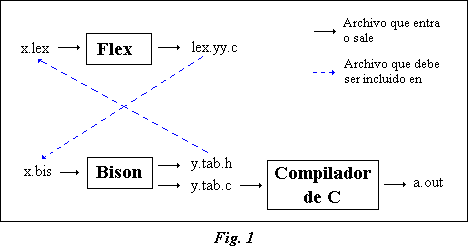
Existen un par de herramientas las cuales hacen más fácil el desarrollo de compiladores a partir de la especificación de los lenguajes en expresiones regulares y reglas en BNF. Estos programas, llamados “flex” y “bison”, nos ayudan a crear la parte del análisis léxico y sintáctico, respectivamente.

Flex, el cual es el sucesor de lex, es un programa que nos genera un archivo en lenguaje c, que representa nuestro analizador léxico.

Al igual que con flex; bison, sucesor de yacc; nos genera un archivo en c, pero a diferencia de flex, lo que representa es nuestro analizador sintáctico.

El analizador léxico puede funcionar por sí solo, mientras que el analizador sintáctico se apoya (hace llamadas) en el analizador léxico, por lo que es necesario incluir este último en el analizador sintáctico (como veremos más adelante).



En la imagen anterior se muestra un esquema de los archivos que deben ser pasados a los respectivos programas y los archivos que se generan. Más adelante en el documento se analizarán las inclusiones de los archivos y la forma de compilarlos.

***Descripción de los archivos flex/bison***

Ambos archivos, el que será compilado en flex y el de bison, tienen una estructura en común, la cual se muestra a continuación:

*declaraciones*

%%

*reglas de traducción flex/bison*

%%

*implementación de funciones de apoyo en c*

básicamente consta de tres partes las cuales están separadas por el símbolo “%%”, y la primera parte a su vez esta separa en declaraciones propias de cada programa y en declaraciones para c, donde estas últimas están limitas por los símbolos: “%{” y “%}”:

%{

*declaraciones para C*

%}

*declaraciones para el programa flex/bison*

Estructura específica para el archivo flex:

Declaraciones para c: es la parte donde se hacen los includes y la declaración de variables en código C que sirven de apoyo.

Declaraciones para el programa flex: Aquí se pueden incluir las definiciones regulares que son proposiciones que se utilizan como componentes en las expresiones regulares que aparecen en las reglas de traducción.

Reglas de traducción flex: Son proposiciones de la forma: pn {acciones}, donde pn es una expresión regular y las acciones son un fragmento de programa escrito en c que describe que instrucciones realizará el analizador léxico cuando el patrón pn concuerda con la cadena leída.

Implementaciones de funciones de apoyo en c: Contiene las funciones auxiliares en C que pueden necesitar las acciones, y donde se define la función main. Para que el analizador léxico se ejecute tendrá debe ponerse en alguna parte del código, según la necesidad del programa, la llamada a la función: yylex().

Estructura específica para el archivo bison:

Declaraciones para c: es la parte donde se hacen los includes y declaración de variables en código c que sirven de apoyo en la creación del compilador.

Declaraciones para el programa bison: es donde se declara el símbolo inicial de la gramática: %start simbolo, donde simbolo es el BNF inicial de la gramática. También se declaran los componentes léxicos (tokens) de nuestro lenguaje, de la forma: %token tok1 tok2 ... tokn, donde tokn representa cada uno de los tokens de nuestra gramática.

Reglas de traducción bison: es aquí donde se ponen las reglas de nuestro lenguaje (en BNF’s), pero con características diferentes a las expuestas en clase, v. gr.:

la regla:

<NT\_1> → <NT\_2> | T

quedaría de la forma:

NT\_1 : NT\_2 | T ;

Implementaciones de funciones de apoyo en c: Contiene las funciones auxiliares que se pueden necesitar, y donde se define la función main. Para que el analizador sintáctico se ejecute deberá escribirse en alguna parte del código, según la necesidad del programa, la llamada a función: yyparse.

Bison es un generador de analizadores sintácticos de propósito general que convierte una descripción gramatical para una gramática independiente del contexto en un programa en C que analice esa gramática. Una vez que sea un experimentado en Bison, podría utilizarlo para desarrollar un amplio rango de analizadores de lenguajes, desde aquellos usados en simples calculadoras de escritorio hasta complejos lenguajes de programación.