

```
S : $
EXP : $, )
EXP' : $, )
TERM : +, -, $, )
TERM' : +, -, $, )
FACTOR : *, /, +, -, $, )
ID : *, /, +, -, $, )
ID' : *, /, +, -, $, )
```

همچنین در زیر تصویر **parseTable** محاسبه شده برای گرامر فوق را می بیند که در نهایت در ترمینال برنامه باید چاپ شود :

parse table

	+	-	*	/	id	++	--	num	()	\$
S	[1]	[1]	[1]	[1]	[0]	[0]	[0]	[0]	[0]	[1]	[1]
EXP	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[1]
EXP'	[2]	[3]	[1]	[1]	[1]	[1]	[1]	[1]	[1]	[4]	[4]
TERM	[1]	[1]	[1]	[1]	[5]	[5]	[5]	[5]	[5]	[1]	[1]
TERM'	[8]	[8]	[6]	[7]	[1]	[1]	[1]	[1]	[1]	[8]	[8]
FACTOR	[1]	[1]	[1]	[1]	[9]	[9]	[9]	[16]	[17]	[1]	[1]
ID	[1]	[1]	[1]	[1]	[10]	[15]	[14]	[1]	[1]	[1]	[1]
ID'	[13]	[13]	[13]	[13]	[1]	[11]	[12]	[1]	[1]	[13]	[13]

نکات تحویل:

- تحویل این تمرین بر روی سامانه ی درس افزار خواهد بود اما تمرین تحویل حضوری و یا مجازی (مثلاً با اسکایپ) خواهد داشت.
- برای انجام تکلیف می‌توانید در قالب گروه های دو نفره اقدام به پیاده سازی و ارسال آن کنید.
- سعی کنید کد برنامه را به صورت خوانا و با رعایت قواعد برنامه نویسی بنویسید.
- در صورت مشاهده ی هر گونه تقلب، اقتباس و ... با افراد خاطی برخورد خواهد شد !
- زمان تمرین تمدید نمی‌شود و امکان ارسال با تأخیر تکلیف وجود ندارد.
- به هیچ عنوان انجام این تکلیف را به روز های آخر موکول نکنید.
- سوالات خود را از طریق درس افزار مطرح کنید تا در صورت برطرف شدن ابهام همه بتوانند از آن استفاده کنند.

با آرزوی موفقیت

گروه حل تمرین