Modelos de Computación

Lex, localizador de expresiones regulares

Hugo Bárzano Cruz

1. Descripción del problema

El problema que he decidido llevar a cabo es la validación de documentos html con el objetivo de saber si cumplen o no el estándar impuesto para este lenguaje de marcas. Esto se traduce en comprobar si las etiquetas utilizadas son las correctas, además de comprobar su correcta apertura, cierre e indexación.
El segundo punto que voy a tratar en la práctica es la descarga de contenido multimedia desde archivos.html, básicamente la descarga de imágenes.

2. Desarrollo de la práctica

Para realizar la primera parte, correspondiente a la validación, la solución que he tomado es la de crear distintas reglas para la apertura y cierre de las etiquetas que configuran el lenguaje. Una vez creadas estas reglas con el uso de expresiones regulares, lo que hago es utilizar un mecanismo de pila e ir insertando o eliminando elementos siempre y cuando las etiquetas tengan correspondencia

Correcto: <*html*>...</*html*>
Error: <html>...<html> o mtl>...</html>....etc

Para la segunda parte del problema, detecto mediante el uso de expresiones regulares, las imágenes incrustadas en el código html y mediante la llamada al sistema *wget* las descargo. Previamente utilizo funciones auxiliar para limpiar el texto capturado por lex.

3. Funcionamiento

Podemos ejecutar el programa de dos formas distintas.

./ejecutable fichero.html → Comprobará si el contenido de fichero.html es correcto o tiene errores. En caso de tener imágenes las descargará.

./ejecutable -u "www.enlace.com" → Descargará el archivo.html correspondiente a la web dada como argumento, lo validará y en caso de contener imágenes, las descargará. Las comillas son necesarias.

4. Consideraciones del problema

La forma en la que he atacado el problema no es única, en el código están comentadas dos formas de validación complementarias. La primera y más pobre, lo único que hace es comprobar si el numero de etiquetas abiertas y etiquetas cerradas es el mismo. No comprueba que las etiquetas estén bien escritas e indexadas.

La segunda es un mecanismo de doble pila. Utilizo un pila para las etiquetas abiertas y otra para las etiquetas cerradas. Cuando se ha completado el análisis de todo el documento, lo que hago es un "reverse" de una de las dos, por ejemplo las cerradas y voy comparando el top de ambas pilas. Si al acabar la comparación de ambas, todos los elementos han tenido su correspondiente pareja, significa que el documento está bien formado. Con este método se valida el numero de etiquetas, que las etiquetas estén bien escritas y además la indexación. No me ha gustado este método por su ineficiencia ya que para documentos extensos hay que almacenar e iterar bastante sobre las pilas.

5. Compilado

Puesto que hago uso de la stl y otros elementos c++ es necesario compilar mediante g++ lex.yy.c -o ejecutable -lfl