REFERENCIAS DISTANCIALES DE LEVENSHTEIN EN LA ESTRUCTURA DE BURKHARD-KELLER ORGANIZADA SEGUN LA DISTANCIA INVARIANTE TRASPOSICIONAL. PARTE I.

AUTORES: SANTANA, O.; PEREZ, J.; ESPINO, M.; RODRIGUEZ, J.C.
Departamento de Informática y Sistemas
Universidad Politécnica de Canarias
Apto.: 550. Las Palmas de Gran Canaria. España.

RESUMEN:

Para la búsqueda de las cadenas más similares en el sentido de la Distancia Direccional de Levenshtein, **DD**, en este trabajo se propone la introducción de referencias **DD** en la estructura de Burkhard_Keller, organizada según la Distancia Invariante Trasposicional, **DIT**, a fin de poder realizar pruebas de **DD**_candidatura, basadas en la desigualdad triangular, con el propósito de disminuir los cálculos de **DD** y así mejorar la realización global del esquema **BK_DIT+DD creciente**, [SP89a].

0.- INTRODUCCION:

El problema de la detección y corrección de errores, en cadenas de caracteres, constituye un tema ya clásico que ha sido objeto de estudio por bastantes autores. Un sistema detector_corrector, detecta cadenas erróneas para tratar de encontrar la cadena correcta más parecida. Este problema contiene elementos de reconocimiento de modelos y de teoría de la codificación, siendo factible su solución gracias a la redundancia intrínseca del diccionario.

Alberga, [AL67], usó matrices binarias para evaluar una medida de la disimilitud entre dos cadenas. Szanser, [SZ73a,b], desarrolló un proceso matemático de coincidencia elástica que era efectivo en un 95%. Morgan, [MO70], realizaba un o_exclusivo entre dos cadenas para determinar si había ocurrido un único error simple. Wagner y Fisher, [WF74], desarrollaron un algoritmo de programación dinámica que puede determinar la distancia entre dos cadenas, **DD**, medida como el mínimo coste de la secuencia de operaciones de edición (sustitución, extracción e inserción). Ukkonen, [UK83], ideó una forma de cálculo eficiente de la matriz de diferencias de Levenshtein, [LE66], usando sus diagonales. Landau_Vishkin, [LV85a,b,86a,b], Galil_Giancarlo, [GG86], y el propio Ukkonen, [UK85], utilizan en sus trabajos, con las adaptaciones necesarias, la computación eficiente de la distancia de edición introducida en [UK83], Tanaka_Kojima [TK87] continúan utilizando la introducida por Wagner_Fisher, [WF74].

Cada cadena puede considerarse como un punto en el espacio multidimensional de secuencias de caracteres, donde no todas las secuencias de caracteres son posibles. Los métodos para la recuperación de datos multidimensionales permiten llevar a cabo búsquedas asociativas, del tipo que interesan a este trabajo, teniendo que acceder tan sólo a una porción reducida de la base de datos en cuestión. Los antecedentes de los algoritmos actualmente más desarrollados se encuentran en los primeros trabajos de Rivest, en los que se exploran desmenuzamientos y técnicas digitales; Bentley y Finkel, quienes proponen los árboles **QUAD** y los **K_D** que son estructuras basadas en la comparación de claves; y Burkhard y Keller que presentan la estructura **BK** que se organiza a partir de las distancias.

Se ha definido una nueva distancia, [SD87], que se ha denominado Distancia Invariante Trasposicional, **DIT**, debido al hecho de que su valor no depende de las operaciones de trasposición a que pueda ser sometida una cadena. La importancia de la distancia **DIT**, si bien no puede usarse por si sóla para la determinación de las cadenas más similares, deviene de la circunstancia de que su valor entre

dos cadenas es siempre inferior o igual a la **DD** entre estas dos mismas cadenas, siendo su coste computacional sensiblemente inferior; lo cual puede ser aplicado para la construcción de un filtro adaptivo **DIT/DD** que tenga por misión reducir el número de cadenas de la base de datos a las que se calcula la **DD**, con la cadena de búsqueda. Sólo aquellas cadenas de la base de datos tales que su **DIT** a la cadena de búsqueda sea inferior o igual a la **DD** mínima alcanzada son **DD**_candidatas y sólo ellas han de sufrir el cálculo de **DD** con la cadena de búsqueda; además esa **DD** mínima alcanzada se redefine cada vez que una cadena **DD**_candidata posea una **DD** que sea inferior al radio de búsqueda.

Se ha abordado en la estructura de Burkhard_Keeler organizada según **DIT**, **BK_DIT**, el tema de las listas de las componentes de **DIT** que penden de cada nodo, [SP89b]. La idea que se ha llevado a efecto es la de organizar estas listas de tal forma que puedan ser compartidas, por más de una cadena, cada una de las componentes. El objetivo que se persigue es que, en virtud de esta compartición, se aprovechen cálculos parciales de **DIT** realizados durante la búsqueda actual en la visita a otros nodos de la estructura **BK_DIT**. El estudio experimental llevado a cabo revela una mejor realización, en los esquemas decrecientes, cuando se introduce esta innovación, sobre todo para distorsiones altas.

Se ha realizado el estudio de una nueva estrategia de búsqueda en una estructura **BK_DIT**, siguiendo un esquema de búsqueda **BK_DIT+DD creciente**, [SP89a], en el que el radio de búsqueda va en aumento hasta encontrar respuesta; ya que se puede garantizar que si ésta es encontrada lo será completamente, es decir, estarán presentes todas las cadenas más similares. Tal esquema de búsqueda ha resultado más eficiente que los esquemas decrecientes, [SP88], estudiados.

En este trabajo se propone la introducción de referencias **DD** en la estructura de Burkhard_Keller, organizada según la Distancia Invariante Trasposicional, **DIT**, a fin de poder realizar pruebas de **DD**_candidatura, basadas en la desigualdad triangular, con el propósito de disminuir los cálculos de **DD** y así mejorar la realización global del esquema **BK_DIT+DD creciente**, [SP89a].

En la seccion 1 se define el concepto de holgura relativa y se sugiere la introducción de nuevos argumentos discriminatorios a fin de disminuir los cálculos de **DD** durante el proceso de búsqueda. En la sección 2 se plantean las **DD** relaciones como herramienta discriminadora; presentándose, en sus tres subsecciones, diferentes grados de incorporación de **DD**_referencias. Los resultados experimentales y conclusiones aparecen en la sección 3.

1.- HOLGURA RELATIVA DD/DIT:

El aumento del número de **DD**s calculadas, con la distorsión, no sólo se debe al incremento de la multiplicidad de la respuesta, como consecuencia de un radio de búsqueda más grande, sino que, en mayor medida, es debido al aumento de la

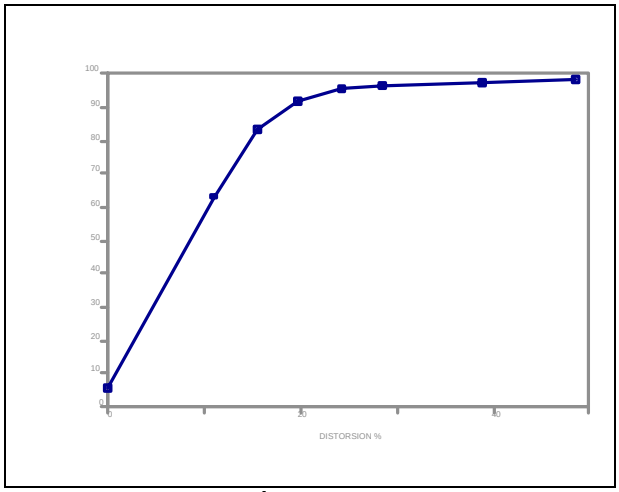


Figura 1

holgura relativa, **HR**, del filtro **DIT/DD**, figura 1. **HR** es la relación entre el número de cadenas DD_candidatas que no forman parte de la respuesta y el número total de cadenas **DD**_candidatas, respecto de sus **DIT**s.

La evolución desfavorable de **HR** con la distorsión, sugiere la introducción de otros argumentos discriminatorios, auxiliares al filtro **DIT**, que puedan reducir el número de cadenas a las que se ha de calcular **DD**.

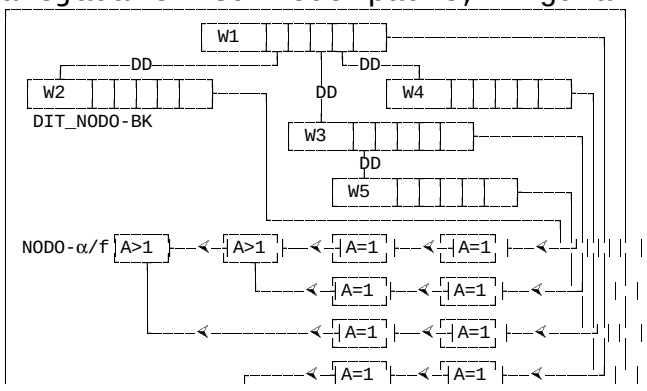
2.- REFERENCIAS DISTANCIALES, DD:

La actuación de la distancia **DD**, en su interrelación con la distancia **DIT**, en los esquemas de búsqueda estudiados hasta ahora, es unicamente algorítmica. Ya que el costo de cálculo de **DD** es alto, comparado con el de **DIT**, sería interesante plantearse la siguiente cuestión: ¿Puede **DD** intervenir, además, estructuralmente a fin de disminuir el número de cadenas **DD**_candidatas?.

La implantación de referencias **DD**, en la estructura **BK_DIT+DD**, proporciona una herramienta adicional apreciable en el sentido de aumentar la potencia de filtrado, haciendo uso de la desigualdad triangular con las **DD**s.

2.1.- REFERENCIA AL NODO PADRE:

Una primera aproximación consiste en almacenar en cada nodo la **DD** entre el mismo y la cadena alojada en su nodo padre, figura 2.



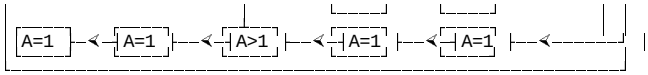


Figura 2

Con ello se puede realizar una prueba de **DD**_candidatura, que va perdiendo significación a medida que se va alejando del nodo en el que se calcula la **DD** con la cadena de búsqueda, pudiendo llegar a ser inútil.

Sea el nodo **i+k**, figura 3, en el que se encuentra la cadena $\mathbf{W_{i+k}}$ a la que se va a realizar la prueba de **DD**_candidatura. Considérese que el nodo **i** es el antecesor más cercano al **i+k**, tal que, se ha calculado anteriormente la **DD** entre $\mathbf{W_i}$ y la cadena de búsqueda **CB**. Para poder realizar esta prueba es necesario que tal nodo **i** exista.

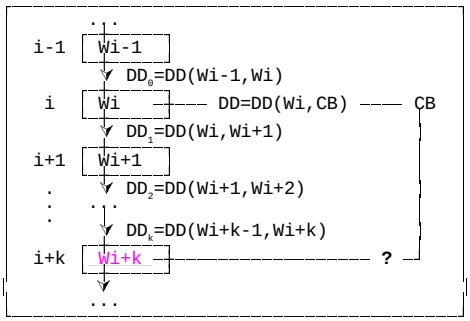


Figura 3

La prueba a realizar será:

Si $\mathbf{|DD-DD_1|-DD_2-DD_3-...-DD_k > DDM}$ entonces $\mathbf{W_{i+k}}$ **no** es **DD**_candidata.

La justificación de esta prueba se basa en la desigualdad triangular para la **DD**, en el Apéndice A se exponen los argumentos.

2.2.- REFERENCIAS A LOS ANCESTROS:

Una incorporación de referencias **DD** más abundante y por tanto conducente a pruebas de **DD**_candidatura más finas, consiste en almacenar en cada nodo, las **DD**s entre dicho nodo y todos sus antecesores.

Sean los nodos, antecesores del nodo **k**, $\mathbf{i_1, i_2, ..., i_n}$, figura 4, en

los que se ha calculado la **DD** entre las cadenas correspondientes, $W_{i1}, W_{i2}, \ldots, W_{in}$, y la cadena de búsqueda, **CB**. Para poder realizar la prueba de **DD**_candidatura de W_k, alojada en el nodo **k**, es necesario que **n≥1**.

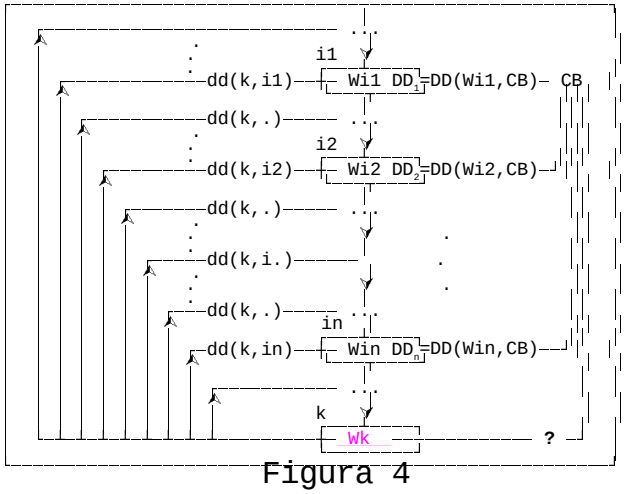


Figura 4

La prueba a realizar será:

Si existe algún **r**ε{**1,...,n**} tal que $|DD_r - dd(k,i_r)| > DDM$ entonces W_k **no** es **DD**_candidata.

La justificación de esta prueba se basa en la desigualdad triangular para la **DD**, los argumentos se exponen en el Apéndice B.

2.3.- REFERENCIAS A TODOS LOS HERMANOS POR LA IZQUIERDA EN CADA NIVEL:

Pueden seguirse ampliando las referencias **DD**, con lo que se obtendrán pruebas de **DD**_candidatura aún más finas. Una posibilidad interesante consiste en introducir, para cada nodo, referencias a sus hermanos izquierdos además de a todos sus ancestros y hermanos izquierdos de éstos. La idea que se persigue es aumentar el número de referencias en la zona previamente recorrida durante el proceso de búsqueda, por esta razón y teniendo en cuenta que en el esquema de búsqueda creciente la zona a explorar está determinada a priori por el radio de búsqueda, es posible forzar el esquema de búsqueda para que se explore de izquierda a derecha.

Los nodos referenciados, en cada nodo, se numeran de forma ascendente desde el nodo raíz. Los hijos de éste, se numeran consecutivamente de izquierda a derecha hasta alcanzar el ramal por el que se ha de descender para llegar al nodo origen de las referencias. El proceso se repite, en cada nuevo nivel, hasta acceder a dicho nodo, $I_1+I_2+\ldots+I_k$ en la figura 5.

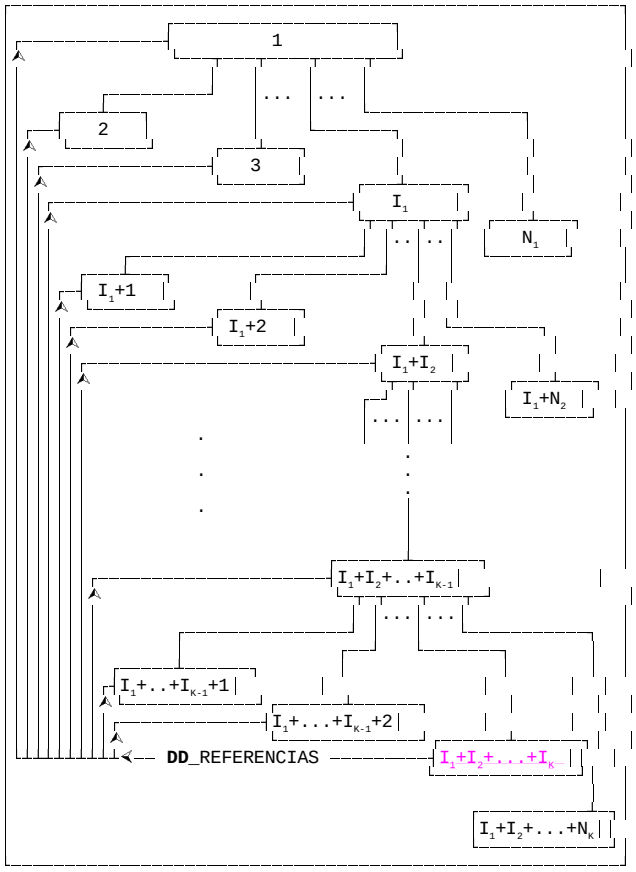


Figura 5

La prueba a realizar será similar a la planteada en la sección 2.3; con la salvedad de que las **DD**s que se tienen en cuenta, durante el proceso de búsqueda, corresponden a

los nodos **DD**_referenciados aquí, en lugar de sólo a los ancestros.

3.- RESULTADOS EXPERIMENTALES Y CONCLUSIONES:

La introducción de argumentos **DD**_discriminatorios, basados en referencias **DD**, produce una mejora más notoria en la realización de los esquemas de búsqueda crecientes que en los decrecientes, debido a que las pruebas **DD** mencionadas son mucho más eficientes cuando los radios de búsqueda son menores.

En el esquema de búsqueda

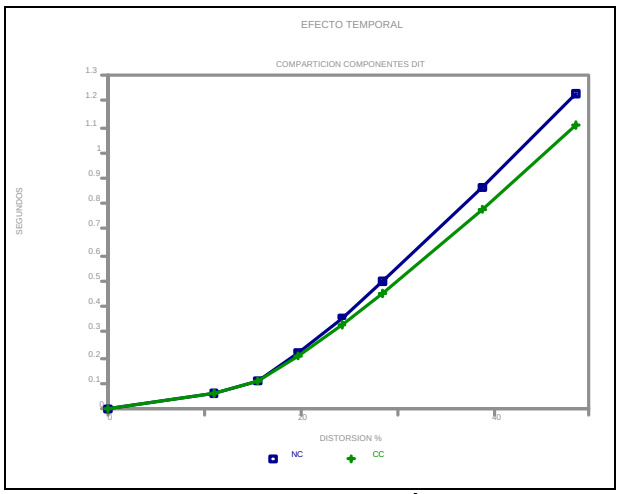


Figura 6

BK_DIT+DD creciente, la compartición de los nodos_α/f de las listas que soportan las componentes de **DIT** (esquema **CC**) promueve una mejora en la realización, respecto al que no

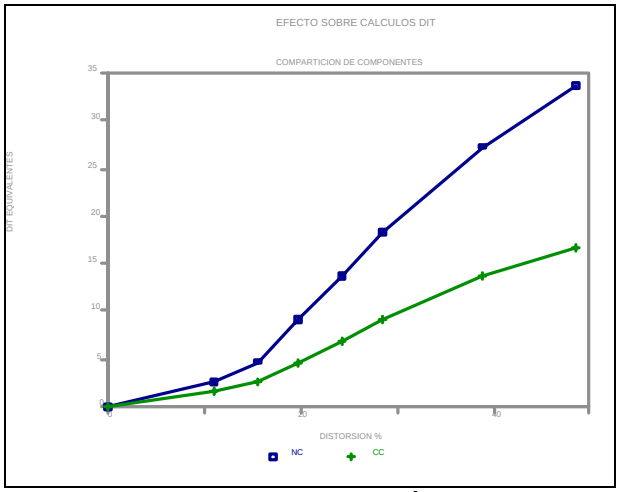


Figura 7

tiene en cuenta esta optimización (esquema **NC**) que se pone más de manifiesto en cuanto aumenta la distorsión, figura 6, de igual forma que en el esquema de búsqueda **decreciente**, [SP89b]. Esta reducción temporal viene determinada esencialmente por la disminución de calculos de **DIT**, figura 7, aunque algo perjudicada por la mayor complejidad inherente a sus cálculos. El aprovechamiento de la compartición es más relevante al aumentar la distorsión, debido a que éste aumenta al hacerlo el número de cadenas a las que se calcula **DIT**. A partir de este punto se usarán la estructura y esquema de búsqueda que hacen uso de la compartición de

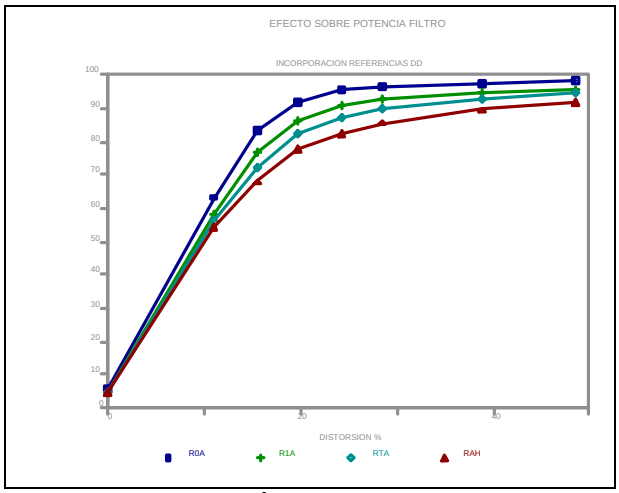


Figura 8

componentes de **DIT**.

Cuando se usa el esquema **BK_DIT+DD creciente** con **compartición de nodos_α/f**: sin **DD**_referencias, **R0A**; con **DD**_referencia al nodo padre, **R1A**; a todos los ancestros, **RTA**; y a los hermanos izquierdos, ancestros y hermanos izquierdos de éstos, **RAH**; la holgura relativa, **HR**, se reduce, figura 8. Ello se debe a que, con la progresiva inclusión de **DD**_referencias, disminuye el número de cadenas a las que se ha de evaluar la distancia direccional.

Sin embargo la realización global de estos esquemas no responde a esta pauta, figura 9, ya que la

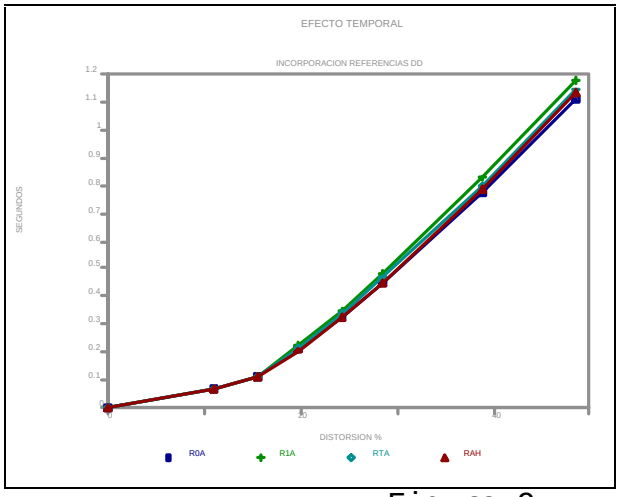


Figura 9

disminución en el número de **DD**s que se han de calcular no compensa el costo computacional inherente a la realización de las pruebas **DD**, en el caso de los esquemas **R1A** y **RTA**. No obstante, el esquema **RAH** tiene una realización mejor que cualquiera de los otros, llegando a superar para distorsiones bajas (no superiores al 25%) la realización del esquema **R0A**.

Estos comportamientos pueden explicarse en base a la relación entre el costo de las pruebas de **DD**_candidatura y la eficacia de las mismas en cuanto a reducción de los cálculos de **DD**s.

Como conclusión, podemos decir que la introducción de referencias distanciales de Levenshtein puede ser rentable si es lo suficientemente abundante como para que las pruebas de **DD**_candidatura sean eficaces, sin que el costo de las mismas llegue a ser excesiso en demasía. Convendría por tanto continuar en la línea de introducción de un mayor número de referencias.

APENDICE A

A partir de la desigualdad triangular para la **DD**, como:

$DD(CB,W_i) \leq DD(CB,W_{i+k}) + DD(W_{i+k},W_{i+k-1}) + \ldots + DD(W_{i+1},W_i)$.

Teniendo en cuenta la notación de la figura 3, se deduce que:

$DD(CB,W_{i+k}) \geq (DD-DD_1)-DD_2-\ldots-DD_k$ {I}

Por otra parte:

$DD(W_i,W_{i+k}) \leq DD(W_i,CB) + DD(CB,W_{i+k}) \Longrightarrow$

$\Longrightarrow DD(CB,W_{i+k}) \geq DD(W_i,W_{i+k}) - DD(CB,W_i)$ {*}

Y como:

$DD(W_i,W_{i+1}) \leq DD(W_i,W_{i+k}) + DD(W_{i+k},W_{i+k-1}) + \ldots + DD(W_{i+2},W_{i+1}) \Longrightarrow$

$DD(W_i,W_{i+k}) \geq DD_1 - DD_2 - DD_3 - \ldots - DD_k$ {**}

Y a partir de {*} y {**}:

$DD(CB,W_{i+k}) \geq (DD_1 - DD) - DD_2 - DD_3 - \ldots - DD_k$ {II}

Y de {I} y {II} se deduce que:

$DD(CB,W_{i+k}) \geq |DD - DD_1| - DD_2 - DD_3 - \ldots - DD_k$

Y por tanto:

Si $\mathbf{|DD - DD_1| - DD_2 - DD_3 - \ldots - DD_k > DDM}$ $\Longrightarrow DD(CB,W_{i+k}) > DDM$, con lo que $\mathbf{W_{i+k}}$ **no** es **DD**_candidata.

APENDICE B

A partir de la desigualdad triangular para la **DD** y como para cualquier cadena **W** se verifica que:

$DD(W,CB) \leq DD(W,W_k) + DD(W_k,CB) \Longrightarrow DD(W_k,CB) \geq DD(W,CB) - DD(W_k,W)$

$DD(W_k,W) \leq DD(W_k,CB) + DD(CB,W) \Longrightarrow DD(W_k,CB) \geq DD(W_k,W) - DD(W,CB)$

De lo cual se deduce que:

$DD(CB,W_k) \geq |DD(W,CB) - DD(W_k,W)|$ para cualquier cadena W.

Por tanto, para todo $r\varepsilon\{1,...,n\}$ se verifica, siguiendo la notación introducida en la figura 4, que: $DD(CB,W_k) \geq |DD_r - dd(k,i_r)|$

De lo que se infiere que **si existe $r\varepsilon\{1,...,n\}$** tal que:

$|DD_r - dd(k,i_r)| > DDM$ entonces **W_k no** es **DD**_candidata.

Bibliografía:

[AL67] ALBERGA, C.N.: "String Similarity and Misspellings". Comm. ACM., Vol. 10 (5), 302/313, (1967).

[GG86] GALIL, Z.; GIANCARLO, R.: "Improved String Matching with k Mismatches", SIGACT News, 17, 4, 52/54, (1986).

[LE66] LEVENSHTEIN, V.I.: "Binary Codes Capable of Correcting, Insertions and Reversals". Soviet Phys. Dokl. 10, 707/710, (1966).

[LV85a] LANDAU, G.M.; Vishkin, U.: "Efficient String Matching in the Presence of Errors". Proc. 26th IEEE FOCS, 126/136, (1985).

[LV85b] LANDAU, G.M.; VISHKIN, U.: "Efficient String Matching with k Differences". TR_36/85, Department of Computer Science, Tel Aviv University, Submitted for Journal Publication, 1985.

[LV86a] LANDAU, G.M.; VISHKIN, U.: "Efficient String Matching with k Mismatches", Theoretical Computer Science, 43, 239/249, (1986).

[LV86b] LANDAU, G.M.; VISHKIN, U.; NUSSINOV, R.: "An Efficient String Matching Algorithm with k Differences for Nucleotide and Amino Acid Sequences". Nucleic Acid Research 14 (1), 31/46, (1986).

[MO70] MORGAN, H.L.: "Spelling Correction in System Programs". CACM., Vol. 13, 2, 90/94, (1970).

[SD87] SANTANA, O.; DIAZ, M.; MAYOR, O.; REYES, J.: "Esquemas y estructura para la búsqueda de las palabras más similares a una dada". XIII Conferencia Latinoamericana de Informática, Vol. II, 1169/1189, (1987).

[SP88] SANTANA, O.; PEREZ, J.; LOPEZ G.; RODRIGUEZ, G.: "La estructura de Burkhard_Keller en la búsqueda de las cadenas más similares a una dada". XIV Conferencia Latinoamericana de Informática, (1988).

[SP89a] SANTANA, O.; PEREZ, J.; RODRIGUEZ, J.C.: "Increasing Radius Search Schemes for the Most Similar Strings on the Burkhard_Keller Tree". International Workshop on Computer Aided Systems Theory, EUROCAST'89, (1989).

[SP89b] SANTANA, O.; PEREZ, J.; HERNANDEZ, Z; RODRIGUEZ H., G.: "Sharing of Transposition_Invariant Distance, DIT, on DIT_organized Burkhard_Keller Structure in Searchs for Best Matching Strings". Submitted to 1990 IEEE International Conference on Computer Systems and Software Engineering, Tel-Aviv, Israel, (1990).

[SZ73a] SZANSER, A.J.: "Bracketing Technique in Elastic Matching". Comput. J., Vol. 16, 2, 132/134, (1973).

[SZ73b] SZANSER, A.J.: "Automatic Error Correction of Natural Text: Part 1". Computer Science, No. 46. National Physics Laboratory, England, (1973).

[TK87] TANAKA, E.; KOJIMA, Y.: "A high speed string correction method using a hierarchical file". IEEE Trans. Pattern Anal. Mach. Intell. Vol. PAMI-9, 6, 806-815, (1987).

[UK83] UKKONEN, E.: "On Approximate String Matching". Proc. Int. Conf. Found. Comp. Theor., Lecture Notes in Computer Science 158, Springer_Verlag, 487/495, (1983).

[UK85] UKKONEN, E.; "Finding Approximate Pattern in Strings". J. of Algorithms, 6, 132/137, (1985).

[WF74] WAGNER, R.A.; FISCHER, M.J.: "The String_to_String Correction Problem". JACM, 21 (1), 168/173, (1974).