Sintaxis y semántica de los lenguajes

Trabajo Práctico Integrador:
Diseño e implementación de Lexer y Parser

Ciclo lectivo: 2023
Primer cuatrimestre

Integrantes Grupo 19 ISI B:

Acosta Quintana, Lautaro

Galeano, Martín

Niveyro, Iván

Morel, Francisco

Localidad:

Resistencia, Chaco

Fecha primera entrega:

30/04/23

Universidad:

Universidad Tecnológica Nacional, Facultad Regional Resistencia

Contenido

1. Introducción

El trabajo consiste en la construcción de un analizador léxico y un analizador sintáctico con el fin de analizar un archivo de entrada, de formato y extensión .docbook, y validarlo para posteriormente traducirlo a otro archivo con formato y extensión .HTML.

Para lograrlo, primero realizamos el diseño de las distintas reglas de construcción con las que se evaluará la entrada del programa. Este diseño está realizado en pseudocódigo y representa cómo se comportaría el analizador sintáctico.

En el desarrollo del analizador léxico, elegimos implementarlo en el lenguaje de programación C (estándar C17), usando el compilador GCC (11.3.0) y el generador de analizadores léxicos Flex (2.6.4). Partiendo de esto, diseñamos expresiones regulares para cada Token.

2. Pseudocódigo

En esta sección se presenta el pseudocódigo inicial en el que se basará el analizador sintáctico. Los terminales están escritos en minúscula y los no terminales en *CamelCase.* Con la intención de facilitar la lectura, los no terminales están coloreados en azul y las producciones gramaticales están divididas en secciones como en el pdf de guía.

Tabla de Tokens	
Token	**Contenido**
texto	Letras, números, signos de puntuación y caracteres especiales
url	
<article>; </article>	Etiqueta de inicio y final de un bloque articulo
<info>; </info>	Etiqueta de inicio y final de un bloque info
<title>; </title>	Etiqueta de inicio y final de un bloque title

<abstract>; </abstract>	Etiqueta de inicio y final de un bloque abstract
<para>; </para>	Etiqueta de inicio y final de un bloque para
<author>; </author>	Etiqueta de inicio y final de un author
<personname>; </personname>	Etiqueta de inicio y final de un nombre completo de persona
<firstname>; </firstname>	Etiqueta de inicio y final del primer nombre de una persona
<surname>; </surname>	Etiqueta de inicio y final del segundo nombre de una persona
<date>; </date>	Etiqueta de inicio y final para una fecha
<section>; </section>	Etiqueta de inicio y final de una sección
<simplesect>;</simplesect>	Etiqueta de inicio y final de una sección simple
<copyright>; </copyright>	Etiqueta de inicio y final para bloque de copyright
<address>; </address>	Etiqueta de inicio y final de una dirección
<city>; </city>	Etiqueta de inicio y final del nombre de una ciudad
<state>; </state>	Etiqueta de inicio y final del nombre de una provincia/estado
<postcode>; </postcode>	Etiqueta de inicio y final de un código postal
<street>; </street>	Etiqueta de inicio y final del nombre de una calle
<email>; </email>	Etiqueta de inicio y final de una dirección de

	correo
<phone>; </phone>	Etiqueta de inicio y final de un número de correo
<itemizedlist>; </itemizedlist>	Etiqueta de inicio y final para una lista itemizada
<listitem>; </listitem>	Etiqueta de inicio y final para una lista
<emphasis>; </emphasis>	Etiqueta de inicio y final de bloque emphasis
<holder>; </holder>	Etiqueta de inicio y final de bloque holder
<simpara>; </simpara>	Etiqueta de inicio y final de bloque simpara
<year>; </year>	Etiqueta de inicio y final para un número de año
<comment>; </comment>	Etiqueta de inicio y final para realizar comentarios
<important>; </important>	Etiqueta de inicio y final de bloque important
<link>; </link>	Etiqueta de inicio y final del bloque link
<videodata fileref =”; />	Etiqueta para adjuntar un archivo del tipo video
<imagedata fileref=”; />	Etiqueta para adjuntar un archivo del tipo imagen
<informaltable>; </informaltable>	Etiqueta de inicio y final de bloque informaltable
<tgroup>; </tgroup>	Etiqueta de inicio y final de bloque tgroup
<row>; </row>	Etiqueta de inicio y final para indicar el

	número de filas de una tabla
<table>; </table>	Etiqueta de inicio y final para crear una tabla
<thead>; </thead> <tfoot>; </tfoot> <tbody>; </tbody>	Etiquetas de inicio y final para la dimensiones de una tabla
<entrytbl>; </entrytbl>	Etiqueta de inicio y final de bloque entrytbl
<link xlink:href: ; />	Etiqueta para URL´s

Etiquetas estructurales:

```
∑ →?xml version="1.0" encoding="UTF-8"?> Article
Article → <article>Info Title Content Section</article>
| <article>Info Title Content SimSection</article>
| <article>Info Title Content</article>
| <article>Info Content Section</article>
| <article>Info Content SimSection</article>
| <article>Info Content</article>
| <article>Title Content Section</article>
| <article>Title Content SimSection</article>
| <article>Title Cuerpo</article>
| <article>Content Section</article>
| <article>Content SimSection</article>
| <article>Content</article>

Content → ItemizedList Content      | ItemizedList
| Important Content                 | Important
| Para Content                      | Para
| SimPara Content                   | SimPara
| Address Content                   | Address
| MediaObject Content               | MediaObject
| InformalTable Content             | InformalTable
| Comment Content                   | Comment
| Abstract Content                  | Abstract
```

Secciones:

```
Section → <section> Info Title Content SimSection</section>
| <section>Info Title Content Section</section>
```

```
| <section>Info Content Section</section>
| <section>Info Content SimSection</section>
| <section>Title Content Section</section>
| <section>Title Content SimSection</section>
| <section>Content Section</section>
| <section>Content SimSection</section>
| <section>Content </section>

SimSection → <simplesect>Info Title Content</simplesect>
| <simplesect>Info Content</simplesect>
| <simplesect>Title Content</simplesect>
| <simplesect>Content</simplesect>
```

Etiquetas básicas de párrafo:

```
InfoContent → Title InfoContent | Title
| InfoContent MediaObject        | MediaObject
| InfoContent Abstract           | Abstract
| InfoContent Address            | Address
| InfoContent Author             | Author
| InfoContent Date               | Date
| InfoContent Copyright          | Copyright

Info → <info> InfoContent </info>

AbstractContent → Para AbstractContent | Para
| SimPara AbstractContent | SimPara

Abstract → <abstract> Title AbstractContent </abstract>
| <abstract> AbstractContent </abstract>

AddressContent → texto AddressContent | texto
| Street AddressContent | Street
| City AddressContent   | City
| State AddressContent  | State
| Phone AddressContent  | Phone
| Email AddressContent  | Email

Address → <address>AddressContent</address>

AuthorContent → Firstname AuthorContent
| Surname AuthorContent

Author → <author> AuthorContent </author>
```

```
CopyrightYearContent → Year CopyrightYearContent
| Year CopyrightHolderContent
| Year

CopyrightHolderContent → Holder CopyrightHolderContent | Holder

Copyright → <copyright>CopyrightYearContent</copyright>

TitleContent → Emphasis TitleContent | Emphasis
| Link TitleContent          | Link
| Email TitleContent         | Email
| texto TitleContent         | texto

Title → <title>TitleContent</title>

SimParaContent → Emphasis | texto
| Link            | Email
| Author          | Comment

SimPara → <simpara>SimPara SimParaContent</simpara>
| <simpara>SimParaContent</simpara>

Emphasis → <emphasis>Emphasis SimParaContent</emphasis>
| <emphasis>SimParaContent</emphasis>

Comment → <comment>Comment SimParaContent</comment>
| <comment>SimParaContent</comment>

Link → <link>Link SimParaContent</link>
| <link>SimParaContent</link>

ParaContent → Emphasis  | Link
| Email     | Author     | Comment        | ItemizedList
| Important | Address    | MediaObject    | InformalTable

Para → <para>Para ParaContent</para>
| <para>ParaContent</para>

Important → <important>Title Content</important>
| <important>Content</important>

SharedContent →SharedContent Comment | Comment
```

```
| SharedContent Emphasis      | Emphasis
| SharedContent Link          | Link
| SharedContent texto         | texto

FirstName   → <firstname>SharedContent</firstname>
Surname     → <surname>SharedContent</surname>
Street      → <street>SharedContent</street>
City        → <city>SharedContent</city>
Phone       → <phone>SharedContent</phone>
Email       → <email>SharedContent</email>
Date        → <date>SharedContent</date>
Year        → <year>SharedContent</year>
Holder      → <holder>SharedContent</holder>
State       → <state>SharedContent</state>
```

Imágenes y multimedia:

```
MediaObjectContent → VideoObject MediaObjectContent | VideoObject
| ImageObject MediaObjectContent | ImageObject

MediaObject → <mediaobject>Info VideoObject
MediaObjectContent</mediaobject>
| <mediaobject>Info ImageObject MediaObjectContent</mediaobject>
| <mediaobject>ImageObject MediaObjectContent</mediaobject>
| <mediaobject>VideoObject MediaObjectContent</mediaobject>
| <mediaobject>VideoObject</mediaobject>
| <mediaobject>VideoObject</mediaobject>

ImageObject→<imageobject> Info ImageData</imageoobject>
|<imageobject>ImageData</imageoobject>

VideoObject→<videoobject>Info VideoData</videoobject>co
|<videoobject>VideoData</videoobject>

VideoData → <videodata fileref ="Ruta" />
ImageData → <imagedata fileref="Ruta" />
Ruta → /texto Ruta/ | texto.extension
```

Listas:

```
ItemizedList→ <itemizedlist>ListItem</itemizedlist>
| <itemizedlist>ListItem ItemizedList</itemizedlist>

ListItem → <listitem>ListItem Content</listitem>
| <listitem>Content</listitem>
```

Tablas:

```
InformalTableContent → MediaObject InformalTableContent | MediaObject
| Tgroup InformalTableContent | Tgroup

InformalTable →<informaltable>InformalTableContent</informaltable>

Tgroup → <tgroup>Thead Tbody Tfoot</tgroup>
| <tgroup>Thead Tfoot</tgroup>
| <tgroup>Tbody Tfoot</tgroup>
| <tgroup>Tfoot </tgroup>

TableContent → Row TableContent | Row
Thead→<thead>TableContent</thead>
Tbody→<tbody>TableContent</tbody>
Tfoot→<tfoot>TableContent</tfoot>

RowContent → Entry RowContent | Entry
| EntryTbl RowContent          | EntryTbl

Row → <row>RowContent</row>

EntryContent → texto EntryContent | texto
| ItemizedList EntryContent     | ItemizedList
| Important EntryContent        | Important
| Para EntryContent             | Para
| SimPara EntryContent          | SimPara
| MediaObject EntryContent      | MediaObject
| Comment EntryContent          | Comment
| Abstract EntryContent         | Abstract

Entry→<entry>EntryContent</entry>

Entrytbl→<entrytbl>Thead Tbody</entrytbl>
Entrytbl→<entrytbl>Tbody</entrytbl>
```

Enlaces:

```
Link → <link xlink:href: Url />
Url → Protocolo://Dominio:Puerto/Ruta#LocalizadorInterno
| Protocolo://Dominio:Puerto/Ruta
| Protocolo://Dominio:Puerto
| Protocolo://Dominio/Ruta # LocalizadorInterno
| Protocolo://Dominio/Ruta
| Protocolo://Dominio#LocalizadorInterno
| Protocolo://Dominio
```

```
Protocolo → http | https | ftp | ftps
Dominio → texto
Puerto → número
LocalizadorInterno → texto
```

3. Analizador Léxico

Un programa escrito usando Flex (2.6.4) consiste de tres secciones, separadas por '%%'. La primera sección contiene declaraciones de variables, librerías a utilizar y opciones del generador. La segunda contiene la lista de patrones (Las expresiones regulares que usaremos) y el código en C a ejecutar cuando un patrón es reconocido. Por último, la tercera sección es código C que es copiado al escáner generado.

```
%{
int chars = 0;
int words = 0;
int lines = 0;
%}

%%

[a-zA-Z]+  { words++; chars += strlen(yytext); }
\n         { chars++; lines++; }
.          { chars++; }

%%

main(int argc, char **argv)
{
  yylex();
  printf("%8d%8d%8d\n", lines, words, chars);
}
```

"Programa contador de palabras"(Levine J. R., 2009, p.3)

El programa desarrollado sigue esta misma estructura. En la primera sección definimos variables e incluimos librerías útiles para el control de errores. También definimos opciones del analizador como:

- **"*noyywrap*"** : Se encarga de evitar que el analizador llame a la función "*yywrap()*", la cual busca el próximo archivo a escanear.
- **"*nodefault*"**: Evita que el analizador incluya una regla que copie la input no aceptada a una variable *yyout*, optando por reportar un error si los patrones no reconocen todas las formas de input posibles.

En la segunda sección, definimos el conjunto de expresiones regulares que utilizaremos para reconocer los tokens que ingresen. Cuando un patrón sea reconocido se mostrará por pantalla la cadena de caracteres que ha sido aceptada por la expresión regular.

Luego, en la tercera sección ocurre el llamado a la función *"yylex()"* la cual es la llamada a la ejecución del escáner. De manera auxiliar, se definieron una serie de funciones y controles que se ejecutan de manera previa a la llamada del escáner:

- **getExtension:** Tiene el objetivo de devolver la extensión del archivo, si es que existe.
- **printWelcome:** Imprime por pantalla un mensaje de bienvenida al usuario.
- **errorControl:** Se encarga de controlar si el archivo enviado al analizador existe y, si existe, controla que la extensión sea la correcta. En cada caso, si no se cumplen las condiciones, termina la ejecución del programa e imprime por pantalla el error correspondiente.

El programa soporta los modos de ejecución interactivo y a partir de un archivo.

Bibliografía

- Levine, J. R. (2009). Flex & Bison: Text Processing Tools. O'Reilly Media.