

تعریف فاز دوم پروژه اصول طراحی کامپایلر

دستان پرنوازش بهار ، طبیعت خفته را از خواب بیدار می سازد و زمین و درخت ، رازهای رنگارنگ و عطراگین خویش را نثار نگاه ما می کنند. در سال جدید ، سبزی ، شادی ، کامیابی ، بهره وری ، اثربخشی فعالیت ها و بهروزیتان را از درگاه ایزد منان آرزومندیم.

قبل از هر چیز، فرا رسیدن سال نو را خدمت همه دوستان گرامی تبریک عرض می کنیم.

همان گونه که در فاز اول پروژه مشاهده کردید، هدف از قسمت اول طراحی یک تحلیل گر لغوی^۱ برای کامپایلر بود. حال قصد داریم در فاز دوم به طراحی تحلیل گر نحوی بپردازیم. در ابتدا، هدف از انجام این فاز و سپس نحوه پیاده سازی و چندین نکته در رابطه با آن را ذکر می کنیم.

هدف تحلیل گر لغوی

زمانی که یک برنامه توسط کامپایلر خوانده می شود، باید علاوه بر این که کلمات نوشته شده درست باشند، لازم است تا برنامه از نظر نحوی نیز صحیح باشد. وظیفه کنترل درستی گرامر برنامه، بر عهده ی قسمت تحلیل گر نحوی است. این کار با تشکیل درخت مربوط انجام می شود. ساخت درخت همان گونه که در کلاس درس ذکر شده است، توسط جدولی موسوم به جدول shift-reduce می باشد که در مراحل مختلف ساخت آن ممکن است به مشکلاتی از قبیل conflict برخورد کند. این مشکلات می تواند ناشی از عواملی از قبیل مبهم بودن گرامر باشد که یکی از وظایف شما در این فاز، رفع ابهام های گرامر داده شده است.

نحوه پیاده سازی تحلیل گر نحوی

در این قسمت، از tokentokenهایی که در بخش قبلی شناسایی کرده بودیم، استفاده خواهیم کرد. به این صورت که، تحلیل گر لغوی این tokenها را تشخیص می دهد و به تحلیل گر نحوی اعلام می کند و در این قسمت، برنامه با توجه به گرامر کنترل می شود. نحوه نوشتن دستورات، مشابه قسمت قبلی می باشد. یعنی، بخش ها در این قسمت نیز با %% از هم جدا می شوند و ترتیب قسمت ها چه در زبان C و چه در زبان جاوا همانند فاز اول است.

در ابتدا می بایست تمامی دستورات چاپ را که در فاز اول نوشته بودید، به صورت زیر تغییر دهید:

(در زبان جاوا)

```
{INT_KW} {System.out.println("INT_KW");} → {INT_KW} {return INT_KW;}
```

(در زبان C)

```
"program" {printf("PROGRAM_KW");} → "program" {return PROGRAM_KW;}
```

در واقع، به جای این که تحلیل گر لغوی، نوع کلمه خوانده شده را چاپ کند، آن را به تحلیل گر نحوی اعلام می کند.

حال در فاز دوم، در قسمت تعاریف^۲ باید کلیه مقادیری را که تحلیل گر لغوی به بخش نحوی می دهد را تعریف کنید. به

عنوان مثال:

```
%token INT_KW PROGRAM_KW
```

^۱ Lexical analyzer

^۲ البته لازم به ذکر است که در عمل هیچ درختی ساخته نمی شود و تنها با استفاده از همین جدول، بخش نحوی کار خود را انجام می دهد.

^۳ Declaration

تعریف فاز دوم پروژه اصول طراحی کامپایلر

پس از تعریف کلیه `token`ها، به سراغ قسمت قوانین^۱ می‌رویم. در این بخش، مشابه فاز اول عمل می‌کنید؛ با این تفاوت که باید قوانین گرامر را وارد کنید. به طور مثال، اگر جمله اول گرامر به صورت زیر باشد:

```
program → declaration_list
```

تبدیل به جمله زیر در قسمت قوانین خواهد شد:

```
program : declaration_list {System.out.println("Rule 1: program -> declaration_list");}
```

و به همین ترتیب، تمامی قواعد گرامر را در این قسمت وارد می‌کنیم. بدیهی است که برای کلمات کلیدی^۲ و `token`ها از کلماتی که از تحلیل‌گر لغوی به بخش نحوی داده می‌شوند، استفاده می‌کنیم.

قسمت **user code** نیز همانند فاز قبلی می‌باشد.

پس از وارد کردن تمامی قواعد، با استفاده از ابزارهای مربوطه و فایل‌های راهنما^۳ موردنظر برای تحلیل‌گر لغوی تولید می‌شود. تنها نکته قابل ذکر باقیمانده مربوط به رفع ابهام‌هاست که می‌توان آن‌ها را به دو روش رفع کرد:

(۱) تغییر گرامر

(۲) استفاده از امکانات یک^۴

که برخی از موارد با استفاده از مورد دوم قابل حل هستند ولی در بعضی از آن‌ها لازم است تا خود گرامر را تغییر دهید. به عنوان مثال، یکی از `conflict`هایی که معمولاً به آن برمی‌خورند، اولویت‌های مربوط به عملگرهاست که می‌توان با دستورات زیر در پایان قسمت تعاریف، به آن‌ها اولویت داد:

```
% left MINUS_KW.PLUS_KW
```

```
%left MULTIPLY_KW.DIVIDE_KW
```

دستور `left` به منظور `left associative` بودن عملگرهاست^۵ و نکته دیگر این‌که هرچه عملی پایین‌تر قرار گیرد، اولویت بیشتری دارد. علاوه بر عنوان `left`، عناوین `right` و `nonassoc` نیز برای `right associative` و `non associative` بودن نیز موجود است.

برخی نکات لازم

- (۱) توصیه می‌شود تمامی خطاها و حتی هشدارها در این فاز رفع شوند تا در قسمت‌های بعدی به مشکل نخورید.
- (۲) می‌توانید برای مشاهده خطاها و موارد ابهام از پسوند `verbose` -- در ابزار `bison` استفاده کنید.
- (۳) اگر تا این مرحله کامنت را در تحلیل‌گر لغوی خود وارد نکرده‌اید، آن را با توجه به نکات موجود در گرامر در تحلیل‌گر لغوی خود درج کنید.^۶

^۱ Rules

^۲ Keyword

^۳ برای کامپایل و تبدیل فایل‌های نوشته شده از ابزارهایی که به پیوست خدمتتان تقدیم گردیده، استفاده کنید و با توجه به فایل‌های **How to** عمل نمایید.

^۴ Yacc

^۵ در کلاس درس به طور کامل توضیح داده شده است.

^۶ کامنت با // شروع می‌شود و تا پایان خط مربوطه ادامه می‌یابد.

تعریف فاز دوم پروژه اصول طراحی کامپایلر

(۴) برای اجرا و تست تحلیل‌گر نحوی، می‌بایست همانند فاز قبلی عمل کنید؛ با این تفاوت که این بار باید تحلیل‌گر لغوی به عنوان ورودی تحلیل‌گر باشد.

❖ کلاس توجیهی و رفع اشکال برای فاز دوم، هفته اول بعد از عید خواهد بود. لذا برای هماهنگی خواهشمندیم در نظرسنجی مربوط به زمان کلاس در مودل شرکت نمایید.

❖ هرگونه سوال و نظر خود را می‌توانید با رایانامه‌های farzin.heidary@gmail.com و saeid.dadkhah@live.com در میان بگذارید.