SOLUCIONANDO PROBLEMAS CON EL PARADIGMA FUNCIONAL(PROYECTO PRIMERA PARTE)

Lenguajes de Programación

VARGAS ARENAS PEDRO - 201734553
RAMOS LOPEZ LIZBETH - 201749275
TEPOZ ROMERO BELEN – 201770060

03/03/2021

BUAP | FCC.

Historia sobre el código Morse

En 1836, Samuel Morse, José Henry y Alfredo Vail inventaron el telégrafo. En aquel entonces sólo podían transmitir impulsos eléctricos de duración breve (puntos) y larga (líneas). El señor Vail desarrolló un sistema de codificación que mezclaba puntos y líneas para poder transmitir letras y algunos signos de puntuación. A la hora de asignar un código a cada letra se basó en la frecuencia con la que aparecía cada letra en textos de su lengua materna, el inglés.

De modo que a las letras que más aparecían le asignó códigos más cortos, es decir, con menos signos y preferiblemente puntos. Por ejemplo, las dos letras que más aparecen en los textos ingleses son la E y luego la T. Así que a la E le asignó un único punto y a la T una única líneas.

Claves mnemotécnicas

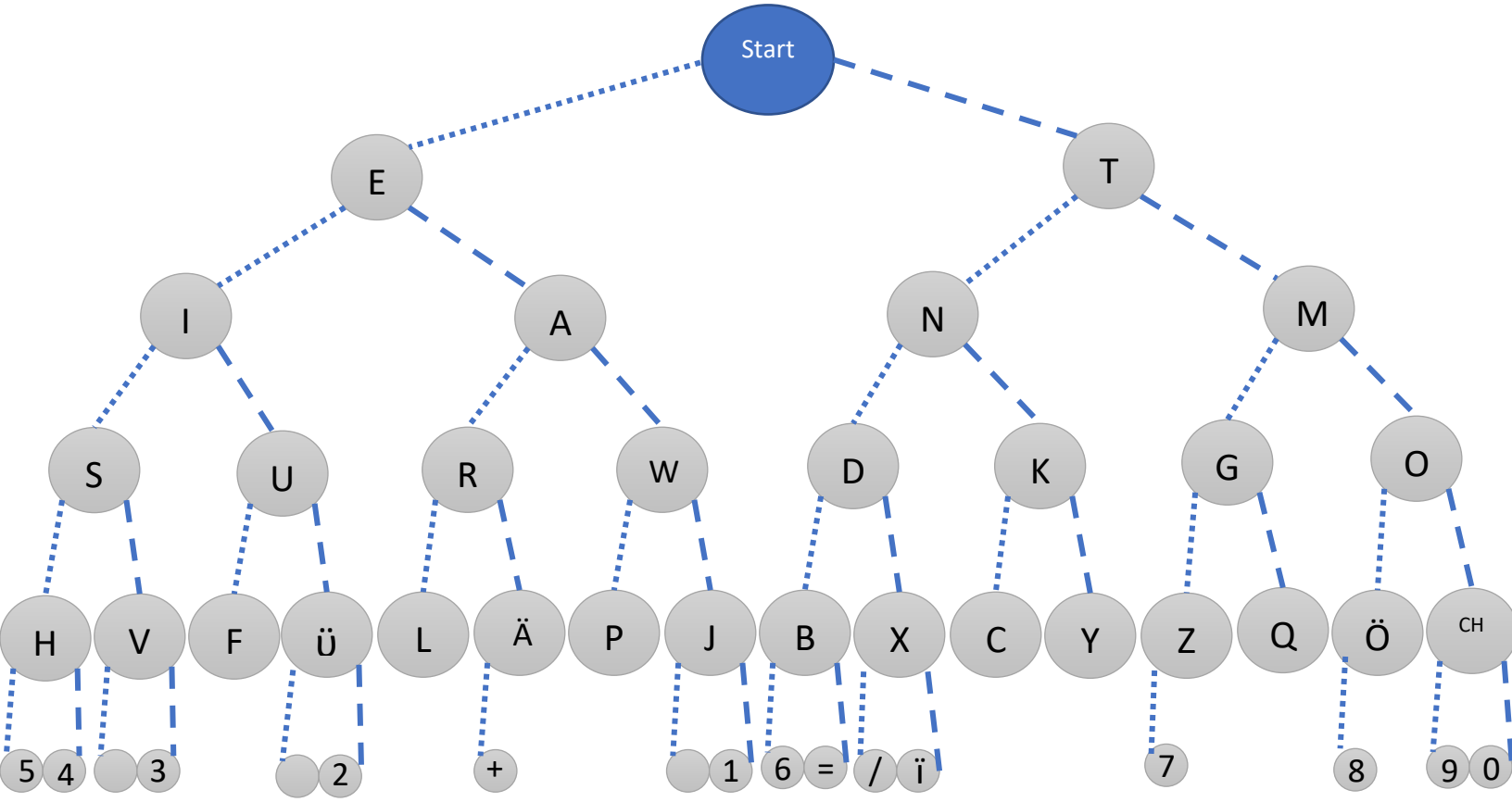
Si queremos codificar un mensaje en Morse debemos conocer los signos de cada letra del alfabeto. Una forma relativamente sencilla de aprender la secuencia de puntos y líneas para cada palabra son las claves mnemotécnicas:

- A cada letra del alfabeto se le asigna una palabra que empieza por esa misma letra.
- Cada clave tiene tantas sílabas como signos tenga la letra.
- Las sílabas con la vocal O, se corresponden con líneas.
- Las sílabas con cualquier otra vocal se corresponden con puntos.
- Se puede usar cualquier palabra que cumpla con esas cuatro condiciones.

Letra	Código	Letra	Código	Letra	Código
A	• –	N	– •	**1**	•----
B	– • • •	Ñ	– – • – –	**2**	••– – –
C	– • – •	O	– – –	**3**	••• – –
CH	– – – –	P	• – – •	**4**	•••• – •
D	– • •	Q	– – • –	**5**	•••••
E	•	R	• – •	**6**	– ••••
F	• • – •	S	• • •	**7**	• • –––
G	– – •	T	–	**8**	–––––
H	• • • •	U	• • –	**9**	––––•
I	• •	V	• • • –	**0**	–––––
J	• – – –	W	• – –	**Ä**	• – • –
K	– • –	X	– • • –	**ï**	– • • –•
L	• – • •	Y	– • – –	**Ÿ**	•• – –
M	– –	Z	– – • •	Ö	– – – •

Árbol de decodificación

Para decodificar un mensaje en Morse, un método muy rápido consiste en usar el siguiente árbol binario:



1. Se decodifica letra a letra, empezamos siempre por el círculo grande (Start).
2. Si el primer signo es un punto, tomamos la rama de la izquierda, si es una línea, la de la derecha.
3. Hacemos lo mismo con el segundo signo, si es un punto, vamos a la izquierda, si es una línea, a la derecha.
4. Repetimos hasta que no nos quedan más signos, el círculo donde nos quedemos es la letra en cuestión.

Por ejemplo, tenemos que decodificar la siguiente letra: · - - ·

- ✓ Empezamos en el círculo grande.
- ✓ El primer signo es un punto, tomamos la rama izquierda: E (·)
- ✓ El segundo signo es una raya, tomamos la rama derecha: A (· -)
- ✓ El tercer signo es una raya, tomamos la rama derecha: W (· - -)
- ✓ El último signo es un punto, tomamos la rama izquierda: P (· - - ·)
- ✓ Nuestra letra es la P.

Justificación del uso del paradigma funcional para solucionar el problema:

El paradigma funcional se basa en la idea central de que la programación es mediante funciones, como lo indica su nombre, ahora bien, esto nos brinda gran robustez a nuestro código ya las funciones son independientes y no comparten variables, esto se debe a la transparencia referencial que es cuando la evaluación de una expresión depende única y exclusivamente de sus parámetros de entrada, esto significa que invocar una función con los mismos argumentos siempre produce el mismo resultado, por lo tanto, siempre que quedamos decodificar una palabra varias veces el resultado será el mismo y no hay una función externa que lo modifique.

Este paradigma se basa en no ciclos y da paso al desarrollo de funciones recursivas, donde se retorna la salida en función a las entradas (los parámetros que recibe como argumento la función)

Tipo de evaluación se realiza.

Como primer ejemplo tenemos la siguiente función:

```
palabra.reverse.chop.reverse -> Llamada por nombre
palabra.chop.reverse
palabra.reverse
palabra -> Resultado final de la Evaluación.
```

Observemos que la evaluación se realiza por llamada por nombre ya que iniciamos a evaluar nuestra función desde el extremo y sucesivamente hacia dentro, así es como nos retornaría el resultado final de la función.

Como segundo ejemplo tenemos la siguiente función:

```
decodificar(palabra.reverse.chop.reverse, nodoActual.hijoIzq) -> Orden
                                                                 Aplicativo
```

Como iniciamos por resolver lo que se encuentra dentro del cuerpo de la función lo mas que se pueda y así hasta llegar al extremo de nuestra función, la evaluación se realiza en Orden Aplicativo.

Funcionamiento del sistema con dos ejemplos de entradas distintas.

Al ejecutar el programa pedirá el código en Morse que necesita ser codificado.

```
Bienvenido al decodificador de código Morse
Ingresa la palabra a decodificar (Deja un espacio entre letra y dos entre palabras)
```

En caso de que sea más de una palabra la que se quiere decodificar es necesario que se escriba con cierto formato:

Dejar un espacio entre letras y dejar dos espacios entre palabras. Como primer ejemplo tenemos el siguiente texto

B U E N D I A 0 5 / 0 3 / 2 0 2 1

```
-···␣··-␣·␣-·␣␣-··␣··␣·-␣␣-----␣·····␣-··-·␣-----␣···--␣-··-·␣··---␣-----␣··---␣·----
```

```
PS C:\Users\btepo\Documents\LenguajesdeProgramacion> ruby decodificadorMorse.rb
Bienvenido al decodificador de código Morse
Ingresa la palabra a decodificar (Deja un espacio entre letra y dos entre palabras)
-... ..- . -.  -.. .. .-  ----- ..... -..-. ----- ...-- -..-. ..--- ----- ..--- .----
Tu palabra decodificada es: BUEN DIA 05/03/2021
```

Como segundo ejemplo tenemos el siguiente texto (siguiendo el formato descrito en el primer ejemplo)

L E N G U A J E S D E P R O G R A M A C I O N

```
·-·· · -. --. ..- .- .--- · ···  -·· ·  ·--. ·-· --- --. ·-· ·- -- ·- -·-· ·· --- -·
```

```
Bienvenido al decodificador de código Morse
Ingresa la palabra a decodificar (Deja un espacio entre letra y dos entre palabras)
.-.. . -. --. ..- .- .--- . ...  -.. .  .--. .-. --- --. .-. .- -- .- -.-. .. --- -.
Tu palabra decodificada es: LENGUAJES DE PROGRAMACION
```

Ventajas y desventajas de emplear el paradigma funcional:

Ventajas	Desventajas
Código más preciso y más corto	No apto para todas las tareas
Los programas no tienen estados	Los datos (por ejemplo, las variables) no se pueden modificar
Muy adecuados para la paralelización	No se permite el acceso eficiente a grandes cantidades de datos
Código fácilmente verificable, incluso las funciones sin estado se pueden verificar	No es adecuado para muchas recursiones de la misma pila
Fácil de combinar con la programación imperativa y orientada a objetos	La programación recurrente puede dar lugar a errores graves