INSTITUTO TECNOLOGICO DE COSTA RICA

DOCUMENTACION DE COMPILADOR SEMANTICO DE XHTML

IC 5701 COMPILADORES

ESCUELA DE COMPUTACION

INTEGRANTES

Andrés Ramírez Fuentes 201013880

Larissa Rivas Carvajal 201022184

FECHA DE ENTREGA

16 de junio del 2013

Contenido

Descripción del problema

El problema consiste en realizar un analizador semántico para la gramática de un xhtml reducido (que contenga algunos tags en específico, con sus respectivos atributos), que pueda construir un árbol sintáctico a partir del parser que fue desarrollado en Flex, como además detectar errores que sean obtenidos en la fase tanto de análisis léxico, análisis sintáctico, y además análisis semántico. El compilador debe de ser capaz de construir un árbol que al finalizar la etapa del parser pueda imprimir el árbol y sus errores de errores semánticos (si los posee) de forma completa mostrando así de manera gráfica como este ha sido construido a través producciones.

Para la realización de este problema se deberá utilizar un generador de parsers y scanners, que en nuestro caso es bison que obtiene tokens del scanner que es realizado por Flex. Además se incluye código en C el cual realiza validaciones correspondientes al análisis semántico del parser.

Se deberá de construir el árbol de parser, que podrá poseer errores semánticos o no, lo cual no evitará que finalice el proceso de compilación. De existir errores de este tipo, se deberá indicar al usuario la existencia del mismo.

Además, a pesar de que los generadores sintácticos son capaces de resolver algunos tipos de conflictos, la compilación del programa tiene que ser correcta, sin *warnings* o errores presentes.

Dividir el análisis semántico en un archivo aparte, del resto de código, así como también código de C para la validación de los atributos, tags y contenidos, lo cual ayude a una visibilidad y claridad a la lectura del mismo.

Utilizar recursos suministrados por los generadores tales como variables que permitan encontrar la localización del token en el archivo fuente, sin utilizar librerías que realicen funciones del análisis semantico ni ningún tema relacionado.

Los errores deben de ser impresos en el stderr y la salida del parser en el stdout. Se usará la redirección suministrada desde la consola, y no lectura de archivos, por lo que se requiere la construcción de un make que realice la unión de todos los ficheros necesarios para un correcto funcionamiento.

Diseño del programa

Lista de tags, atributos, contenido validado

Existe un conjunto de tags, atributos y contenido de atributos que son válidos dentro del compilador. Cada tag posee una serie de atributos permitidos que a su vez, estos atributos poseen un formato de contenido. El contenido se dividió en tres grandes conjuntos los cuales denominados text, URL, y num. A Continuación se enlista los elementos validos:

- **a:**
  - charset (NO SOPORTADO EN HTML5)
  - coords (NO SOPORTADO EN HTML5)
  - href URL
  - hreflang TEXT
  - media (HTML5) TEXT
  - name (NO SOPORTADO EN HTML5)
  - rel TEXT
  - rev (NO SOPORTADO EN HTML5)
  - shape (NO SOPORTADO EN HTML5)
  - target TEXT
  - type (HTML5) URL

- Atributos globales

- **b:**
  - Atributos globales
- **blockquote:**
  - cite URL
  - Atributos globales
- **body:**
  - alink TEXT (NO SOPORTADO EN HTML5)
  - background URL (NO SOPORTADO EN HTML5)
  - bgcolor (NO SOPORTADO EN HTML5)
  - link (NO SOPORTADO EN HTML5)
  - text (NO SOPORTADO EN HTML5)
  - vlink (NO SOPORTADO EN HTML5)
- **br:**
  - Atributos globales
- **button:**
  - autofocus (HTML5) TEXT
  - disabled
  - form (HTML5) TEXT
  - fomaction (HTML5) URL
  - formenctype (HTML5) TEXT
  - formmethod (HTML5) TEXT
  - formnovalidate (HTML5) TEXT
  - formtarget (HTML5) TEXT
  - name TEXT
  - type TEXT
  - value TEXT
  - Global Attribute
- **caption:**
  - align TEXT (NO SOPORTADO EN HTML5)

- Global Attribute

- **code**
  - Global Attribute
- **div:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - Global Attribute
- **dl:**
  - Global Attribute
- **dt:**
  - Global Attribute
- **dd:**
  - Global Attribute
- **emt:**
  - Global Attribute
- **form:**
  - accept (NO SOPORTADO EN HTML5)
  - accept-charset TEXT
  - action URL
  - autocomplete (HTML5) TEXT
  - enctype TEXT
  - method TEXT
  - name TEXT
  - novalidate (HTML5) TEXT
  - target TEXT
  - Atributos globales
- **h1-h2-h3-h4-h5-h6:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - Atributos globales
- **head:**
  - profile URL (NO SOPORTADO EN HTML5)
  - Global Attribute

- **hr:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - noshade (NO SOPORTADO EN HTML5)
  - size NUM (NO SOPORTADO EN HTML5)
  - width NUM % (NO SOPORTADO EN HTML5)
  - Global Attribute
- **html:**
  - manifest (HTML5) URL
  - xmlns http://www.w3.org/1999/xhtml
  - Atributos globales
- **img:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - alt TEXT
  - border NUM (NO SOPORTADO EN HTML5)
  - crossorigin (HTML5) TEXT
  - height NUM
  - hspace NUM (NO SOPORTADO EN HTML5)
  - ismap CONTENT
  - longdesc URL (NO SOPORTADO EN HTML5)
  - scr URL
  - usemap #TEXT
  - vspace NUM (NO SOPORTADO EN HTML5)
  - width NUM
  - Global Attribute
- **input:**
  - accept TEXT/*
  - align TEXT (NO SOPORTADO EN HTML5)
  - alt TEXT
  - autocomplete (HTML5) TEXT
  - autofocus (HTML5)
  - checked TEXT
  - disabled TEXT
  - form (HTML5) TEXT
  - formaction (HTML5) URL
  - formenctype (HTML5)

    - formmethod (HTML5) TEXT
    - formnovalidate (HTML5) TEXT
    - formtarget (HTML5) TEXT
    - height (HTML5) NUM
    - list (HTML5) TEXT
    - max (HTML5) NUM
    - maxlength NUM
    - min (HTML5) NUM
    - multiple (HTML5)
    - name TEXT
    - pattern (HTML5) TEXT
    - placeholder (HTML5) TEXT
    - readonly TEXT
    - required (HTML5) TEXT
    - size NUMBER
    - scr URL
    - step (HTML5) NUM
    - type TEXT
    - value TEXT
    - width (HTML5) NUM
    - Global Attribute
- **li:**
    - type TEXT (NO SOPORTADO EN HTML5)
    - value NUMBER
    - Global Attribute
- **link:**
    - charset TEXT (NO SOPORTADO EN HTML5)
    - href URL
    - hreflang TEXT
    - media TEXT

- rel TEXT
- rev TEXT (NO SOPORTADO EN HTML5)
- sizes (HTML5)
- target TEXT (NO SOPORTADO EN HTML5)
- type TEXT
- Global Attribute TEXT

- **meta:**
  - charset (HTML5) TEXT
  - content TEXT
  - http-equiv
  - name TEXT
  - scheme URL (NO SOPORTADO EN HTML5)
  - Global Attribute
- **object:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - archive URL (NO SOPORTADO EN HTML5)
  - border NUM (NO SOPORTADO EN HTML5)
  - classid NUM (NO SOPORTADO EN HTML5)
  - codebase (NO SOPORTADO EN HTML5)
  - codestype (NO SOPORTADO EN HTML5)
  - data URL
  - declare (NO SOPORTADO EN HTML5)
  - form (HTML5) TEXT
  - height NUM
  - hspace NUM (NO SOPORTADO EN HTML5)
  - name TEXT
  - standby TEXT (NO SOPORTADO EN HTML5)
  - type TEXT
  - usemap #TEXT
  - vspace NUM (NO SOPORTADO EN HTML5)

    - width NUM
    - Global Attribute
- **ol:**
    - compact TEXT (NO SOPORTADO EN HTML5)
    - reversed (HTML5) NO TIENE
    - start NUM
    - type TEXT
    - Global Attribute
- **option:**
    - disable TEXT
    - label TEXT
    - selected TEXT
    - value TEXT
    - Atributos globales
- **p:**
    - align TEXT (NO SOPORTADO EN HTML5)
    - Global Attribute
- **pre:**
    - width NUM (NO SOPORTADO EN HTML5)
    - Global Attribute
- **script:**
    - async (HTML5) NO TIENE
    - charset TEXT
    - defer NO TIENE
    - src URL
    - type TEXT
    - xml:space (NO SOPORTADO EN HTML5)
    - Atributos globales
- **span:**
    - Atributos globales

- **strong:**
  - Atributos globales
- **style:**
  - media TEXT
  - scoped (HTML5) NO TIENE
  - type TEXT
  - Atributos globales
- **table:**
  - align TEXT (NO SOPORTADO EN HTML5)
  - bgcolor TEXT (NO SOPORTADO EN HTML5)
  - border NUM
  - cellpadding NUM (NO SOPORTADO EN HTML5)
  - cellspacing NUM (NO SOPORTADO EN HTML5)
  - frame (NO SOPORTADO EN HTML5)
  - rules (NO SOPORTADO EN HTML5)
  - summary TEXT (NO SOPORTADO EN HTML5)
  - width NUM (NO SOPORTADO EN HTML5)
  - Atributos globales
- **td:**
  - abbr TEXT (NO SOPORTADO EN HTML5)
  - align TEXT (NO SOPORTADO EN HTML5)
  - axis (NO SOPORTADO EN HTML5)
  - bgcolor TEXT (NO SOPORTADO EN HTML5)
  - char (NO SOPORTADO EN HTML5)
  - charoff NUM (NO SOPORTADO EN HTML5)
  - colspan NUM
  - headers TEXT
  - height NUM % (NO SOPORTADO EN HTML5)
  - nowrap NUM (NO SOPORTADO EN HTML5)
  - rowspan NUM (NO SOPORTADO EN HTML5)

- scope (NO SOPORTADO EN HTML5)
- valign (NO SOPORTADO EN HTML5)
- width NUM % (NO SOPORTADO EN HTML5)
- Atributos globales

- **th:**
  -
  - abbr TEXT (NO SOPORTADO EN HTML5)
  - align TEXT (NO SOPORTADO EN HTML5)
  - axis (NO SOPORTADO EN HTML5)
  - bgcolor TEXT (NO SOPORTADO EN HTML5)
  - char (NO SOPORTADO EN HTML5)
  - charoff NUM (NO SOPORTADO EN HTML5)
  - colspan NUM
  - headers TEXT
  - height NUM % (NO SOPORTADO EN HTML5)
  - nowrap NUM (NO SOPORTADO EN HTML5)
  - rowspan NUM
  - scope TEXT
  - valign (NO SOPORTADO EN HTML5)
  - width NUM % (NO SOPORTADO EN HTML5)
  - Atributos globales
- **tr:**
  - char (NO SOPORTADO EN HTML5)
  - align TEXT (NO SOPORTADO EN HTML5)
  - bgcolor TEXT (NO SOPORTADO EN HTML5)
  - charoff NUM (NO SOPORTADO EN HTML5)
  - valign TEXT (NO SOPORTADO EN HTML5)
  - Atributos globales
- **textarea:**
  - autofocus (HTML5) NO TIENE

    - cols NUM
    - disabled TEXT
    - form (HTML5) TEXT
    - maxlength (HTML5) NUM
    - name TEXT
    - placeholder (HTML5) TEXT
    - readonly TEXT
    - required (HTML5) TEXT
    - rows NUM
    - wraps (HTML5) TEXT
    - Global Attribute
- **title:**
    - Atributos globales
- **ul:**
    - compact TEXT (NO SOPORTADO EN HTML5)
    - type TEXT (NO SOPORTADO EN HTML5)
    - Atributos globales

Atributos globales

- accesskey TEXT
- class TEXT NUM
- contenteditable (HTML5) TEXT
- contextmenu (HTML5) TEXT
- dir TEXT
- draggable (HTML5) TEXT
- dropzone (HTML5) TEXT
- hidden (HTML5) NO TIENE
- id TEXT NUM _
- lang TEXT
- spellcheck (HTML5) TEXT
- style TEXT NUM _ ; :
- tabindex NUM
- title TEXT
- tanslate (HTML5) TEXT

Los atributos que no son soportados en html5 muestran un error que indica este problema.

Análisis semántico:

Para el diseño del programa buscamos las reglas que se describen en la documentación de la w3c que se encuentra en http://www.w3.org/TR/2000/REC-xml-20001006, la cual describe todas las reglas para la construcción de un documento en formato xhtml el cual fue descrito en el punto anterior.

Para el diseño de nuestro analizador semantico, se utilizó un subconjunto de reglas las cuales permiten validar el conjunto suministrado por parte del profesor como requisitos del programa.

Primeramente se buscan todos los elementos que se encuentran en la w3c que se deben de validar para este programa http://www.w3schools.com/tags/default.asp [1](tags, atributos, formato de contenidos). Esto permitiría verificar que lo que se pide pueda ser construido en un documento de xhtml y validado siguiendo los estándares.

Como segundo paso, se construye una gramática de manera que todos los tokens que son recolectados por el scanner puedan pertenecer al parser, y aquellos elementos de las reglas que no son recolectadas por el scanner, son eliminadas evitando así, construir una gramática que posee elementos inútiles, o que reciba elementos que no se encuentran en sus reglas sintácticas. Además en el análisis sintáctico se encarga de asignar el tipo de nodo, el cual guarda esta información que será utilizada para las validaciones en el momento de análisis semántico.

El tercer paso, consiste en realizar el análisis semántico, específicamente en la parte donde se construye los nodos de los tags que contienen atributos y en la construcción de los atributos. Esto por motivo, que no se realiza una pasada por separado para la realización del mismo, sino que el análisis semántico se realiza de manera conjunta con el análisis sintáctico.

Para la verificación tanto de atributo como de los tag, se utiliza la estructura de arreglo para poder hacer comparaciones con los tokens de entrada del archivo. Cuando se construye tanto el token de atributo como el de tag (solo aquellos que llevan atributo) se invoca una función la cual realiza un recorrido en el arreglo buscando coincidencias con el nodo de entrada. Si no encuentra alguna, o si la encuentra en el arreglo del error especial para los no soportados en html5, imprime el error indicando su número de línea y columna, los cuales fueron recibidos como parámetros de la función gracias a la implementación de la variable yylloc.

En el caso de contenido del atributo, solo existen 3 opciones que son NUM, URL, o TEXT por lo que con una simple implementación a través de if, se recorre el arreglo y se verifica que el atributo este con su correspondiente contenido.

Por otro lado, tenemos el caso cuando es comprobación de tags, que funciona de manera análoga al caso anterior, con la diferencia de unas funciones extra, que permiten recorrer los hermanos atributos, ya que un tag puede contener de 1 a n atributos, siempre y cuando estos sean permitidos para él.

Librerías externas

Para la realización de este programa, solo se utiliza una librería externa para el manejo de cadenas. <string.h> la cual es una librería estándar del lenguaje de programación C que contiene funciones utilizadas en el programa.

Entre las funciones utilizadas, tenemos el strcpy el cual recibe como parámetro 2 punteros de cadena, y copia en la cadena del primer parámetro la cadena que se encuentre en el segundo puntero.

La otra función utilizada es el strcmp que retorna un numero entero mayor, igual o menor a cero dependiendo del resultado de la comparación. Obtendremos un 0 en caso de que las dos cadenas sean del mismo tamaño, en

caso contrario se obtendrá otro número diferente a 0 el cual indicará que las cadenas no son iguales.

Esta librería presenta problema de seguridad con ciertas funciones como por ejemplo el desbordamiento de buffer o como es conocido el buffer overflow, pero en este programa no se implementan strings largos los cuales podrían provocar este tipo de problema.

Análisis de resultados

Se alcanzaron todos los objetivos que fueron propuestos en la asignación. El programa es capaz de reconocer los errores semánticos y no detener el proceso de compilación, pero sí mostrarlo en consola en el caso de que haya. Además, este es capaz reconocer un atributo que no es aceptado en el en html5 e indicarlo como error de este tipo.

Incluso, en los dos casos, es capaz de mostrar el número de fila y columna donde se encuentra el error para que el usuario pueda corregirlo conociendo adonde se presentó el error.

Conclusión personal

La implementación del analizador semántico, nos representó un replanteo de nuestros programas anteriores, ya que esta etapa debe de poseer información más detalla acerca de los tokens, lo cual nos indica que se debe de contar con información tal como contexto para un correcto análisis.

Además, la forma de implementación, que nos muestra un trabajo conjunto de análisis semántico y sintáctico en uno solo, lo cual nos ahorra código ya que después no debemos de recorrer el árbol para buscar la posición y tiempo de compilación gracias que los 2 procesos se realizan de manera conjunta.

También la integración de las herramientas permite el rápido desarrollo de compiladores eficientes, ahorrando tiempo de desarrollo, costo de operaciones entre otros, ya que en pocos días se puede desarrollar un compilador complemente funcional gracias a las herramientas existentes que permiten comprobar que las gramáticas, scanners o análisis semánticos sean correctos y presenten problemas relacionados con el lenguaje creado.

Bibliografía

[1] Refsnes Data, «w3school,» w3school.com, 1999-2013. [En línea]. Available: http://www.w3schools.com/tags/default.asp. [Último acceso: 2013 junio 3].