به نام خدا

گزارش فاز اول پروژه
طراحی تحلیل‌گر لغوی

فاطمه شفیعی اردستانی

رضا پورمحمدی

در ابتدا برای راحتی در نوشتن قوانین متغییرهای زیر را تعریف می کنیم. digit به منظور نمایش ارقام ۰ تا ۹ ، printabe به منظور نمایش تمام کارکترهای قابل چاپ و alpha برای نمایش تمام حروف انگلیسی کوچک و بزرگ در نظر گرفته شده اند.

```
digit      [0-9]
printabe   [ -~]
alpha      [a-zA-Z]
```

باتوجه‌به پروژهٔ تعریف شده، توکن‌های زیر باید شناسایی شوند:

۱-مقادیر ثابت عددی:

ازآنجایی‌که اعداد صحیح می‌توانند مثبت یا منفی باشند، برنامه باید بتواند تشخیص دهد علامت قبل از عدد ثابت، عملگر است یا علامت عدد است؛ به این منظور همواره دو توکن اخری که شناسایی شده‌اند را ذخیره می‌کنیم.

سپس هنگام دیدن دنباله‌ای از اعداد و بعد از تشخیص اینکه توکن ثابت داریم، بررسی می‌کنیم که آیا توکن قبلی، عملگر + یا عملگر - بوده است. اگر چنین عملگرهایی وجود نداشتند، بررسی میکنیم که مقدار عدد ثابت، دربازه ی اعداد ۱۶ بیتی هست یا نه و با توجه به نتیجه، توکن مناسب را به ان اختصاص می‌دهیم.

اگر قبل از عدد، عملگر وجود داشت، بررسی می‌کنیم که قبل از عملگر چیست، اگر عدد یا متغیر بود، مجدداً عدد مشاهده شده را باتوجه‌به مقدارش با توکن مناسب، مشخص می‌کنیم. اما اگر قبل از علامت مثبت یا منفی، توکنی جز عدد یا متغیر داشتیم، به این معناست که علامت، متعلق به خود عدد است، بنابراین توکنی که با عنوان توکن علامت نام‌گذاری شده بود را، با توکن عدد ثابت، رونویسی می‌کنیم.

```
{digit}+ {
    if(pre2 == "TOKEN_PLUS\n" || pre2 == "TOKEN_MINUSE\n"){
        if(pre1 == "TOKEN_INTCONST\n" || pre1 == "TOKEN_INVALIDINTCONST\n" ||
pre1 == "TOKEN_IDENTIFIER\n"){
            fprintf(fp, pre1);
            pre1 = pre2;
```

```
            n = atoi(yytext);
            if (n<=32767){
                pre2 = "TOKEN_INTCONST\n";
            }
            else{
                pre2 = "TOKEN_INVALIDINTCONST\n";
            }
        }
        else{
            n = atoi(yytext);
            if (n<=32767){
                pre2 = "TOKEN_INTCONST\n";
            }
            else{
                pre2 = "TOKEN_INVALIDINTCONST\n";
            }
        }
    }
    else{
        fprintf(fp, pre1);
        pre1 = pre2;
        n = atoi(yytext);
        if (n<=32767){
            pre2 = "TOKEN_INTCONST\n";
        }
        else{
            pre2 = "TOKEN_INVALIDINTCONST\n";
        }
    }

}
```

۲-مقادیر ثابت کاراکتری:

مقادیر ثابت کارکتری بین دو دبل کوتیشن قرار دارند و ممکن است شامل هر کارکتر قابل چاپی باشند. بنابر این قانون آن به فرم زیر نوشته می شود:

```
(\"{printabe}*\")
```

۳- نام متغیرها و توابع:

باتوجه‌به اینکه نام متغیر و توابع باید با حروف یا _ شروع شوند عبارت زیر را برای تشخص آن داریم:

```
({alpha}|_)({alpha}|_|{digit})*
```

۴- کامنت تک خطی یا چند خطی:

در کامنت چند خطی می‌توانیم بعد از دیدن کاراکتر # هر کاراکتری به جز \n را به هر تعداد ببینیم، بنابراین عبارت منظم آن به شکل زیر است:

```
(#(.)*)
```

در کامنت چند خطی، بعد از دیدن !# تا زمان دیدن #! می‌توان همهٔ کاراکترها را به هر تعداد دید بنابراین داریم:

```
(#!({printabe}|"\n")*!#)
```

۵-کلمات کلیدی، عملگرها و سایر توکن‌ها:

برای تشخیص این توکن‌ها، کافی است نام انها را در بین "" قرار دهیم.

```
"int"
```