Tarea 2 - Búsqueda en Texto

CC4102 - Diseño y Análisis de Algoritmos
Profesor: Pablo Barceló Auxiliar: Jorge Bahamonde

Fecha de Entrega: 31 de Mayo de 2015

1 Introducción

El objetivo de esta tarea es implementar, evaluar y comparar en la práctica diferentes estructuras para la búsqueda en texto:

- Patricia Trees
- Suffix Tries
- Autómatas para búsqueda de patrones

Nuevamente, se espera que se implementen los algoritmos y se entregue un informe que indique claramente los siguientes puntos:

1. Las *hipótesis* escogidas antes de realizar los experimentos.
2. El *diseño experimental*, incluyendo los detalles de la implementación de los algoritmos, la generación de las instancias y las medidas de rendimiento utilizadas.
3. La *presentación de los resultados* en forma de una descripción textual, tablas y/o gráficos.
4. El *análisis e interpretación* de los resultados.

2 Las Estructuras

Se explicará en detalle el caso de los árboles Patricia, pues las otras estructuras fueron vistas en clases.

Tries y árboles Patricia

Un trie[1] es un árbol en el que cada arista tiene como etiqueta un caracter. De esta forma, si las cadenas están formadas por caracteres de un alfabeto de tamaño σ, un trie que almacena estas cadenas es un árbol σ-ario.

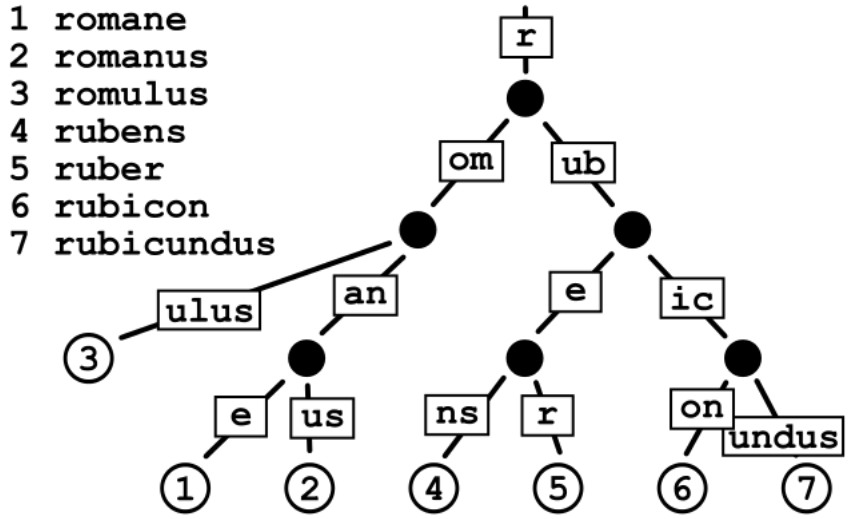
Un problema con esta estructura es que el árbol ocupa siempre un nodo por caracter, pudiendo generar ramas muy largas en el caso de ciertas palabras. Los árboles Patricia[2] buscan remediar esto.

Un árbol Patricia (también llamados *radix trees/tries*) es un trie en el que se han comprimido las ramas unarias, reemplazándolas por arcos. En otras palabras, se reemplazan caminos unarios

[1] Existen diferentes formas de ver los tries y árboles Patricia. En algunas, la palabra completa se encuentra almacenada en las hojas; en otras, los nodos intermedios almacenan subcadenas. Otras variantes utilizan etiquetas numéricas en vez de subcadenas en el caso de los árboles Patricia. Estos cambios alteran ligeramente los algoritmos de inserción y búsqueda. Si decidiera utilizar una variante diferente para sus implementaciones, explicítela en el informe, detallando las diferencias con las versiones aquí expuestas y justificando su decisión.

[2] de PATRICIA: "Practical Algorithm To Retrieve Information Coded In Alphanumeric"

de la forma $N_1 \xrightarrow{c_1} ... \xrightarrow{c_k} N_{k+1}$ por $N_1 \xrightarrow{c_1 ... c_k} N_{k+1}$. Gráficamente, un árbol Patricia tiene la siguiente forma[3]:



Un árbol Patricia tiene una hoja por cada palabra que almacena. Adicionalmente, un árbol de este tipo con n hijos tiene a lo sumo n nodos internos, por lo que ocupa espacio $O(n)$.

Búsqueda

Para buscar una palabra $P[1, m]$ en un árbol Patricia, se navega en éste utilizando las etiquetas de los nodos, comparando las subcadenas de la palabra con éstas. Si se llega a una hoja a la vez que se termina de comparar la palabra que se busca, se ha encontrado el patrón. En caso contrario, no. De esta forma, la búsqueda toma tiempo $O(m)$

Inserción

Para la inserción de una palabra P, se busca en el árbol: la acción que se toma depende de la razón por la que no se ha encontrado:

- Si fue porque se llegó a un nodo que no tiene un hijo que comience con la siguiente letra, se busca una hoja del nodo en el que no se pudo bajar y se reinserta P desde esa hoja.
- Si fue porque se acabó el patrón en un nodo antes de llegar a una hoja, en medio de una arista, se busca una hoja cualquiera del hijo de esa arista y se reinserta P desde ésta.
- Si fue porque se llegó a una hoja, se reinserta P desde esta hoja.

Reinserción desde una hoja

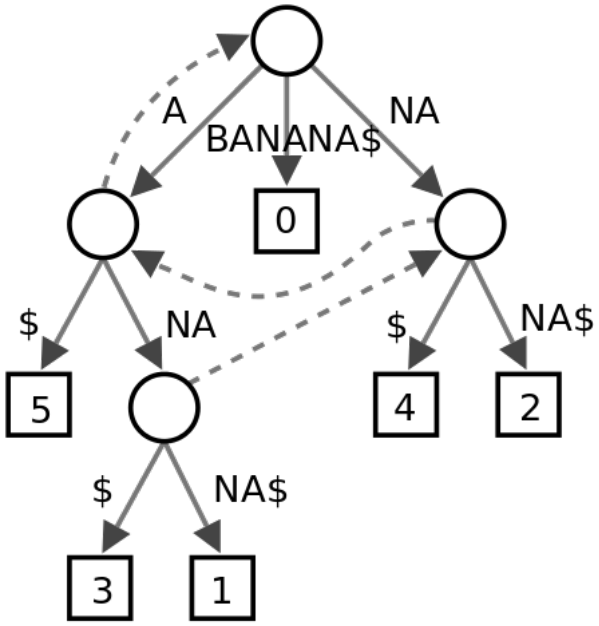
Esta subrutina se utiliza en ciertos casos de inserción, como ya se mencionó. Se tiene una palabra P que se desea insertar en la estructura, y una hoja a la que se llegó, que representa una palabra P'

- Se compara P con P', encontrándose p, el prefijo máximo común.
- Se entra al árbol desde la raíz, buscándose p.
  - Si p se encontró a la mitad de una arista, se corta la arista con un nodo de dos hijos: la hoja P y el hijo original de la arista.

[3]Fuente: `http://en.wikipedia.org/wiki/Patricia_tree`

– Si p se encontró en un nodo, se agrega P como nuevo hijo de ese nodo.

Un suffix trie puede ser visto como un árbol Patricia para un texto, donde las palabras ingresadas son los sufijos de éste. En las hojas del árbol se almacenan las posiciones en que comienza cada sufijo. A continuación se muestra el suffix tree para la palabra BANANA. Las flechas punteadas son *suffix links* utilizados en la construcción de la estructura [4]:



3 Pruebas y Datos

Escoja un conjunto de al menos 5 libros [5] cuyo tamaño en texto plano supere los 2 MB. Preprocese los libros que utilizará, de modo que la entrada de sus estructuras sólo contenga espacios y letras minúsculas (elimine saltos de línea, puntuación, lleve todo a minúsculas, etc). Para cada libro, construya cada una de las estructuras mencionadas, documentando los tiempos de construcción.

Para cada texto, escoja $N/10$ palabras de forma aleatoria de éste y encuentre todas las ocurrencias de éstas en el texto, utilizando cada una de las estructuras. Para el árbol Patricia, considere el texto como un conjunto de palabras que almacena en el árbol. De esta forma, puede almacenar en cada hoja las posiciones de las ocurrencias de cada palabra como una lista. En el caso de los suffix tries y los autómatas, considere el texto completo como una gran cadena. En este caso, el buscar una palabra a es buscar las subcadenas "a" (con los espacios, salvo en el caso particular de la primera y última palabras del texto).

Repita estos procesos (construcción y búsqueda) para obtener promedios confiables para los tiempos registrados. Mida tanto el tiempo de construcción como los tiempos de búsqueda.

Note que tiene libertad para escoger los textos que utilizará: de esta forma, puede escogerlos para ayudarse a poner a prueba su hipótesis, si fuera necesario.

4 Entrega de la Tarea

- La tarea puede realizarse en grupos de a lo más 2 personas.

[4] Fuente: `http://en.wikipedia.org/wiki/Patricia_tree`

[5] Si fuera necesario, concatene libros, pero cuide que no perturbe su experimento: por ejemplo, que estén en el mismo idioma :)

- No se permiten atrasos.
- Para la implementación puede utilizar `C`, `C++`, `Java` o `Python`. Para el informe se recomienda utilizar LaTeX.
- Escriba un informe claro y conciso. Las ponderaciones del informe y la implementación en su nota final son las mismas.
- La entrega será a través de U-Cursos y deberá incluir el informe junto con el código fuente de la implementación (y todas las indicaciones necesarias para su ejecución).

5 Links

- En `http://www.gutenberg.org` puede encontrar una gran cantidad de documentos.
- En `http://www.artamene.org/` puede encontrar *Artamène le Grand Cyrus*, la novela más larga escrita (en francés).