

Tarea 2 - Análisis de Algoritmos y Estructura de Datos

Leonardo Iván Espinoza Ortiz
Universidad de Santiago de Chile
2-2022

1-Introducción

En un documento LaTeX, diferentes partes del texto están limitadas por ciertas etiquetas que alteran la apariencia de un documento. Así, estas etiquetas deben ser emparejadas para limitar su alcance en el documento.

El emparejamiento se realiza a través de una etiqueta que abre y otra que cierra. La forma de una etiqueta que abre es `\begin{<etiqueta>}` y cuando cierra es `\end{<etiqueta>}`. Por ejemplo, un texto centrado puede ser escrito en LaTeX usando la etiqueta *center*, como se muestra a continuación:

```
\begin{center}  
Este es un título  
\end{center}
```

Algunas etiquetas de LaTeX comúnmente utilizadas incluyen:

- *document*: indica el cuerpo del documento que se ocupa una única vez por documento
- *center*: centra horizontalmente un texto en la página, puede ir anidado
- *enumerate*: indica una lista enumerada con elementos indicados por `\item`, puede ir anidado con otras etiquetas
- *itemize*: indica una lista con viñetas con elementos indicados por `\item`, puede ir anidado con otras etiquetas

Además, existe el operador `$` que se abre y cierra sin `/begin` ni `/end`. Todo el texto contenido entre `$` de apertura y `$` de cierre se visualiza en letra itálica

Un ejemplo de un documento LaTeX usando algunas de las etiquetas/operador indicadas, junto con el documento generado se muestra a continuación:

Así, la correcta escritura del código en LaTeX requiere que por cada etiqueta que se abre exista un correspondiente cierre. Además, un cierre debe corresponder siempre a la última etiqueta que se abrió

El objetivo de esta tarea es diseñar una estrategia para validar un documento de LaTeX que determine si el código tiene las etiquetas/operador correctamente emparejadas o no y en la ubicación adecuada . En el caso de determinar que el documento no es válido, se deberá indicar el tipo de error y en qué línea ocurre. Dado que un documento puede tener múltiples errores de emparejamiento, es suficiente que se reporte el primer error encontrado. Por ejemplo, para el caso del código de error anterior debería reportar algo como "Error en línea 4: se esperaba `\end{center}` en lugar de `\end{document}`".

2-Solución Propuesta

La solución propuesta es Diseñar un algoritmo en C, que permita verificar si las etiquetas son válidas o no.

Para eso, el programa se dividirá en dos funciones y el uso de dos TDA:

a-TDAlista

b-TDApila

Se ajustan los TDA lista y pila para la realización de este algoritmo, los nodos se ajustan a datos tipo `char*` y cada función también para su correspondiente desarrollo.

c-función leer archivo por línea

Función que abre el archivo, lee línea por línea y va guardando cada línea de este de forma ordenada a una lista enlazada.

Pseudocodigo:

```
Escribirporlinea(nombreampochivo,Tdalista):Tdalista
Textoenptraa<-fopen(nombreampochivo)
leer(textoenptraa)
insertarInicio(lista,textoenptraa)
while (leer(textoenptraa)<>findelarchivo)
    insertarNodoFinal(lista,textoenptraa)
cerrar archivo(nombreampochivo)
return Tdalista
```

Complejidad: $O(n)$

d-función verificación

Función que recibe una lista enlazada, crea una pila auxiliar y empieza a recorrer la lista enlazada con condicionales:

Si ve una etiqueta Begin, se ingresa a la pila auxiliar.

Si ve una etiqueta End, se verifica si corresponde a la última etiqueta Begin que se abrió. y si es así. Continúa el ciclo y desapila la etiqueta de la pila.

Al final del desarrollo de esta función, si la pila se encuentra vacía, significa que está ordenada de forma correcta.

Si la pila queda con algún elemento dentro de ella, significa que las etiquetas no se encuentran ordenadas de forma correcta.

```
verificacion(tdalista, num): boolean
pila<-crearpilavacia(num)
...se asignan las variables char
...comienza el ciclo while con las condicionales
```

3-Resultados y análisis

Tras realizar análisis y pruebas con distintos archivos de texto, se puede verificar que el desarrollo y planteamiento de la solución está de forma correcta, pero al momento de ingresar a las condicionales ocurre un error.

4- Conclusiones

El objetivo de la tarea se cumplió de forma parcial, ya que no se logró el resultado esperado. Pero el desarrollo de este código iba directo a la solución.

5-Anexo manual de usuario

Para ejecutar el programa, se debe escribir vía línea de comando el nombre del ejecutable, y el nombre del archivo de entrada(en su defecto Entrada1.in)

```
C:\Users\espin\Desktop\Semestre 02-2022\EDA\Tarea 2>test entrada1.in
```

Ejemplo: Nombre Programa Entrada.in