Práctica PRO2

Primavera 2019

26 de abril de 2019

1. Introducción

Queremos disponer de un programa para codificar y decodificar textos escritos en diversos idiomas. La codificación que buscamos estará compuesta exclusivamente por ceros y unos.

En lo que afecta a este trabajo, un idioma se define por un identificador (`string`) y una tabla de frecuencias (una colección de dos o más caracteres o símbolos, cada uno de ellos con su correspondiente frecuencia en el idioma).

El método propuesto se basa en asignar un pequeño código (en concreto, una `string` formada por ceros y unos) a cada carácter del idioma correspondiente, de forma que bajo ciertas condiciones dicha asignación sea reversible, es decir, que si se detecta dicho código en una `string` más grande se pueda obtener el carácter original.

Con el objeto de conseguir algunas buenas propiedades, por ejemplo que los caracteres más frecuentes tengan asignados códigos más cortos, la asignación se realiza de la siguiente manera, para un idioma I cualquiera de N caracteres, $N \geq 2$:

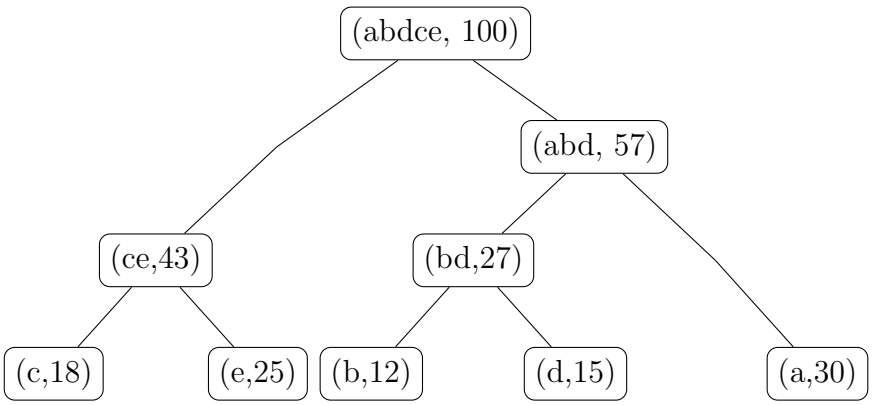
- se construyen N árboles binarios de pares <`string, int`>, cada uno formado por un único nodo, que representa un carácter de I y su frecuencia
- dichos árboles se reducen a uno (A, llamado árbol de códigos o `treecode` de I), aplicando $N-1$ veces los siguientes pasos:
  - sean $a1$ y $a2$, respectivamente, el árbol con raíz más pequeña y el árbol con la segunda raíz más pequeña respecto al siguiente criterio: si los enteros no son iguales, es más pequeña la raíz con el entero más pequeño; en caso de empate, es más pequeña la raíz con la string más pequeña en orden lexicográfico
  - los árboles $a1$ y $a2$ se eliminan y se sustituyen por un tercero (b) cuya raíz contiene la suma de los enteros de las raíces de $a1$ y $a2$ y la concatenación de sus strings (primero, la más pequeña); los subárboles de b son $a1$, como subárbol izquierdo, y $a2$, como subárbol derecho
- se asigna a cada carácter x el código y formado por el camino en A desde la raíz hasta x (0=izquierda, 1=derecha)

A partir de dicha asignación $(x_1, y_1), ..., (x_N, y_N)$, un texto se codifica simplemente sustituyendo cada carácter x_i por su correspondiente código y_i.

Debido a otra buena propiedad de este método, un texto codificado se decodifica descomponiéndolo en trozos consecutivos que coincidan con caminos de la raíz a una hoja de A. La decodificación de cada uno de estos trozos es el carácter de dicha hoja. Si esta descomposición no es posible, significa que el texto no se puede decodificar en el idioma. En caso de que sí sea posible, el método garantiza que el resultado es único.

Ejemplo: consideremos un idioma cuya tabla es `(a 30) (b 12) (c 18) (d 15) (e 25)`. Respecto a los conceptos que acabamos de introducir podemos decir lo siguiente:

- El `treecode` resultante es



- Los códigos resultantes son `a: 11; b: 100; c: 00; d: 101; e: 01`
- Un texto `baacddecc` se codifica como `100111100101101010000`
- Un texto `asdfase` no pertenece al idioma
- Un texto codificado como `001001110001101` se decodifica como `cbabed`
- Un texto codificado como `10101010` no se puede decodificar en el idioma

2. Operaciones

Tanto los nombres de los idiomas como los textos (sin codificar) pueden contener cualquier carácter de los que aparecen en un teclado QWERTY español standard (por ejemplo, los de los laboratorios de la FIB), más los que se pueden conseguir acentuando vocales, con la excepción de los caracteres invisibles (blanco, tabulador, salto de línea) y del símbolo €. En consecuencia, los textos estarán formados por una sola línea (de hecho, una sola `string`).

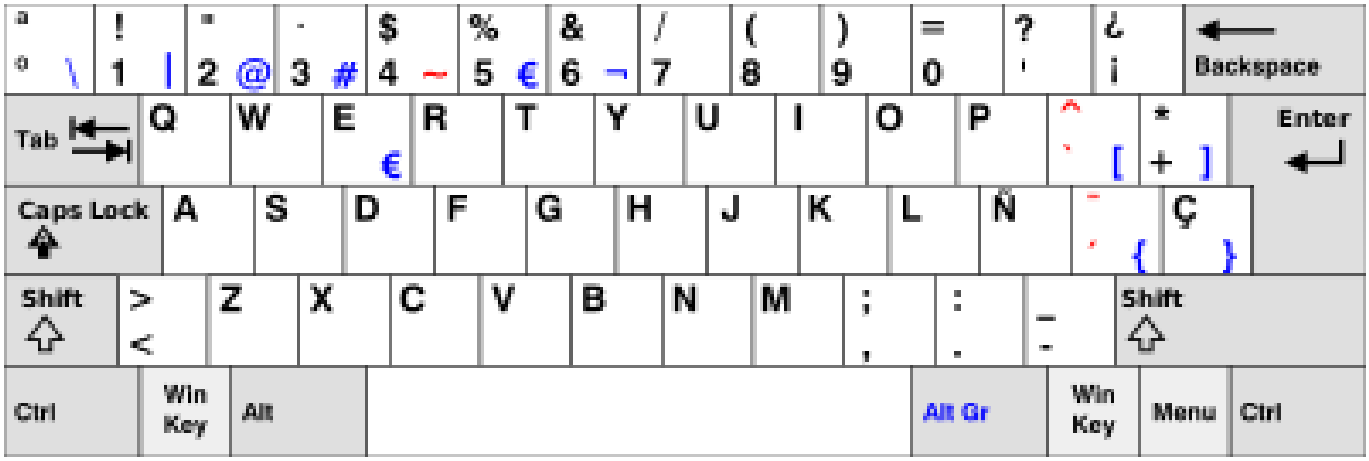


Figura 1: Teclado QWERTY español (fuente: Wikipedia)

El programa siempre comenzará con la lectura de una colección inicial de idiomas. Se leerá un entero n y a continuación n idiomas (cada uno con su identificador y su tabla). Después el programa permitirá ejecutar las siguientes funcionalidades.

- `añadir/modificar idioma`. A partir de un nombre y una tabla de frecuencias, si no existe ya un idioma con ese nombre, se añade un idioma nuevo a partir de dichos datos; en otro caso, se suman a la tabla del idioma existente las frecuencias de la nueva tabla y, por lo tanto, se ha de obtener un nuevo `treecode` y sus correspondientes códigos para el idioma

- `codifica`. Se da un nombre de idioma y un texto. Si el texto pertenece al idioma, se obtendrá su codificación; si no es así o el idioma no existe, el programa nos informará de esta situación
- `decodifica`. Se da un nombre de idioma y un texto formado exclusivamente por ceros y unos. Si el texto procede de la codificación de un texto del idioma, se obtendrá su decodificación; si no es así o el idioma no existe, el programa nos informará de esta situación
- `consultar tabla de frecuencias`. Se da un nombre de idioma y, si el idioma existe, se escribe su tabla de frecuencias; en caso contrario, el programa nos informará de esta situación
- `consultar treecode`. Se da un nombre de idioma y, si el idioma existe, se escribe su `treecode`; en caso contrario, el programa nos informará de esta situación
- `consultar códigos`. Se da un nombre de idioma y, si el idioma existe, se escriben sus códigos; en caso contrario, el programa nos informará de esta situación
- `acabar`. Sin datos. Cesa la ejecución del programa

Para ver los detalles concretos conviene estudiar el juego de pruebas público.