Laboratorio #7

Para este laboratorio extenderá la funcionalidad de su laboratorio anterior para el adecuado procesamiento de *tokens* según la gramática de Cocol/R.

Su compilador generado en este laboratorio se enfocará exclusivamente en el reconocimiento de TOKENS. En el laboratorio #6 se desarrolló la capacidad de reconocer la pertenencia de caracteres individuales a conjuntos (letras, dígitos, dígitos hexadecimales, etc.), y el caso especial de palabras clave como una secuencia específica de caracteres. Este laboratorio se apoyará sobre esa funcionalidad para ver si la forma en la que se relacionan los caracteres leídos conforma un *token* de acuerdo a la gramática. Su analizador generado, sin embargo, dejará de reportar ocurrencias aisladas de conjuntos en CHARACTERS.

Por ejemplo, si nuestra gramática ingresada al generador es la siguiente:

```
COMPILER Ejemplo

CHARACTERS
      letter = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".
      digit = "0123456789".
      hexdigit = digit+"ABCDEF".
KEYWORDS
      if="if".
      while="while".
TOKENS
      id = letter{letter} EXCEPT KEYWORDS.
      number = digit{digit}.
      hexnumber = hexdigit{hexdigit}"(H)".

IGNORE " ".

END Ejemplo.
```

Y el programa dado al analizador generado es el siguiente:

```
hola 1234 567ABC(H) while
```

Los elementos reconocidos deben ser:

<id> <number> <hexnumber> <while>

Nótese que sólo se reportan *tokens*. No reportamos **<id>**, y además cada caracter en "`hola`" como una **<letter>**. Es también importante observar que aunque "`while`" conforma con la definición de **<id>** no se reconoce como tal ya que **<id>** incluye `EXCEPT KEYWORDS`, lo que indica que si un **<id>** leído también conforma con la definición de una palabra clave se reportará la palabra clave.

Con el objetivo de dejar los requerimientos lo más claros posible se explicarán algunos casos de error que se podrían dar en el ejemplo anterior:

- Si **<id>** no incluye `EXCEPT KEYWORDS` hay dos posibles caminos: reportar un error porque no se puede distinguir si "`while`" **<id>** o es la palabra clave **<while>**; o reportar "`while`" como **<id>** porque los *tokens* tienen prioridad
- Si el programa dado al analizador incluye algo como `567ABC` esto debe ser reportado como un error debido a que no hay un *token* en la definición que consista únicamente de **<hexdigit>**s. Si **<hexnumber>** no incluyera "`(H)`" al final sería imposible distinguir una secuencia de números sin letras como `1234` entre **<hexnumber>** y **<number>**
- Si el programa dado al analizador incluye algo como `hola1234` esto también debe ser reportado como un error, ya que no hay un *token* cuya definición combine letras y números de esa forma. Nótese que **<hexnumber>** permite letras y números pero las letras deben ser mayúsculas, y no incluyen '`h`', '`o`' ni '`l`'

Se recomienda predeterminar un conjunto de *white space* consistente de *tabs*, espacios en blanco y cambios de línea. Recuerde, sin embargo, que si el diseñador del lenguaje incluye su propio conjunto `IGNORE` en la gramática se deberá reemplazar el conjunto predeterminado por lo que el diseñador especificó.