شرح کلی

این برنامه یک فایل ورودی شامل تعدادی کلمه ( یک کد منبع برنامه ) را به عنوان ورودی می گیرد ( فایل input.txt که در محل کد منبع برنامه است ) و آن را از نظر لغوی تحلیل می کند؛ یعنی مشخص می کند که هر کلمه از چه نوع می باشد ( کلمه ی کلیدی یا شناسه یا ثابت یا عملگر یا علامت ) و خطا های لغوی را تشخیص داده و دنباله ای از نشانه ها ( tokens ) ایجاد می کند و در آخر دنباله ی نشانه ها، خطاها و تعداد خطاها را نمایش می دهد.

نکته ای که در فایل ورودی باید در نظر گرفته شود

برنامه به گونه ای است که هر رشته ای که میان آن فاصله نباشد را یک واحد در نظر می گیرد و عامل جداکننده ی کلمات از یک دیگر کاراکترهای فضای خالی می باشد ( space , tab, break line, new line ).

برای مثال رشته ی a=2+3; یک واحد در نظر گرفته می شود و برای اصلاح باید آن را به صورت a = 2 + 3 ; در فایل نوشت. برای اینکه عناصر مختلف تشخیص داده شوند باید میان آن ها فاصله باشد. نمونه های دیگر:

Int fun(a,b) باید به صورت int fun ( a , b ) و a() باید به صورت a ( ) نوشته شود

شرح جزئیات

جزئیات توابع به ترتیب الگوریتمی که در تابع اجرا شده شرح داده می شود.

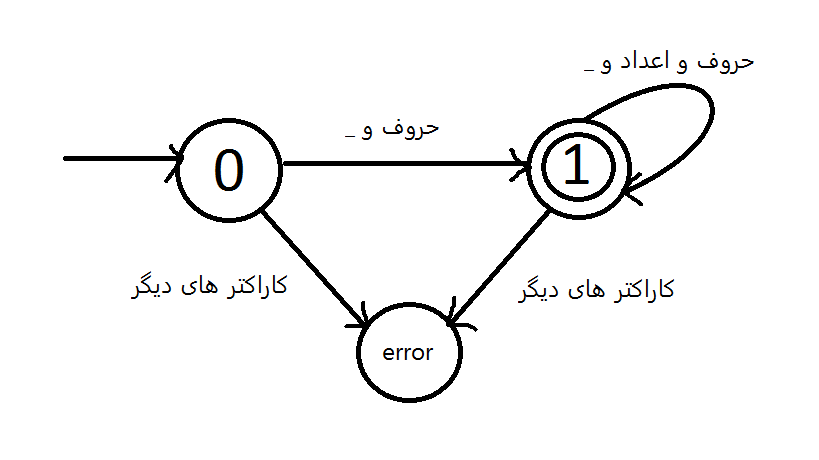
* تابع main

کار این تابع اجرا کردن برنامه است.

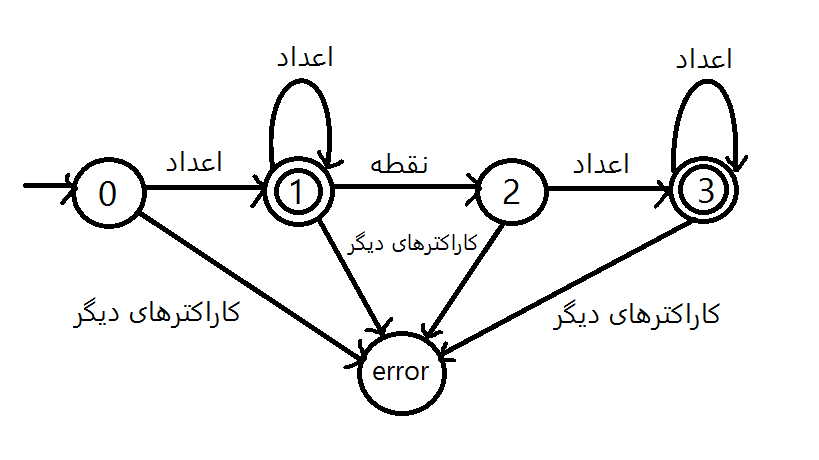
* 1. پیغام های نخستین را به نمایش در می آورد.
  2. فایل ورودی را از محل کد منبع می خواند ( input.txt ).
  3. یک آرایه برای ذخیره ی دنباله ی نشانه ها اعلان می کند ( tok ).
  4. متغیر های مورد نیاز را اعلان می کند.
  5. عناصر آرایه ی نشانه ها را با مقادیر درست مقدار دهی می کند :
     + فایل ورودی را کاراکتر به کاراکتر بررسی می کند.
     + تا زمانی که به کاراکتر فضای خالی برسد، کاراکترها را تک به تک به یک رشته ی موقتی اضافه می کند.
     + وقتی به کاراکتر فضای خالی رسید ابتدا رشته موقتی که کاراکترها را داخل آن ریخته بودیم بررسی می کند؛ اگر آن رشته خالی نباشد به این نتیجه می رسد که یک کلمه را تشخیص داده است.
     + تابع type را با کلمه ی تشخیص داده شده به عنوان ورودی فراخوانی می کند تا نوع کلمه تشخیص داده شود. خروجی تابع type نوع کلمه مورد نظر است. سپس عنصر بالای آرایه ی tok را با نوع کلمه ( نشنانه ی برگشت داده شده توسط تابع type ) مقداردهی می کند.
     + رشته ی موقتی را خالی می کند و تا زمانی که به انتهای فایل ورودی برسیم همین روند را تکرار می کند.
  6. دنباله ی نشانه ها (عناصر آرایه tok به ترتیب) را به نمایش در می آورد.
  7. تعداد خطاها را اعلام می کند.
  8. برنامه را خاتمه می دهد.
* تابع type

این تابع نوع رشته ای که به عنوان ورودی به آن داده شده است را برگشت میدهد.

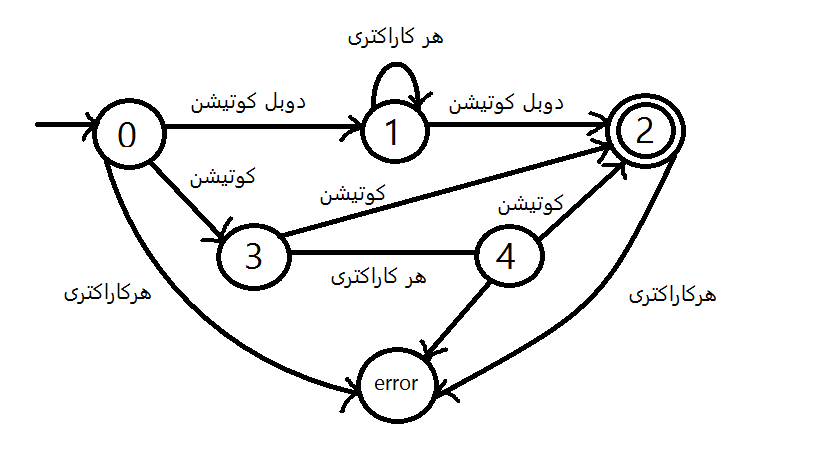
* 1. یک رشته به عنوان ورودی به آن داده شده است.
  2. کاراکتر اول آن را بررسی می کند.
  3. برطبق کاراکتر اول رشته توابع دیگر را برای تشخیص نوع رشته فراخوانی می کند:
     + اگر کاراکتر اول رشته یکی از حروف کوچک یا بزرگ یا کاراکتر underline بود، تابع isId را (با رشته به عنوان ورودی) فراخوانی می کند تا مشخص شود رشته مورد نظر شناسه است یا کلمه ی کلیدی یا دارای خطا می باشد.
     + اگر کاراکتر اول رشته یکی از اعداد 0 تا 9 باشد تابع isNum را (با رشته به عنوان ورودی) فراخوانی می کند تا مشخص شود که رشته یک ثابت از نوع عدد صحیح است یا عدد اعشاری و یا دارای خطا می باشد.
     + اگر کاراکتر اول رشته دوبل کوتیشن ( " ) یا کوتیشن ( ' ) بود، تابع isStr (با رشته به عنوان ورودی) فراخوانی می شود تا مشخص شود رشته یک ثابت رشته ای است یا دارای خطا می باشد.
     + اگر هیچ یک از شرط های بالا بر قرار نبود، تابع isSign (با رشته به عنوان ورودی) فراخوانی می شود تا مشخص شود که آیا زشته یک عملگر است یا علامت و یا دارای خطا می باشد.
  4. نوع ( نشانه ) یا خطای برگشت داده شده توسط تابعی که فراخوانی کرده است را به تابع فراخواننده خود ( تابع main ) برگشت می دهد.
* تابع isId
  1. یک رشته به عنوان ورودی به آن داده شده است که احتمال می رود شناسه یا کلمه ی کلیدی باشد.
  2. توسط ماشین زیر شناسه بودن رشته را بررسی می کند:



* 1. اگر رشته توسط این ماشین پذیرفته نشد تابع یک خطا برگشت می دهد.
  2. اگر رشته توسط این ماشین پذیرفته شد، رشته با تک تک اعضای مجموعه ی کلمات کلیدی ( آرایه ی key ) مقایسه می شود، در صورت برابر بودن با یکی از آن ها تابع نوع ( نشانه ) کلمه ی کلیدی ( keyword ) را برگشت می دهد. در غیر اینصورت نوع ( نشانه ) شناسه ( identifier ) را برگشت می دهد ( به تابع فراخواننده ی خود ( type ) ) .
* تابع isNum
  1. یک رشته به عنوان ورودی به آن داده می شود که احتما می رود یک ثابت عددی باشد.
  2. رشته توسط ماشین زیر بررسی می شود:



* 1. اگر رشته توسط ماشین بالا پذیرفته نشود، تابع خطا برگشت می دهد.
  2. اگر رشته توسط ماشین بالا در حالت 1 پذیرفته شود، تابع نوع ( نشانه ) ثابت عددی صحیح ( literal-number-integer ) را برگشت می دهد.
  3. اگر رشته توسط ماشین بالا در حالت 3 پذیرفته شود، تابع نوع ( نشانه ) ثابت عددی اعشاری ( literal-number-float ) را برگشت می دهد (به تابع فراخواننده ی خود (type)).
* تابع isStr
  1. یک رشته به عنوان ورودی به این تابع داده می شود که احتمال می رود یک ثابت رشته ای باشد.
  2. رشته توسط ماشین زیر بررسی می شود:



* 1. اگر رشته توسط ماشین پذیرفته نشود، تابع یه خطا برگشت می دهد.
  2. اگر رشته توسط ماشین پذیرفته شود، نوع ( نشانه ) ثابت رشته ای (literal-string) را برگشت می دهد (به تابع فراخواننده ی خود (type)).
* تابع isSign
  1. یک رشته را به عنوان ورودی دریافت می کند که احتمال می رود یا جزو عملگرها باشد یا جزو علائم.
  2. رشته را با تک تک اعضای مجموعه ی عملگرها ( آرایه ی op ) مقایسه می کند، اگر با یکی از عناصر مجموعه برابر بود نوع ( نشانه ) عملگر (operator) را برگشت می دهد.
  3. در غیر اینصورت رشته را با تک تک اعضای مجموعه ی علائم ( آرایه ی si ) مقایسه می کند، اگر با یکی از عناصر مجموعه برابر بود نوع (نشانه) علامت ( sign ) را برگشت می دهد.
  4. در غیر اینصورت یک خطا برگشت می دهد. ( به تابع فراخواننده ی خود ( type ) )
* بیرون از توابع
  + یک آرایه برای تعریف مجموعه ی کلمات کلیدی اعلان شده است ( key ) و یک متغیر برای ذخیره ی اندازه ی این آرایه ( sizeOfKey ).
  + یک آرایه برای تعریف مجموعه ی عملگرها اعلان شده است ( op ) و یک متغیر برای ذخیره ی اندازه ی این آرایه ( sizeOfOp ).
  + یک آرایه برای تعریف مجموعه ی علائم اعلان شده است ( si ) و یک متغیر برای ذخیره ی اندازه ی این آرایه ( sizeOfSi ).
  + یک متغیر برای شمارش تعداد خطاها اعلان شده است ( e ).

نکات

* خطاهایی که توابع برگشت می دهند یک رشته است شامل کلمه ی error، به دنبال آن خط فاصله (-) و بعد از آن کلمه ای که شامل خطا می باشد، که در نهایت در نشانه ی (token) مربوط به کلمه، در آرایه ی tok ذخیره می شود و این خطاها در هنگام نمایش عناصر آرایه ی tok به نمایش در می آیند.
* مجموعه های کلمات کلیدی، عملگرها و علائم تنها شامل بخشی از آن ها می باشند که اگر بخواهیم عنصر جدیدی به آن ها اضافه کنیم باید توجه داشته باشم که باید مقدار متغیرهای اندازه ( sizeOf ) مربوط به آن ها را نیز متناسب با آن تغییر دهیم.
* برای شمارش خطاها، در هنگامی که یک تابع میخواهد خطا برگشت دهد، قبل از آن شمارنده ی خطا ( e ) را یک واحد افزایش می دهد.