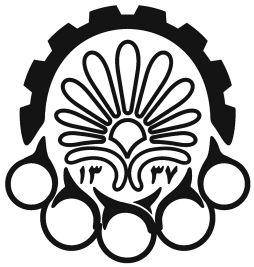

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه اول و دوم

پیاده‌سازی تحلیل‌گر لغوی و نحوی

امیر حقیقتی ملکی
۹۳۳۱۰۰۹

استاد درس:
دکتر ممتازی

زمستان ۹۶

فهرست مطالب

۱ مقدمه

پروژه اول و دوم درس طراحی کامپایلرها که در ترم اول سال تحصیلی ۹۶-۹۷ ارائه می‌گردد، با هدف پیاده‌سازی یک تحلیل‌گر واژگان[۱] و همچنین یک تحلیل‌گر نحوی[۲] برای یک گرامر مفروض است. به منظور پیاده‌سازی این دو مورد، از زبان JAVA استفاده شد که طی فاز اول آن فقط با استفاده از ابزار JFlex توانستم تحلیل‌گر واژگان را پیاده‌سازی کنم که در فاز دوم صرفا عملیات در صورت برخورد به هر کلیدواژه[۳] را عوض نموده و به جای نوشتن در خروجی، به تحلیل‌گر نحوی پاس می‌دهد.
همچنین مطابق تعریف پروژه، از ابزار BISON برای تولید تحلیل‌گر نحوی استفاده شد که در ادامه جزییات آن ذکر می‌شود.

۲ شرح پیاده‌سازی

۲-۱ فاز ۱ (تحلیل‌گر واژگان)

با توجه به گرامر داده شده، ابتدا تمامی کلیدواژه‌ها و حروف رزرو استخراج شدند که برای نوشتن فایل توصیف JFlex مورد نیاز بودند. سپس با تعریف مناسب آن‌ها در قالب فایل توصیف JFlex، عملیات مورد نیاز در صورت مشاهده هر کلیدواژه تعریف شد که در این فاز صرفا نوشتن آن کلید واژه به همراه ضمایم و متعلقاتش در کنسول خروجی بود. فایل توصیف JFlex فاز اول، در آدرس زیر موجود و قابل مشاهده است:

```
src\Ph1\lexer.flex
```

۲-۲ فاز ۲ (تحلیل‌گر نحوی)

در این فاز، در ابتدا می‌بایست که قوانین گرامر را به قالب نحو ابزار YACC و در چارچوب یک فایل توصیف YACC می‌نوشتم. پس از انجام این‌کار، فایل را به عنوان ورودی به ابزار BISON تحویل داده و خروجی آن که یک فایل با پسوند CACC بود تحویل گرفته شد. با تغییر پسوند فایل خروجی از CACC به JAVA، امکان اجرای آن فراهم شد. همچنین با تغییر فایل JFlex، نسبت به پاس دادن کلیدواژه خوانده شده به تحلیل‌گر نحوی مبادرت ورزیدم که در نتیجه آن، صرفا می‌بایست تکه کد قابل اجرا در صورت مشاهده هر کلیدواژه را عوض می‌نمودم. با انجام تغییرات ذکر شده و با استفاده از کلاس Main، با یکپارچه کردن فایل خروجی تحلیل‌گر واژگانی و تحلیل‌گر نحوی توانستم در نهایت به تحلیل نحوی یکی از مثال‌ها با استفاده از گرامر داده شده بپردازم.

۳ مسائل و مشکلات

تحلیل‌گر واژگان که یکی از مشکلاتی که مربوط به این بخش بود که فقط در پیاده‌سازی فاز دوم خودش را نشان داد، تغییر دادن دستورهای مواجهه با هر کلیدواژه بود که بنده در ابتدا با استفاده از متد `echoFinding` صرفا نام کلیدواژه و مقدار آن را در کنسول خروجی می‌نوشتم که در پیاده‌سازی فاز دوم، می بایست به‌جای این‌کار، دستور بازگرداندن توکن برای تحلیل‌گرنحوی را اجرا می‌کردم. پس از صرف وقت و مطالعه و جستجو از منابع آنلاین، به این نتیجه رسیدم و پیاده‌سازی تحلیل‌گر واژگان را کامل نمودم.

تحلیل‌گر نحوی در پیاده‌سازی این فاز از پروژه به مشکلات عدیده‌ای برخورد کردم که در ادامه هرکدام را توضیح می‌دهم:

۱. وجود سمبل خالی در گرامر برای رفع این مشکل، تمامی قواعد گرامر را مجددا بازنگری کرده و بدون سمبل خالی[۴] نوشتم. این کار منجر شد که از ۸۲ قاعده تولیدی، به ۹۰ قاعده تولیدی برسم. در حین اعمال این تغییر، در هر قاعده تولیدی که امکان رخ‌دادن سمبل خالی بود، یکبار به صورت بدون سمبل خالی و یکبار با حذف عبارتی که منجر به تولید سمبل خالی می‌شد نوشته می‌شد؛ که در نتیجه مجموعه ۸ قاعده به گرامر اضافه شد.

۲. وجود ابهام در گرامر که منجر به تولید تناقض از نوع RR یا SR در تولید تحلیل‌گر نحوی آن می‌شد. برای حل این مشکل می بایست که قوانین را به صورت بهینه‌تر بازنویسی می‌کردم. با انجام این مهم، بسیاری از تناقضات برطرف شد که از جمله آن‌ها می‌توان به تناقض میان قاعده ۱۷ و ۱۸ گرامر داده شده اشاره کرد که در نهایت با ادغام این دو قاعده، موفق به رفع این مشکل شدم.
همچنین برای رفع ابهام قاعده اگر و آنگاه (قسمت دوم و سوم قاعده ۱۶) - مشکل dangling else - در ابتدا سعی به برطرف‌سازی از

---

[۱] Lexer
[۲] Parser
[۳] Token
[۴] ϵ

طریق تغییر گرامر داشتم که بی‌نتیجه ماند و دچار بروز تناقضات بیشتر در گرامر می‌شد. با جستجو از منابع آنلاین، به این نتیجه رسیدم که یکی از روش‌های حل این موضوع در ابزار ،YACC استفاده از قابلیت اولویت‌دهی به کلیدواژه‌های آن است که این مسئله باعث می‌شود ابزار به طور اتوماتیک محل شیفت یا کاهش را تشخیص دهد. بنابراین با استفاده از مشخصه `nonassoc%` برای دو کلیدواژه «آنگاه» و «وگرنه» توانستم این مشکل را رفع کنم. شایان ذکر است که در این تغییر، قواعد گرامر مربوط، بدون تغییر باقی ماندند.

۳. اعمال اولویت‌های ریاضی و منطقی که همانطور که گفته شد با مشخص کردن اولویت‌های کلیدواژه‌های هرکدام از عملگرها در فایل توصیف قادرYACC به انجام این کار شدم.

اجرای تحلیل‌گر نحوی که با استفاده از کلاس انجامMain می‌گیرد، کلاس‌های جاوا که در خروجی‌های ابزارهای YACC و FLex تولیدشده‌اند اندکی با یکدیگر تعارض داشتند که یک مورد دچار عدم اجرای برنامه می‌شد: کلاس مربوط به تحلیل‌گر واژگان، در خروجی متد `()yylex` خود، نتیجه‌ای از جنس کلاس `YYToken` بازمی‌گرداند که در تعارض با خروجی مورد انتظار تحلیل‌گر نحوی بود. تحلیل‌گر نحوی نیاز به خروجی از نوع `Integer` داشت که این مورد را با اعمال تغییر در نوع خروجی متد `()yylex` تحلیل‌گر واژگانی برطرف ساختم.