

پروژه کامپایلر فاز سوم

ملیکا داودزاده ۶۱۰۳۹۴۱۵۷

این فاز از پروژه بر اساس گرامر Bison که در اختیار ما قرار گرفت انجام شده است. نمای کلی گرامر Bison به صورت زیر است :

`%{`

*Prologue*

`%}`

*Bison declarations*

`%%`

*Grammar rules*

`%%`

*Epilogue*

هرکدام از قسمت های بالا کار خاصی را انجام میدهند برای مثال Bison declarations نام ترمینال ها و غیر ترمینال ها را مشخص می کند همچنین به کمک آن precedence ها را مشخص می کنیم. در گرامر ها دو نوع اشتراک داریم : چپ و راست. با استفاده از `%left`, `%right`, `%nonassoc`, `%precedence` توکن ها را تعریف میکنیم و precedence آن را مشخص می کنیم. فرض کنید `op` یک عملگر باشد. حال عبارت زیر را در نظر بگیرید:

“aopbopc”

اشتراک چپ به این معناست که ابتدا `a` و `b` را باهم گرفته و به همین منوال اشتراک راست یعنی ابتدا `b` و `c` را باهم بگیریم. ( با `%left`, `%right` نشان میدهیم)

`%nonassoc` هیچ اشتراکی را نشان نمیدهد که به این معناست که عبارت ورودی یک خطای گرامری است.

`%precedence` نیز هیچ اشتراکی را تعریف نمیکند اما به آن یک مقدمه ( precedence ) نسبت می دهد و ممکن است بعدا اشتراک برای آن تعریف شود.

حال به گرامر میپردازیم:

حالت کلی:

*result: components . . . ;*

در گرامر زیر مقداری برگردانده می شود:

```
exp : NUM          {$$=$1 ; }  
    | exp '+' exp {$$=$1 + $3 ; }
```

با توجه به این که زبان مورد نظر ما یعنی COOL یک زبان شی گراست و طبق اولین نکته ای که راجع به آن در ویدیوها و هم چنین COOL manual گفته شد هر برنامه ای به این زبان از تعدادی کلاس تشکیل شده است. پس باید قانون class\_list را تعریف کنیم. به کد زیر توجه کنید :

```
class_list : class  
            {$$ = single_Classes( $1 ) ;    parse_results = $$}  
            | class_list class {  
                $$ = append_Classes ( $1 , single_Classes ( $2 ) ) parse_results =  
                $$  
            } ;
```

حال باید وراثت در کلاس ها را در نظر بگیریم :

```
class : class TYPEID ' { ' dummy_feature_list ' } ' ' ; '  
        { $$ = class _ ( $2 , idtable.add_string ( " Object " ) , $4 ,  
            stringtable.add_string ( curr_filename ) ) ; }  
        | class TYPEID INHERITS TYPEID ' { ' dummy_feature_list ' } ' ' ; '  
        { $$ = class ( $2 , $4 , $6 , stringtable.add_string ( curr_filename ) ) ;  
        | error  
        ;
```

همچنین باید قواعد فرمال ها و عبارت های غیر ترمینال که چند حالت مختلف دارد که در کد مشخص شده است. قواعد برای عبارت ها دیگر نیز به صورت کامل و واضح به همراه توضیحات در فایل کد پروژه قرار گرفته است.