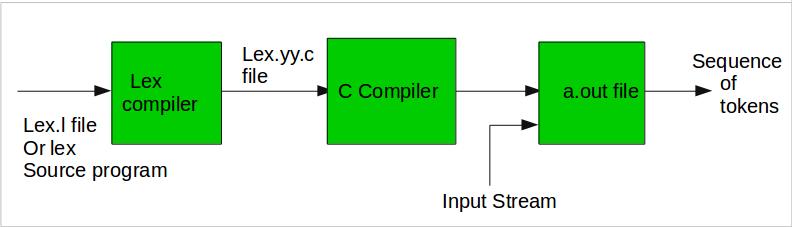
HW1 Simple Pascal Scanner using Lex

B08304029邱品諺

1. Flex 2.6.4
2. Ubuntu 64-bit 20.04.4



利用lex進行編譯的過程如上圖所示，首先先將lex檔編譯成C檔，再將C檔編譯成執行檔並引入輸入檔以scan輸入檔的token。撰寫lex的方式主要要先定義token可能出現的種類，如string、comment等等，並在定義的部分規定其regular expression，再來輸入流的token會由上到下依序對應定義的token種類，並執行其定義的lex rule部分，以印出各token所對應的種類及其合法或非法。

1. Reserved word

由於Pascal是case-insensitive的語言，所以一個word的每個字元有可能會是大寫或小寫，因此在regular expression的表示上需要利用[]來表示每個字元的可能，例如：and需要以[Aa][Nn][Dd]來表示其regular expression。

1. Identifiers

合法的identifiers第一個字只能是英文字母或是底線符號，之後可以是英文字母、數字或底線符號。需要注意的是可能會將分號、逗號、冒號和括號等符號與前面的identifier合在一起表示，因此在regular expression的表示上需要特別處理。而除了第一個字不是英文字母或底線符號所組成的identifier為非法的以外，只要其中任何一個字元只要出現非英文字母、數字、底線符號即為非法identifiers。因此在regular expression的表示上無法僅利用第一個字不為英文字母及底線符號的線索判斷，也需要考慮後續所出現的可能字元。另外，由於規定長度不可超過15，而只需在合法identifier所定義的規則（lex rule）下，利用yyleng判斷長度是否超過15即可。

1. Symbols

利用regular expression表示可能出現的符號，如果出現不只1 byte的符號利用|（alternative）分開即可。需特別注意的是逗號（,）不在列表中，所以本程式特別寫了一個定義comma，使得在scan時可將其忽略。

1. 實數

實數除了要考慮有無正負外，還要考慮有無小數及有無exponential的部分，且小數不可多個0結尾，如：1.00，另外，整數除了0以外，其餘不可以0開頭，如003，而這些利用regular expression即可完整表示。另外，非法的實數其實在regular expression的表示上與合法的類似，只要不加以限制0出現在非法的位置，如上述例子，因此利用regular expression即可完整表示。

1. Quoted String

由於只要在引號（'’）內的任何字元皆為字串，因此在regular expression的表示上非常單純，但是需要考慮一些特別情形，例如：空字串、字串中有空白、跳脫字元等等。跳脫字元的處理上，欲將’You’’ll see’顯示為’You’ll see’，因此需要在lex rule的部分特別處理，本程式中使用迴圈跑過欲顯示的字串，當遇到引號（'）時便向後檢查是否也是引號（'），如果是的話便將引號後的所有字源都向前挪一格，且在最後一個被挪動的加上字串結尾（'\0），挪動完畢後便在向後檢查是否仍有跳脫字元直到結束。而在非法字串的表示上需要考慮引號未刮完全，或是字串中只有奇數個引號的部分，例如：’ab、’a’b’, ab’等等。另外，長度30以上的字串被視為非法字串，只需利用yyleng判斷token長度即可。

1. 註解

只要在(\*及\*)內，除非中間出現\*)，否則皆視為合法註解，因此如：(\* co(\*mme\*)nt \*)、 (\*ab\*)\*\*)皆為非法的註解，而(\* co(\*ma\*)為合法註解。標示註解合法或非法上，本程式優先處理非法的註解，剩下的即為合法註解。另外，由於註解可以跨行，因此除了在regular expression的表示上需注意外，在行數的表示上需要調整。

1. 數學運算式中實數及符號的分離

由於在賦值時可能給予變數一串數學運算式，且實數有正負，如i = 1+2。因此需特別分離何者為實數何者為運算符號，本程式設計將所有數學運算式先判斷為非法的identifier，再從中判斷如果所有字元皆為數字及符號的話即為數學運算式。首先，如果正負符號前面如果是數字的話，即表示其為運算符號而非表示實數正負的符號，而乘號及除號絕對為運算符號，另外，判斷實數的部分需先考慮其數字前符號的前一個位置是否為數字，如果為數字的話即表示其為無正負的實數，反之即為有正負的實數，此部分本程式利用一個字串及迴圈將該實數取出並顯示。

1. Scanner如何在發生錯誤時做recover？

首先需要先能夠偵測出錯誤，本程式針對每個token的型態表示其合法及非法的regular expression，因此可判斷出該token為何種型態且為合法或非法。所以當出現非法的token時，可持續辨識接下來的token直到結束，中途並不會暫停。

1. 實做一個scanner並不容易，首先，想出一個regular expression需要絞盡腦汁外，還有可能存在有些特例因而無法表示所有的情況，所以需要在lex rule的部分撰寫規則進行分類，或是分析、拆解字元等等。

