Reconocimiento de tokens

Previamente se trató el problema de cómo especificar los componentes léxicos. Ahora abordaremos el reconocimiento de los componentes léxicos y se utiliza como ejemplo el lenguaje generado por una gramática donde los terminales if, then, else, oprel, id y num generan conjuntos de cadenas dados por las definiciones regulares y donde letra y digito ya se definieron anteriormente.

prop →	**if** expr **then** prop
	|**if** expr **then** prop **else** prop
	|**ε**
expr →	termino oprel termino
	|termino
termino →	id | num
if →	**if**
then →	**then**
else →	**else**
oprel →	**< | <= |** = **| < > | > | >=**
id →	letra (letra | digito) *
num →	digito(. digito+)? (E (+ | –)? digito +)?
delim→	[**\t\n**]
ws →	delim +

Recordar que los terminales son los tokens, y que los **no terminales** producen terminales.

Para este lenguaje de ejemplo, el analizador léxico reconocerá las palabras clave if, then, else. así como los lexemas representados por oprel, id y núm. Para simplificar las cosas, se supone que las palabras clave son reservadas; es decir, no se pueden usar como identificadores y se supone también que los lexemas están separados por espacio en blanco. El analizador léxico eliminará los espacios en blanco. El objetivo es construir un analizador léxico que aísle el lexema para el siguiente componente

léxico del buffer de entrada y que produzca como salida un par formado por el componente léxico apropiado y el valor de atributo, utilizando la tabla de traducción.

EXPR.REG.	COMP. LEX . (token)	VALOR DEL ATRIBUTO
ws	-	-
if	IF	-
then	THEN	-
else	ELSE	-
id	ID	apuntador a la entrada en la tabla
num	NUM	valor entero correspondiente a num
<	OPREL	MEN
<=	OPREL	MEI
=	OPREL	IGU
<>	OPREL	DIF
>	OPREL	MAY
>=	OPREL	MAI