

Desconcatenación mediante matrices evolutivas

notas

Fernando Galindo Soria

Julio del 2012

Uno de los problemas clásicos de la lingüística y en general del reconocimiento de patrones es el de encontrar los componentes de un objeto (palabras de una oración), y aunque existen múltiples métodos tiende a ser ineficientes, en ese documento planteo un método basado en la combinación de 2 herramientas desarrolladas a mediados de los 80, la primera se basa en la búsqueda de cadenas que se repiten dentro de grandes cadenas y la segunda es el uso de matrices evolutivas.

La búsqueda de cadenas en el primer caso consiste en tomar una cadena grande, por ejemplo

mgasdfgasdhjkasmhjdefg

y buscar dentro de ella cadenas que se repiten.

Por facilidad a la cadena la veremos como una oración de algún lenguaje y la representaremos mediante la regla de reescritura

$S \rightarrow \text{mgasdfgasdhjkasmhjdefg}$

Donde S se conoce como *Axioma* y representa al sistema u objeto que se esta analizando y la flecha $\rightarrow$ indica que S se puede *rescribir* o sustituir por la cadena

Existen muchos métodos para realizar estas búsquedas pero los mas generales parten de buscar las cadenas mas grandes posibles y dentro de ellas seguir buscando las subcadenas mas grandes posibles que se repiten, y por el otro extremos se tiene los métodos que buscan las cadenas mas pequeñas posibles y luego las combinan para obtener cadenas cada vez mas grandes.

En este documento, y con el fin de poder mostrar la aplicación con las matrices evolutivas, mostraremos uno de los métodos que se han desarrollado para encontrar cadenas buscando primero las subcadenas mas pequeñas.

Entonces dada la cadena

$S \rightarrow \text{mgasdfgasdhjkasmhjdefg}$

Empezamos de izquierda a derecha y buscamos la cadena mas pequeña que aparece al menos dos veces, encontrando que **as** se repite varias veces:

$S \rightarrow \text{mgasdfgasdhjkasmhjdefg}$

Inventamos un nuevo símbolo A y dondequiera que aparece **as** lo sustituimos por A, quedando

S -> mgAdfgAdhjkAmhjdefg
A -> as

Siguiendo el proceso encontramos que Ad se repite, por lo que lo sustituimos por B, quedando:

S -> mgBfgBhjkAmhjdefg
B -> Ad
A -> as

Y así se sigue mientras sigan encontrándose cadenas repetidas, sustituyendo en este caso **fg** por C y **hj** por D, quedando al final la siguiente representación gramatical:

S -> mgBCBDkAmDdeC
B -> Ad
A -> as
C -> fg
D -> hj

Donde S, A, B, C, D se conocen como *elementos no terminales* y representan a las cadenas que se repiten.

El ejercicio continua para encontrar cadenas mas representativas, para lo cual excepto en el axioma las demás regla se expanden, por ejemplo en este caso en la regla

B -> Ad

Se puede sustituir A por as (porque se tiene la regla A -> as), quedando

B -> asd

Con lo que queda:

S -> mgBCBDkAmDdeC
B -> asd
A -> as
C -> fg
D -> hj

El problema con este método es que puede requerir una gran cantidad de comparaciones (en el peor de los casos del orden n^2 por lo que si el objeto tiene por ejemplo 100 elementos se pueden requerir de orden de diez mil comparaciones, pero si se tienen 1000 elementos se requiere del orden de 1 millón de comparaciones)

Aquí es donde intervienen las matrices evolutivas

Se tiene originalmente una matriz M con tantas entradas como elementos básicos se manejen (por ejemplo en texto en código ASCII se manejan 256 elementos, por lo que la matriz sería de 256 por 256) llena de ceros

Se va leyendo la secuencia de elementos (caracteres) de la cadena a analizar
Por ejemplo si se tiene la cadena

mgasdfgasdhjkasmhjdefg

Como los dos primeros elementos son **m** y **g** se suma 1 a la entrada que corresponde a la intercepción del renglón **m** y la columna **g**, para indicar que existe una secuencia **m** y **g**
 $M(m, g)++$;

Lo mismo se hace con la intercepción del segundo y tercer elemento **g** y **a**

$M(g, a)++$;

Y así para todos los elementos de la cadena

Al final queda una matriz donde las entradas con valor 0 indican que esas letras no aparecen juntas en las cadenas, las entradas con 1 indican que esos elementos solo aparecen una vez juntos en la cadena y las entradas con un valor de 2 o mas indican que existen al menos dos repeticiones y esas son las combinaciones que se tiene que sustituir por los elementos no terminales.

Se introducen tantos renglones y columnas como elementos no terminales se crearon en este ciclo, y se repite el proceso mientras se tengan cadenas repetidas.