به نام خدا

داکیومنت پروژه پایانی کامپایلر

این برنامه از دو فاز اصلی تشکیل شده است: تحلیلگر لغوی و تحلیلگر نحوی

تحلیلگر لغوی توکن ها را مشخص می­کند، تحلیل گر نحوی درخت را رسم میکند.

در این پروژه با استفاده از این دو فاز، ابتدا توکن های برنامه ورودی را تشخیص داده و در جدول توکن ها ذخیره می­کنیم. و سپس درخت را رسم می کنیم .

الف) تحلیل گر لغوی:

* تابع InputFileRE() یک عبات منظم را دریافت می نماید. این عبارت منظم، اجزای تشکیل دهنده زبان پاسکال را توصیف میکند.
* تابع ConvertInfixToPostfix() عبارت منظم میانوندی را به پسوندی تبدیل می نماید. (با استفاده از الگوریتم لهستانی)
* تابع ConvertREtoNFA() به کمک الگوریتم تامسون، عبارت منظم پسوندی را بهnfa تبدیل میکند.
* تابع ConvertNFAtoDFA() خروجی مرحله قبل را بهdfa تبدیل میکند. (با استفاده از الگوریتم EpsilonClosure )
* تابع InputFileProgram() برنامه ورودی به زبان پاسکال را از فایل می خواند.
* تابع Scanner() به کمک بافر وdfa مرحله قبل، جدول توکن ها را ایجاد می نماید.

ب) تحلیلگر نحوی

در این مرحله میخواهیم درخت اشتقاق را برای برنامه ورودی رسم نماییم.

بدین منظور از الگوریتم recursive descent parser استفاده می نماییم.

از آنجا که این الگوریتم، درخت را به روش Top Down ایجاد می نماید، ابتدا گرامر داده شده در صورت پروژه را بررسی میکنیم و بازگشتی چپ و فاکتور چپ را بر طرف میکنیم.

گرامری که در برنامه استفاده شده و برای ان درخت تولید میشود، بازگشتی چپ و فاکتور چپ آن حل شده است و قوانین لازم اضافه شده است.

سپس ما تمام ترمینال ها و غیر پایانه ها را مشخص میکنیم و برای هر غیر پایانه، تابع مورد نظر را مینویسیم. که در این تابع ها به ازای ترمینال ها، گره تولید میکند.

تابع CreateTree() درخت را رسم می نماید.

تابع NextToken() توکن بعدی را از جدول میخواند.

تابع PrintTree() پس از ایجاد درخت در حافظه، آن را رسم می نماید.