

## Examen Extraordinario I de Programación

### Curso 2018-2019

# Cadena Común Mínima

**NOTA:** Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Un **alfabeto**  $V$  es un conjunto de símbolos (letras) a partir de las cuales se puede formar un conjunto de cadenas llamado **vocabulario**. Un **patrón**  $p$  sobre un alfabeto  $V$  es una cadena sobre el conjunto de símbolos  $V \cup \{?, +, *\}$ . Este permite describir un conjunto de cadenas del vocabulario de  $V$  sin enumerar explícitamente sus elementos. En un patrón  $p$  cada símbolo  $i \in V$  representa dicho carácter, mientras que los símbolos  $?$ ,  $+$  y  $*$  son comodines.

- $?$  : Representa la presencia de 1 símbolo de  $V$ .
- $+$  : Representa la presencia de 1 o más símbolos de  $V$ .
- $*$  : Representa la presencia de 0 o más símbolos de  $V$ .

Para denotar al conjunto de cadenas descritas por un patrón  $p$  sobre un alfabeto  $V$  se utiliza la notación  $L(p, V)$ , también conocido como **lenguaje del patrón**. El lenguaje de un patrón puede ser tanto un conjunto infinito de cadenas, como un conjunto finito.

Por ejemplo, dado el alfabeto  $V = \{a, b, c\}$ . Los siguientes patrones sobre  $V$  generan los lenguajes:

PATRÓN		LENGUAJE	EJEMPLOS
$p_1 = aa?b$	$\rightarrow$	$L(p_1, V) = \{aaab, aabb, aacb\}$	
$p_2 = b + c$	$\rightarrow$	$L(p_2, V) = \{\text{cadenas de 3 o más caracteres que comienzan con } b \text{ y terminan en } c\}$	$bbc, baaaac, bcbc$
$p_3 = *a*$	$\rightarrow$	$L(p_3, V) = \{\text{cadenas de 1 o más caracteres que contienen al menos una } a\}$	$a, abcbcbc, ba, bbbab$
$p_4 = *$	$\rightarrow$	$L(p_4, V) = V^*$ (todas las cadenas del vocabulario de $V$ , incluida la cadena vacía)	
$p_5 = ?c?c$	$\rightarrow$	$L(p_5, V) = \{\text{cadenas de 4 caracteres que tienen } c \text{ en la segunda y cuarta posición}\}$	$acac, bccc, cccc, ccbc$

Luego, se desea, dado dos patrones  $p_x$  y  $p_y$  sobre el mismo alfabeto  $V$ , determinar la cadena de más corta longitud  $w$  tal que  $w \in L(p_x, V)$  y al mismo tiempo  $w \in L(p_y, V)$ .

**NOTA:** Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe completar la implementación del método **CadenaComunMinima** en el *namespace* **Weboo**. Examen que pertenece a la clase **Patrones**. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public static class Patrones
    {
        public static string CadenaComunMinima(string p1, string p2, string alfabeto)
        {
            throw new NotImplementedException();
        }
    }
}
```

El método **CadenaComunMinima** deberá **retornar** aquella **cadena de menor longitud  $w$** , tal que pertenezca a los conjuntos de los **lenguajes generados** por los patrones  **$p_1$**  y  **$p_2$**  sobre el alfabeto **alfabeto**. **De no existir** dicha cadena, el método deberá retornar **null**, y en caso de existir varias de la misma longitud mínima, retornar alguna de ellas.

## EJEMPLOS

```
string alfabeto = "abc";

string p11 = "a*";
string p12 = "+ab*";
string ejemplo1 = Patrones.CadenaComunMinima(p11, p12, alfabeto);
/*
 * --- EJEMPLO#1: "aab" -> ES UNICA ---
 */
Console.WriteLine(ejemplo1);

string p21 = "?*?";
string p22 = "+*+";
string ejemplo2 = Patrones.CadenaComunMinima(p21, p22, alfabeto);
/*
 * --- EJEMPLO#2: "aa" -> NO ES UNICA ---
 */
Console.WriteLine(ejemplo2);

string p31 = "abc";
string p32 = "?*?";
string ejemplo3 = Patrones.CadenaComunMinima(p31, p32, alfabeto);
/*
 * --- EJEMPLO#3: "abc" -> ES UNICA ---
 */
Console.WriteLine(ejemplo3);
```

```
string p41 = "c++";
string p42 = "*****";
string ejemplo4 = Patrones.CadenaComunMinima(p41, p42, alfabeto);
/*
 * --- EJEMPLO#4: "caa" -> NO ES UNICA ---
 */
Console.WriteLine(ejemplo4);

string p51 = "*+?";
string p52 = "ba*";
string ejemplo5 = Patrones.CadenaComunMinima(p51, p52, alfabeto);
/*
 * --- EJEMPLO#5: "baa" -> NO ES UNICA ---
 */
Console.WriteLine(ejemplo5);

string p61 = "*";
string p62 = "*";
string ejemplo6 = Patrones.CadenaComunMinima(p61, p62, alfabeto);
/*
 * --- EJEMPLO#6: "" -> ES UNICA ---
 */
Console.WriteLine(ejemplo6);

string p71 = "a?b";
string p72 = "??";
string ejemplo7 = Patrones.CadenaComunMinima(p71, p72, alfabeto);
/*
 * --- EJEMPLO#7: null -> NO EXISTE ---
 */
Console.WriteLine(ejemplo7);

string p81 = "+";
string p82 = "ab+";
string ejemplo8 = Patrones.CadenaComunMinima(p81, p82, alfabeto);
/*
 * --- EJEMPLO#8: "aba" -> NO ES ÚNICA ---
 */
Console.WriteLine(ejemplo8);
```

Usted **puede** asumir que:

- El **string** **alfabeto** nunca será la cadena vacía, no tendrá caracteres repetidos, todos serán letras minúsculas y no contendrá ni símbolos ni números.
- Los **string** **p1** y **p2** nunca serán vacíos.

Usted **no puede** asumir que:

- Siempre va a existir al menos una cadena que pertenezca a los conjuntos de los lenguajes generados por los patrones **p1** y **p2** sobre el alfabeto **alfabeto**.

**NOTA:** Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará.