(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 426 327**

(51) Int. CI.:

G06F 17/22 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **27.05.2010 E 10005512 (8)**

(97) Fecha y número de publicación de la concesión europea: **10.07.2013 EP 2390793**

(54) Título: **Procedimiento para determinar la similitud de porciones de texto**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
22.10.2013

(73) Titular/es:

CÓDICE SOFTWARE S.L (100.0%)
Parque Tecnológico de Boecillo, Edificio Centro, Oficina 103
47151 Boecillo Valladolid, ES

(72) Inventor/es:

RUIZ, ARROYO, BORJA y
SANTOS LUACES, PABLO

(74) Agente/Representante:

MILTENYI, Peter

DESCRIPCIÓN

PROCEDIMIENTO PARA DETERMINAR LA SIMILITUD DE PORCIONES DE TEXTO

La presente invención proporciona un procedimiento para determinar si una primera porción de texto debe considerarse como incluida en una segunda porción de texto, en el que ésta puede estar incluida en una forma modificada o no modificada. Este procedimiento puede ser implementado en una herramienta de soporte a un usuario en diferentes versiones de combinación de un documento de base electrónica.

El proceso de edición de un documento electrónico por lo general incluye componer diferentes versiones del documento. Por ejemplo, un escritor del documento puede, en un momento dado, decidir reestructurar su versión actual al mismo tiempo que no desea abandonarla definitivamente. Por lo tanto, puede que desee mantener la versión antigua para un posible uso posterior. En consecuencia, existirían en paralelo una versión más nueva y una versión más antigua del documento electrónico. En otro ejemplo, varias personas pueden colaborar en la edición de documentos, posiblemente cada una modificando una versión base del mismo, componiendo así una respectiva versión propia del documento electrónico. Por ejemplo, cada uno de los compositores podría suprimir o añadir una porción de texto de la o a la versión base, o puede mover una porción a otra posición en el documento y así sucesivamente. En particular, en el campo de la ingeniería de software, el desarrollo concurrente y paralelo realizado por varios contribuyentes se ha convertido en un concepto básico de la producción de código de software.

En estos casos, las diversas versiones creadas están representadas cada una de ellas en un respectivo documento electrónico. Estos documentos pueden ser diferentes el uno del otro en el hecho de que las porciones de texto que aparecen en un primer documento no aparecen en un segundo documento, o aparecer en una formulación modificada, o aparecer en otra posición del segundo documento con respecto al primero, posiblemente además en una formulación modificada.

Se sabe que las operaciones de combinación son para conciliar los respectivos documentos y combinarlos en uno solo. Para este fin, se realiza un análisis de las diferencias. En una combinación dual, se comparan dos versiones de un archivo entre sí con el fin de generar una combinación. En una combinación triple, se considera además un archivo original común de las dos versiones.

En algunos casos, dicha operación de combinación se puede realizar de forma automática. Sin embargo, hay muchos casos en los que se requiere una intervención manual. En particular, los escenarios en los que el texto de un primer documento es modificado y movido en comparación con una segunda versión, por lo general terminan en una operación de combinación manual. Por ejemplo, este tipo de situaciones aparecen con frecuencia cuando se utiliza la refactorización de software, la cual se centra en la modificación de código existente para mejorar su legibilidad y por lo tanto su mantenimiento, sin necesidad de cambiar el comportamiento de la aplicación externa.

Con el fin de identificar dicho texto movido que ni siquiera podría ser completamente idéntico a su contraparte en otro documento, se debe realizar una evaluación de la similitud entre las porciones de texto. Un objeto de la presente invención es proporcionar un procedimiento para determinar si una primera porción de texto debe ser considerada como incluida en una segunda porción de texto, en el que (si es así) ésta puede estar incluida en una forma no modificada o modificada. Otro objeto de la invención es mejorar las herramientas automáticas de combinación mediante la implementación de dicho procedimiento en las mismas, con el fin de reducir la incidencia de la necesidad de intervenciones manuales, o para asistir y guiar a un usuario en un proceso manual de combinación.

Estos objetos se consiguen con el procedimiento de la reivindicación 1 y el medio legible informáticamente de la reivindicación 14. En las reivindicaciones dependientes se describen realizaciones preferidas.

US 2009/0089754 (ZEIDMAN) 2 de Abril de 2009 describe un procedimiento para comparar dos documentos con el fin de identificar plagio de software.

El procedimiento de la reivindicación 1 determina si una primera porción de texto debe ser considerada o no como incluida en una segunda porción de texto, en el que puede ésta puede estar incluida en una forma no modificada o modificada o de lo contrario diferente, y en el que las porciones de texto están codificadas y estructuradas electrónicamente para tener un respectivo número de una o más líneas. El número puede ser predeterminado.

En este documento, el que una primera porción de texto está "incluida" en una segunda porción de texto debería recoger el significado de que la primera porción de texto es en realidad la totalidad de la segunda porción de texto. Además, el término "similitud" (y sus derivados) ha de entenderse como que incluye el significado de igualdad (o sus derivados). Es decir, cualesquiera dos entidades clonadas también se denominan en este documento como similares.

De acuerdo con el procedimiento de la reivindicación 1, se selecciona una línea L_i de la primera porción de texto y una línea Z_j de la segunda porción de texto, en el que tanto la línea L_i como la línea Z_j incluyen al menos una palabra. En un ejemplo particular, la línea L_i puede incluir múltiples palabras tales como dos, tres o más palabras. Adicional o alternativamente, la línea Z_j puede incluir dos, tres o más palabras.

En este documento, una palabra se define como uno o varios caracteres consecutivos en un texto escrito informáticamente, que están entre dos caracteres consecutivos de un conjunto predeterminado de uno o más caracteres especiales, o que están entre el inicio de una línea y el primero del (o de los) carácter(es) especial(es) de la línea, o entre el último del (o de los) carácter(es) especial(es) de la línea y el final de la línea o, si en la línea no aparece carácter especial alguno, entre el inicio y el final de la línea. El (o los) carácter(es) especial(es) (que se utilizan como carácter(es) de separación de palabras) no forman parte de la palabra o palabras. Por ejemplo, si el conjunto de caracteres especiales se define compuesto de los caracteres ä, ü y ñ, entonces la línea de texto escrita "hereäareñitheüwordsä!" estaría compuesta de las palabras "here", "are", "the", "words" y "!".

El conjunto de estos caracteres especiales puede incluir el carácter de espacio y/o signos de puntuación como la coma, el punto, el punto y coma, los dos puntos, el signo de exclamación, etc. Adicional o alternativamente, el conjunto de caracteres puede incluir diferentes tipos de paréntesis y/o signos de interrogación y/o signos de igualdad y/o signos de desigualdad, por ejemplo. El procedimiento puede incluir la etapa de recibir una selección del conjunto de caracteres especiales. En una forma de realización preferida, el conjunto de caracteres se compone únicamente del carácter de espacio " ".

La notación de "dos" (o más) palabras significa que las cadenas respectivas aparecen como diferentes representaciones de texto, como por ejemplo en dos documentos electrónicos (por ejemplo, guardados en diferentes lugares o en diferentes posiciones de un documento electrónico). No se pretende que esto signifique que las palabras como tales tienen un aspecto o significado diferente. Se supone que se hace una comparación de palabras y la determinación de si son iguales por parejas. Es decir, si una palabra w se compara con varias palabras $w_1,..., w_t$ entonces se determina para cada $j = 1,..., t$ si w es igual o no a w_j.

Como se mencionó anteriormente, se tienen en cuenta las porciones de texto para incluir una segmentación fija del texto incluido en unas entidades. Estas entidades se considera que son líneas. Las líneas de las porciones de texto, es decir, la segmentación de cada una de las porciones de texto en líneas, pueden ser independientes de las respectivas representaciones de las porciones de texto en uno o más visualizador(es). Por ejemplo, las líneas pueden estar definidas en una respectiva codificación interna de las porciones de texto. Las líneas pueden estar separadas por un carácter de retorno de carro. Sin embargo, son concebibles otras particiones diferentes de las definidas por líneas, tales como, por ejemplo, segmentos definidos por el número de palabras que incluyen.

El procedimiento puede incluir la etapa de recibir una selección de la primera y segunda porciones de texto, y/o de un algoritmo predeterminado que se va a utilizar.

La etapa de selección de líneas L_i y Z_j puede incluir un paso de búsqueda de uno o más carácter(es) o palabra(s) especial(es) incluido(s) en las líneas de la primera y/o segunda porción de texto. Si se encuentra (o no se encuentra) un carácter o una palabra (tales caracteres o palabras, respectivamente), puede(n) elegirse para su selección otra línea L_i o Z_j, respectivamente, u otras líneas L_i y Z_j. Por ejemplo, si ambas porciones de texto incluyen código fuente de software, la etapa de seleccionar las líneas L_i y Z_j puede comprender la etapa de comprobar si el primer (y posiblemente también el segundo, tercer y/o cuarto) carácter(es) o palabra(s) de la línea L_i y/o Z_j están o no incluidos en un conjunto predeterminado de uno o más caracteres o palabras de excepción, tales como caracteres que indican un comentario que no debe ser interpretado por un compilador. Si es así, se puede seleccionar otra línea L_i o Z_j, respectivamente, u otras líneas L_i y Z_j. De esta manera, cuando se determina si la primera porción de texto está incluida o no en la segunda porción de texto, pueden ignorarse los comentarios de una primera y/o segunda porción

de texto. El procedimiento puede incluir la etapa de recibir una selección del conjunto de caracteres (o palabras) de excepción.

Alternativamente, todas las líneas de la primera y/o segunda porción de texto pueden ser considerados de la misma manera, por ejemplo, ignorando si incluyen código fuente o uno o más comentarios.

De acuerdo con la presente descripción, un procedimiento implica la aplicación de un algoritmo predeterminado para la tupla de líneas seleccionadas L_i y Z_j. El algoritmo compara cada palabra de la línea L_i con una o algunas o cada palabra(s) de la línea Z_j y determina, en cada caso, si las palabras comparadas por parejas respectivamente son ambas iguales. Mediante la utilización del resultado de esta determinación de igualdad, el algoritmo calcula al menos un valor de resultado (es decir, uno o más valores de resultado).

Como ejemplo, se puede modificar un algoritmo para el cálculo de una distancia de *Levenshtein* de la siguiente manera. Convencionalmente, la distancia de *Levenshtein* entre una primera y una segunda cadena se define como el número mínimo de ediciones necesarias para convertir la primera cadena en la otra, en el que las operaciones de edición permitidas son inserción, borrado y sustitución de un solo carácter. Un algoritmo de *Levenshtein* calcula este número, por ejemplo, como es bien sabido, mediante la computación de los registros de una matriz apropiada, en el que la entrada de más a la derecha de la última línea de la matriz resulta ser entonces dicha distancia. Un algoritmo de *Levenshtein* convencional se basa en una búsqueda de caracteres para comparar porciones de texto. Se compara cada carácter individual de una primera porción de texto con cada uno de los caracteres de una segunda porción de texto con el fin de encontrar coincidencias.

A efectos prácticos, cuando las porciones de texto consideradas comprenden por lo general más de sólo unos pocos caracteres, la implementación de este algoritmo es extremadamente lenta. De acuerdo con un aspecto de la presente invención, se modifica, por lo tanto, el concepto del algoritmo en el hecho de que compara líneas y opera tomando como base la palabra. Es decir, se examinan dos líneas de texto comparando las palabras respectivamente incluidas en las líneas.

Para ser más precisos, un algoritmo de *Levenshtein* modificado aplicado a las líneas L_i y Z_j puede calcular el número mínimo de inserciones, borrados y sustituciones de palabras subsumidas de L_i necesarias para formar la línea Z_j. Este número puede ser considerado como la distancia de *Levenshtein* d_{ij} entre las líneas L_i y Z_j con respecto a las palabras incluidas, y puede entonces ser utilizado como un valor de resultado particular calculado por el algoritmo, para determinar si una primera porción de texto está incluida en un segunda porción de texto, en el que la primera porción de texto puede estar modificada o no modificada.

Se pueden concebir otros algoritmos, tales como los que simplemente operan sobre la base de determinar la ratio del número de palabras de la primera línea que se detectan que aparecen en la segunda línea, en comparación con (por ejemplo, dividido por) el número total de palabras de la primera línea, por ejemplo.

De acuerdo con la presente descripción, el al menos un valor de resultado calculado por el algoritmo predeterminado se utiliza para determinar si la primera porción de texto debe ser considerada como incluida o no en una forma modificada o no modificada, en la segunda porción de texto. Por ejemplo, el algoritmo se puede aplicar a una o más tupla(s) de líneas adicionales incluidas respectivamente en la primera y segunda porción de texto, para calcular al menos un valor de resultado suplementario. Puede aplicarse entonces una función a todo(s) el(los) valor(es) de resultado y valor(es) de resultado suplementario(s), y puede realizarse la determinación de si la primera porción de texto se ha de considerar o no como incluida en la segunda porción de texto mediante la comparación del resultado de la función con un valor de umbral.

Como un ejemplo concreto, el algoritmo de *Levenshtein* modificado mencionado se puede aplicar a todas las tuplas (L_p, Z_k) de las líneas L_p de la primera porción de texto y Z_k de la segunda porción de texto, o a una selección apropiada de todas estas tuplas. Se pueden sumar todas las distancias *Levenshtein* resultantes de los pares de líneas considerados, y se puede determinar que la primera porción de texto se ha de considerar como incluida, con modificaciones o sin modificaciones, en la segunda porción de texto, si y sólo si la suma resultante no excede de un valor de umbral predeterminado. Alternativamente, se pueden normalizar o ponderar las respectivas distancias *Levenshtein* antes de ser sumadas, tal como dividiéndolas por el respectivo número de palabras incluidas en la línea respectivamente considerada de la primera o segunda porción de texto, o por el número de palabras incluidas en

cualquiera de las líneas consideradas (es decir, en ambas líneas consideradas conjuntamente). Adicional o alternativamente, dicha suma de distancias *Levenshtein* (cada una posiblemente normalizada o ponderada adicionalmente, según se ha mencionado) se puede dividir por el número total de tuplas de línea consideradas. De esta manera, se puede elegir el umbral para que sea independiente de las porciones de texto particulares.

Por lo tanto, la presente invención se basa en una comparación de palabras, para determinar si dos palabras son iguales o no. A pesar de que esta determinación en sí puede hacerse tomando como base el carácter, la determinación de la similitud de dos porciones de texto se hace entonces considerando las palabras enteras. Por lo tanto, se ignoran cualesquiera similitudes dentro de las diferentes palabras. Esto mejora la velocidad de la implementación en comparación con una técnica convencional que toma como base el carácter. Por otro lado, permite una amplia aplicabilidad, puesto que no se hace restricción alguna a un código específico o lenguaje natural; en contraste, el análisis sintáctico conocido del lenguaje se basa en un analizador específico que dispone de un vocabulario particular y soporta sólo un lenguaje respectivo.

Una realización particular de la presente invención implica además la determinación de si las líneas L_i y Z_j a las que se les aplica un algoritmo predeterminado, se han de considerar como similares. Esta determinación se basa preferentemente en una comparación del al menos un valor de resultado, calculado por el algoritmo cuando se aplica a la tupla L_i y Z_j, con al menos un valor de umbral dado. Por ejemplo, se puede determinar que estas líneas son similares si y sólo si el al menos un valor de resultado es menor que el al menos un valor de umbral, o si y sólo si es más grande que el al menos un valor de umbral. En una realización particular, el procedimiento incluye además la etapa de recibir una selección del uno o más valor(es) umbral(es). Los valores umbral(es) pueden depender de las líneas consideradas, tal como de la línea L_i. Por ejemplo, se puede determinar que una línea L_i incluye un carácter o palabra que indica una llamada a procedimiento. Como consecuencia, puede escogerse el umbral para que refleje una importancia alta de la línea, a fin de aumentar los requisitos que debe cumplir una línea Z_j para que se considere que es similar a la línea L_i.

Preferiblemente, el algoritmo predeterminado es tal que las líneas se determinan como similares si (pero preferiblemente no sólo si) son iguales.

Por ejemplo, el procedimiento puede comprender relacionar la distancia *Levenshtein* modificada d_{ij} de las líneas L_i y Z_j con el número k_i de palabras de la línea L_i, tal como calculando la ratio $r_{ij} := d_{ij}/k_i$.

Si la ratio no supera un valor predeterminado, se podría decidir que las líneas L_i y Z_j se han de considerar como similares, y se podría continuar comparando otro par de líneas de la primera y segunda porción de texto, respectivamente, en el caso de existir, mediante la aplicación del algoritmo a esta otra tupla de líneas.

La consideración de una determinada similitud o disimilitud (lo contrario de similitud) de líneas enteras puede simplificar la determinación de si la primera porción de texto ha de ser considerada o no como incluida, de forma modificada o no modificada, en la segunda porción de texto, puesto que esta última determinación se puede hacer entonces teniendo en cuenta los valores binarios "similares" y/o "no similares". Además, teniendo en cuenta la (di)similitud de líneas, se incorpora la estructura de las porciones de texto en la determinación de si la primera porción de texto ha de ser considerada o no como incluida, en la segunda porción de texto.

El resultado de una determinación de si las líneas L_i y Z_j de la primera y segunda porción de texto, respectivamente, se han de considerar o no como similares, puede ser que las líneas L_i y Z_j no deben ser consideradas como similares. Puede entonces seleccionarse una línea Z_k diferente a la línea Z_j de la segunda porción de texto, incluyendo la línea Z_k al menos una palabra, y entonces el algoritmo predeterminado que compara cada palabra de la línea L_i con cada palabra de la línea Z_k se puede aplicar a la tupla (L_i, Z_k) que se compone de la línea L_i y la línea Z_k, para calcular, en base al resultado de dicha comparación, al menos un valor de resultado adicional. La etapa de determinar si la primera porción de texto debe ser considerada como incluida o no en la segunda porción de texto, puede entonces basarse también en este al menos un valor de resultado adicional. Por ejemplo, la última determinación puede incluir la etapa de determinar, basándose en el al menos un valor de resultado adicional, si las líneas L_i y Z_k han de ser consideradas como similares, por ejemplo mediante la comparación del al menos un valor de resultado adicional con un valor de umbral. Este valor de umbral puede ser el mismo valor que un valor de umbral utilizado para determinar si las líneas L_i y Z_j han de considerarse como similares, o puede ser otro valor de umbral. En un ejemplo particular, una de las o ambas líneas Z_k y Z_j incluyen dos o más palabras.

De este modo, la segunda porción de texto puede ser examinada hasta encontrar una línea que debe ser considerada como similar a la línea L_i. Esto puede permitir una determinación relativa a la inclusión de la primera porción de texto (en forma modificada o no modificada) en la segunda que es más razonable o que se ajusta mejor a una interpretación común de lo que significa "incluido".

Alternativamente, puede realizarse una determinación de si las líneas L_i y Z_j de la primera y segunda porción de texto, respectivamente, se han de considerar como similares o no, en la que la determinación indica que estas líneas deben ser consideradas en efecto como similares. En una realización ejemplar, el procedimiento incluye entonces seleccionar, de la primera porción de texto, una línea L_p diferente de la línea L_i que incluye también al menos una palabra, y éste puede comprender además seleccionar, de la segunda porción de texto, una línea Z_k diferente a la línea Z_j que incluye al menos una palabra. El procedimiento puede incluir además la etapa de aplicar el algoritmo predeterminado a la tupla que se compone de las líneas L_p y Z_k, comparando el algoritmo cada palabra de la línea L_p con cada palabra de la línea Z_k para calcular al menos un valor de resultado adicional. Análogamente a lo anterior, el al menos un valor de resultado adicional se puede utilizar en la etapa de determinar si la primera porción de texto debe ser considerada como incluida o no en la segunda porción de texto. Por ejemplo, la última determinación puede incluir la etapa de determinar, basándose en el al menos un valor de resultado adicional, si las líneas L_p y Z_k han de ser consideradas o no como similares, por ejemplo, mediante la comparación del al menos un valor de resultado adicional con un valor de umbral. Este valor de umbral puede ser el mismo valor que un valor de umbral utilizado para determinar que las líneas L_i y Z_j se han de considerar como similares, o puede ser uno diferente. En una forma de realización particular de la presente invención, al menos una de las líneas L_i y L_p incluyen(n) dos o más palabras.

De acuerdo con este procedimiento, pueden encontrarse varios pares de líneas incluidas respectivamente en la primera y segunda porciones de texto que han de ser consideradas como similares entre sí. La consideración de la similitud a nivel de línea permite la aplicación de una indicación de las líneas que han de ser consideradas como similares, tal como resaltándolas en las respectivas representaciones de los documentos electrónicos que incluyen las porciones de texto. Esto puede ser particularmente útil para un usuario, por ejemplo un usuario en busca de porciones de texto movidas.

Un procedimiento particular comprende la etapa de determinar el número S de aquellas líneas L de la primera porción de texto para las que se puede encontrar una línea Z en la segunda porción de texto, de tal manera que la línea L se ha de considerar como similar a la línea Z. La determinación de si la primera y segunda porciones de texto han de ser consideradas o no como similares puede hacerse entonces en base a la ratio r de ese número dividido por el número total n de líneas de la primera porción de texto, es decir, teniendo en cuenta r := S/n por ejemplo. Alternativamente, dicha ratio puede ser definida por r := S/m o por r := S/(n+m), en que m es el número total de líneas de la segunda porción de texto. Otras definiciones son también posibles. La ratio puede ser comparada con un valor de umbral. En una realización preferida, el procedimiento incluye la etapa de recibir una selección de este valor de umbral. Se puede determinar que la primera porción de texto ha de ser considerada como incluida en la segunda porción de texto si y sólo si se observa que r es mayor que el valor de umbral, por ejemplo.

Por consiguiente, la última etapa de determinación puede centrarse en tener en cuenta el número de similitudes de línea que se han encontrado. Por lo tanto, pueden nivelarse las diferencias entre líneas dentro de la primera porción de texto, las diferencias relativas a las longitudes de línea particulares (o el número de palabras incluidas). De este modo, puede tenerse en cuenta la estructura de las porciones de texto, siguiendo la formación del texto. En particular, la disposición de líneas de código fuente de software refleja por lo general una estructura lógica a considerar, de alguna manera, lo cual puede ser ventajoso cuando se evalúan las porciones de texto, tal como en relación a su similitud. Por ejemplo, una línea puede componerse de una llamada a una función y otra línea de una definición de un bucle. Mientras que la llamada a la función está posiblemente representada por una única palabra, la definición del bucle a menudo incluye varias palabras. Sin embargo, las líneas pueden ser consideradas como que tienen una importancia relacionada. La manera anterior de determinar si la primera región ha de ser considerada como incluida en la segunda región puede evitar la sobreestimación de la importancia de la línea que incluye la definición del bucle.

En una realización ejemplar, la segunda porción de texto puede consistir en una o más líneas incluidas en una tercera porción de texto. En una realización particular, se puede determinar que la primera porción de texto no ha de

ser considerada como incluida (en una forma no modificada o modificada) en la segunda porción de texto. El procedimiento puede entonces comprender además la etapa de seleccionar una nueva segunda porción de texto que comprende asimismo al menos una línea y está incluida en la tercera porción de texto, y determinar si la primera porción de texto está incluida o no en la nueva segunda porción de texto. Esta determinación puede ser hecha mediante la aplicación a la primera y la nueva segunda porción de texto de uno cualquiera de los procedimientos descritos anterior o posteriormente.

De esta manera, encadenando estos pasos, puede determinarse que una sub-porción de la tercera porción de texto ha de ser en efecto considerada como similar a la primera porción de texto. Este procedimiento es, por lo tanto, particularmente útil cuando es incluido en un proceso de búsqueda de código que se ha movido en una operación de edición.

En esa búsqueda, cuando la tercera porción de texto incluye más líneas que la primera porción de texto, puede ser útil aplicar una estrategia particular para encontrar una sub-porción apropiada de la tercera porción de texto. Para este fin, un procedimiento puede incluir la designación de una sub-porción de la tercera porción de texto para que sea la nueva segunda porción de texto, en el que la nueva segunda porción de texto tiene el mismo número de líneas que, o más líneas que la primera porción de texto. Por ejemplo, se pueden seleccionar y considerar iterativamente nuevas segundas porciones de texto hasta que se determine que la primera porción de texto está incluida, en una nueva segunda porción de texto considerada, o hasta que se han verificado todas las sub-porciones posibles (o todas las que tienen un tipo deseado) de la tercera porción de texto. En esto, la selección puede ser tal que, en un primer ciclo, cualquier (nueva) segunda porción de texto considerada tiene el mismo número de líneas que la primera porción de texto. Cuando se determina, para cualquiera de dichas (nuevas) segundas porciones, que la primera porción de texto no ha de considerarse como incluida en la misma, en un segundo (tercer, cuarto y así sucesivamente) ciclo, puede seleccionarse y considerarse una o más nueva(s) segunda(s) porción(es) de texto que tiene(n) uno (dos, tres y así sucesivamente) más líneas que la primera porción de texto.

Dicho procedimiento puede ser denominado como la aplicación de una "ventana deslizante", y puede proporcionar la localización de una segunda porción de texto en la cual está incluida (con modificaciones o sin modificaciones) la primera porción de texto, en el que la segunda porción de texto encontrada es tan pequeña como sea posible, aumentando así la similitud.

En una forma de realización particularmente preferida, la primera y segunda porciones de texto están incluidas en versiones respectivamente modificadas de un documento electrónico original común, denominado en adelante como el documento de base. Por ejemplo, la primera porción de texto puede estar incluida en otra versión resultante de una modificación de un documento electrónico, dicha modificación realizada por un primer usuario. La segunda porción de texto de manera análoga puede estar incluida en una versión resultante de una modificación del mismo documento de base, dicha modificación llevada a cabo por un segundo usuario. En particular, el documento electrónico de base puede incluir código fuente de software.

El procedimiento puede incluir además notificar a un usuario acerca de las similitudes encontradas. Es decir, si se determina que la primera porción de texto ha de ser en efecto considerada como incluida en la segunda porción de texto, entonces puede enviarse o mostrarse un mensaje a un usuario que indique este hecho. Alternativamente o adicionalmente, la primera y/o segunda porción de texto puede ser al menos parcialmente resaltada o marcada de otro modo en una representación de la misma en una pantalla o visualizador. En una forma de realización particular, que incluye la característica de determinar la similitud de líneas según se ha descrito anteriormente, puede aplicarse una identificación a unos pares de esas líneas dentro de la primera y segunda porción de texto que se han determinado como que son similares entre sí. En esto, la aplicación de una identificación puede incluir resaltar las líneas en una representación en una pantalla, o enviar o mostrar un mensaje informando al usuario acerca de la similitud y las líneas seleccionadas.

Un algoritmo conocido de cálculo de diferencias puede ser aplicado a ambas versiones de un documento electrónico, para identificar diferencias. De este modo, la primera y segunda porciones de texto pueden ser determinadas como regiones de diferencias.

El algoritmo de cálculo de diferencias puede considerar una de las versiones como un documento de origen y el otro como un documento de destino. Se puede entonces detectar una porción de texto que aparece en el documento de

origen pero que no aparece en el documento de destino en la posición correspondiente, o viceversa. Tales porciones de texto, en adelante denominadas en este documento como regiones de diferencias, son candidatas a ser inspeccionadas para determinar si se ha llevado a cabo una operación de movimiento por parte de uno de los usuarios, es decir, si el usuario respectivo ha cortado una porción en una cierta posición de un texto y la ha pegado en otra posición del texto. En una forma de realización preferida de la invención, las porciones de texto de los respectivos documentos que han sido detectadas por un algoritmo de cálculo de diferencias son designadas para ser la primera y la segunda porción de texto con el fin de determinar si una de estas porciones de texto ha de ser considerada como incluida en la otra.

Una realización particular de la presente invención es un procedimiento para detectar una operación de movimiento de código en un proceso de edición o de combinación que involucra unas versiones modificadas A y B de un mismo documento de base. De acuerdo con este procedimiento, se aplica un algoritmo de cálculo de diferencias a las versiones A y B, para determinar al menos una región de diferencias en la versión A y al menos una región de diferencias en la versión B. Se selecciona un par de regiones de diferencias de las versiones A y B. Una primera de dichas regiones de diferencias elegidas es considerada como una primera porción de texto y la otra como una segunda porción de texto en la aplicación de uno de los procedimientos anteriores. De este modo, se determina si la primera de las regiones de diferencias elegidas está incluida, con modificaciones o sin modificaciones, en la otra de las regiones de diferencias elegidas.

Alternativamente, en la aplicación de uno de los procedimientos anteriores, una primera de dichas regiones de diferencias elegidas es considerada como una primera porción de texto y la otra como una tercera porción de texto, mientras que una sub-porción de la tercera porción de texto es considerada como la segunda porción de texto. De este modo, puede reducirse el tamaño de la otra de las regiones de diferencias, lo cual puede mejorar, por lo tanto, la búsqueda de una porción de texto que se ha movido durante la edición de la respectiva versión.

En una realización, el procedimiento puede incluir la determinación de los números de línea de cada una de las regiones de diferencias elegidas, y la región de diferencias que tiene menos líneas puede ser considerada como la primera porción de texto. Esto puede mejorar el procedimiento. En efecto, si la primera porción de texto tiene más líneas que la segunda porción de texto, el número de líneas de la primera porción de texto que no tienen contrapartida similar en la segunda porción de texto será al menos la diferencia del número de líneas incluidas en la primera porción de texto en comparación con el número de líneas incluidas en la segunda porción de texto. Por lo tanto, dependiendo del umbral considerado, incluso antes de comenzar una comparación de contenidos de las líneas, puede considerarse que la probabilidad de similitud entre las regiones es pequeña.

Alternativamente, el procedimiento puede incluir el paso de determinar, para una primera región del par de regiones de diferencias elegidas designada como la primera porción de texto y la otra región de dicho par como la segunda porción de texto (respectivamente, la tercera porción de texto), que la primera porción de texto no ha de ser considerada como incluida, con modificaciones o sin modificaciones, en la segunda porción de texto. El procedimiento puede incluir entonces la etapa adicional de considerar el mismo par de porciones de diferencias elegidas al revés, es decir, aplicando uno de los procedimientos anteriores a la primera región de las regiones de diferencias elegidas como la segunda (respectivamente, tercera) y la otra región del par de regiones de diferencias como la primera porción de texto.

Esta forma de realización puede mejorar la probabilidad de encontrar porciones de texto que pueden haberse movido en una versión en comparación con otra versión de un documento electrónico.

El procedimiento puede incluir la etapa de determinar que el número de líneas de cada región de diferencias es mayor que o al menos igual a un número de líneas mínimo dado. De esta manera, se pueden excluir regiones de diferencias demasiado pequeñas de la determinación de acuerdo con uno de los procedimientos anteriores, lo que puede evitar que se identifiquen erróneamente muchas asignaciones individuales.

Adicionalmente o alternativamente, el procedimiento puede incluir además la etapa de elegir otro par de regiones de diferencias después de considerar un par de regiones de diferencias, y aplicar consecuentemente uno de los pasos de procedimiento anteriores al otro par. De este modo, pueden examinarse dos versiones de un documento para detectar porciones de texto posiblemente movidas.

Una vez que se han encontrado tales porciones de texto, se pueden resaltar o indicar de otro modo a un usuario según se ha detallado anteriormente. En un proceso de combinación, se pueden extraer las etapas adecuadas. Por ejemplo, en una combinación manual, un usuario puede comprobar de nuevo sólo las regiones indicadas, para averiguar los detalles de las correspondientes modificaciones de versión, evitando así la tarea frustrante y propensa al error de inspeccionar los documentos completos.

Se describen realizaciones particulares en los dibujos adjuntos.
Figura 1: muestra dos porciones de texto que incluyen varias líneas y palabras
Figura 2: es un diagrama de bloques que muestra un proceso ejemplar de determinar si dos líneas han de ser consideradas como similares
Figura 3: muestra la estructura de un proceso para determinar si una primera porción de texto ha de ser considerada como incluida en una segunda porción de texto
Figura 4: muestra la funcionalidad de aplicar ventanas de búsqueda (ventanas deslizantes)
Figura 5: muestra un proceso de elegir segundas porciones de texto dentro de una tercera porción de texto (ventana deslizante)

En la figura 1, se muestran dos documentos electrónicos 6, 7 que incluyen cada uno de ellos un código fuente de software. Las porciones de texto 1 y 2 han sido elegidas, según está marcado en el dibujo por medio de los recuadros respectivos. En la situación de ejemplo mostrada, el bloque 1 se puede entender como la primera porción de texto según la nomenclatura de este documento, mientras que el fragmento 2 puede ser considerado como la segunda porción de texto. La primera porción de texto 1 se compone entonces de cuatro líneas 4 y la segunda porción de texto de cinco líneas 5. Ambos documentos, y, en particular, cada una de las porciones de texto incluye varias palabras 3, en el que en este caso, se supone que el carácter de espacio está incluido en el conjunto de caracteres especiales que separan (y con ello, definen) las palabras según se ha descrito anteriormente. La definición de una palabra depende de dicho conjunto predeterminado de caracteres especiales. En la primera porción de texto de ejemplo mostrada, podría ser razonable designar también los paréntesis "(" y ")" como elementos, además del carácter de espacio, de ese conjunto de caracteres especiales. Siendo así, (y suponiendo que no hay más caracteres incluidos que hayan sido definidos para su inclusión en ese conjunto de caracteres especiales) la expresión "GetPath(string)" se considera como las dos palabras "GetPath" y "string". Si ese conjunto incluyera el carácter de espacio como único elemento, la expresión "GetPath(string)" tendría que ser interpretada como una sola palabra.

Como puede verse en la Figura 1, las porciones de texto 1 y 2 se diferencian entre sí en que la segunda porción de texto 2 incluye una línea que comprende "println;" mientras que la primera porción de texto 1 no la comprende. Además, la llamada al procedimiento "GetPath" está aplicada al valor "false" en la primera porción de texto y "true" en la segunda.

Depende de la configuración exacta del algoritmo predeterminado si, en este caso, se determinaría que la primera porción de texto ha de ser en efecto considerada como incluida en la segunda porción de texto, en una forma modificada. Este puede ser el caso, por ejemplo, si se realiza una determinación de similitud de líneas según se ha detallado anteriormente, y si se hubiera establecido un umbral de tal manera que la primera porción de texto ha de ser considerada como incluida en la segunda porción de texto si y sólo si la ratio del número de líneas (en la primera porción de texto) que tiene una contraparte similar en la segunda porción de texto, dividido por el número de líneas totales de la primera porción de texto, es al menos 3/4. En efecto, independientemente de si las líneas que incluyen la llamada "return GetPath(...)"se consideran como similares o no, tres de las cuatro líneas de la primera porción de texto son iguales a unas líneas de la segunda porción de texto, y razonablemente las líneas iguales son consideradas como similares.

Como puede verse además en la figura 1, la primera y la segunda porciones de texto aparecen en diferentes posiciones en los respectivos documentos 6 y 7. Las porciones de texto pueden, por lo tanto, considerarse como que indican un movimiento de texto llevado a cabo en una modificación del primer documento electrónico 6, resultando en la modificación del segundo documento electrónico, estando el movimiento del texto acompañado de una ligera modificación del texto.

La figura 2 representa un proceso de determinar si dos líneas L_i y Z_j se han de considerar como similares. Se supone que las líneas están ya seleccionadas cuando se inicia el proceso. En el paso 21, se determina el número de

palabras k_i incluidas en la línea L_i. En el paso 22 (que está representado a continuación de la etapa 21, pero que igualmente puede ser realizado antes), se determina una distancia *Levenshtein* d_{ij} modificada de las líneas L_i y Z_j con respecto a palabras. La ratio d_{ij}/k_i es comparada, en el paso 23 con un valor de umbral t predeterminado. Si la ratio no es mayor que el valor de umbral, entonces se determina que las líneas L_i y Z_j han de ser consideradas como similares (paso 24), y en caso contrario que no lo son (paso 25).

La figura 3 muestra un posible procedimiento para determinar si una primera porción de texto ha de ser considerada como incluida en una segunda porción de texto, en la que puede estar incluida en una forma modificada o no modificada. En esto, se supone que la primera porción de texto se compone de las líneas L_1, L_2,..., L_n, (n siendo el número de líneas de la primera porción de texto). La segunda porción de texto se supone que tiene m líneas Z_1, Z_2,..., Z_m.

El proceso se inicia con el establecimiento, en el paso 30, de los números enteros i, j, k y S a cero. Durante el proceso, el número S refleja el número de líneas de la primera porción de texto que se determina que tienen una contraparte similar en la segunda porción de texto. El procedimiento entra entonces en un bucle en la etapa 31, en el que se determina si L_i y Z_j se han de considerar como similares. Por ejemplo, el procedimiento representado en la Figura 2 se puede aplicar en el paso 31. Si se determina que L_i y Z_j han de ser consideradas como similares, entonces el proceso pasa a la etapa 32, incrementando en uno el contador de líneas similares S. De lo contrario, el procedimiento continúa determinando, en el paso 36, si la línea Z_j considerada ha sido la última línea de la segunda porción de texto, es decir, si $j + 1 > m$. Si no, entonces se incrementa j en uno, en el paso 41, y el proceso continúa a la etapa 31 para determinar si la línea L_i ha de ser considerada como similar a la siguiente línea de la segunda porción de texto (es decir, la siguiente línea con respecto a la previamente considerada).

Si, sin embargo, en el paso 36 se determina que Z_j ha sido la última línea, esto significa que la línea L_i ya ha sido comparada, con respecto a la similitud, a todas las líneas Z_k, Z_{k+1},..., Zm de la segunda porción de texto. En el paso 37 se determina si L_i es la última línea de la primera porción de texto, es decir, si $i + 1 > n$. Si no, entonces i se incrementa en uno en la etapa 42, y se restablece j con el valor k. El proceso vuelve al paso 31. Cabe señalar que la consideración del valor k permite la comparación de la nueva línea L_i con sólo las líneas de la segunda porción de texto que suceden a la última coincidencia de línea, es decir, la última línea Z_j que se ha determinado que es similar a una línea L_i considerada previamente. De este modo, se puede tener en cuenta el orden de las líneas de la primera y la segunda porción de texto, y se pueden ignorar las similitudes múltiples de líneas.

Si, por el contrario, en el paso 37 se determina que L_i ha sido la última línea de la primera porción de texto, entonces el número S de coincidencias de similitud de líneas encontradas se divide por el número n de líneas de la primera porción de texto, y en el paso 38 se compara la ratio resultante con un valor de umbral T. Si se observa que $S/n \leq T$, entonces se determina, en el paso 39, que la primera porción de texto ha de ser considerada como incluida, posiblemente en una forma modificada, en la segunda porción de texto, de lo contrario, en el paso 40, que no lo es.

Son concebibles otras condiciones, tales como la sustitución de S/n por S/m, o por S/(n+m) y pasar a la etapa 40 cuando la última ratio es mayor que T, y continuar al paso 39 en caso contrario. En esta configuración, se tendrá en cuenta el número de líneas de la segunda porción de texto.

Volviendo al caso en que, en el paso 31, se ha determinado que L_i y Z_j se han de considerar como similares, después de incrementar el contador S en el paso 32, se determina, en el paso 33, si la L_i considerada más recientemente es la última línea de la primera porción de texto, es decir, si $i + 1 > n$. Si es así, el proceso pasa a la etapa de evaluación 38. De lo contrario, en el paso 34, se determina si la línea Z_j considerada más recientemente es la última línea de la segunda porción de texto. Si es así, no hay más comparaciones a realizar, y el proceso continúa al paso de evaluación 38. De lo contrario, en el paso 35, se incrementan i, j y k en uno, y, en el paso 31, se compara el siguiente par de líneas de la primera y segunda porción de texto, respectivamente.

Como se señaló anteriormente, de acuerdo con el procedimiento representado en la Figura 3, una nueva línea L_i sólo se compara con las líneas de la segunda porción de texto que (en el orden dado por los índices) suceden a la última línea encontrada para la cual se determinó que es similar a una línea considerada previamente de la primera porción de texto. Esto permite una reducción de los cálculos a realizar. Por otro lado, si dos líneas de la primera porción de texto aparecen ambas en la segunda porción de texto, pero en orden inverso, el proceso anterior identificaría sólo una de las similitudes de línea. Puede diseñarse, por lo tanto, un proceso alternativo para comparar

cada línea L_i de la primera porción de texto con cada línea de la segunda porción de texto, o con cada una de aquellas líneas de la segunda porción de texto que hasta ahora no se han determinado que sean similares a cualquier línea considerada previamente de la primera porción de texto.

La Figura 4 esboza, de una forma simplificada, una primera porción de texto 4 y una tercera porción de texto 8. La primera porción de texto incluye cuatro líneas 4. Según un patrón relacionado, las respectivas líneas similares son indicadas en la tercera porción de texto. Los bloques verticales indican selecciones de líneas de la tercera porción de texto 8 como segundas porciones de texto. Es decir, se entiende que las respectivas segundas porciones de texto comprenden respectivamente aquellas líneas de la tercera porción de texto cuyas extensiones indicadas en el lado derecho del dibujo contactan con los bloques verticales. Se puede entonces determinar primero si la primera porción de texto está incluida o no, con modificaciones o sin modificaciones, en la segunda porción de texto 2a. Como puede verse, la ratio de líneas similares con respecto al número de líneas de la primera porción de texto sería 2/4, que (suponiendo que el umbral ha sido preseleccionado siendo 1 o 4/5, por ejemplo) es demasiado pequeño para la aprobación de una inclusión (o similitud). Puede seleccionarse entonces una nueva segunda porción de texto 2b. En este caso, la ratio anterior resultaría ser 3/4, por lo tanto todavía no es lo suficientemente grande para considerar que la primera porción de texto está incluida en la segunda porción de texto 2b. Se podría continuar el deslizamiento hacia abajo por la ventana de búsqueda, es decir, los bloques de cuatro líneas considerados como segunda porción de texto, pero dicha ratio nunca alcanzaría el umbral 1. Se puede considerar entonces una nueva segunda porción de texto 2a' que proporciona cinco líneas, calcular la ratio correspondiente y deslizar hacia abajo la ventana cuando se define una nueva segunda porción de texto (tal como 2b') si la ratio es demasiado pequeña. En la situación ejemplar representada, la nueva segunda porción de texto representada por el bloque 2b" produciría finalmente la ratio 1.

La Figura 5 ilustra un proceso de aplicación del proceso de deslizamiento. En éste, se supone que la tercera porción de texto se compone de q líneas T_1, T_2,..., T_q. El número de líneas de la primera porción de texto TP1 se considera que es n, en que se supone además que n es menor que q. En la etapa 50, los números enteros j y k son inicializados a 1 y n, respectivamente. Entonces se define, en la etapa 51, una segunda porción de texto TP2 para que se componga de una sub-porción de la tercera porción de texto, la sub-porción determinada por las k líneas Z_j, Z_{j+1},..., Z_{j+k-1}. A continuación, en la etapa 52, se realiza una determinación de si la primera porción de texto ha de ser considerada como incluida (posiblemente de forma modificada) en la segunda porción de texto. Si es así, el proceso termina, de lo contrario se realiza una determinación, en el paso 53, de si la última línea de la segunda porción de texto recientemente considerada es la última línea de la tercera porción de texto, es decir, si $j + k > q$. Si es así, se comprueba, en el paso 55, si puede elegirse un nuevo bloque que tenga más líneas como nueva segunda porción de texto dentro de la tercera porción de texto. Para este fin, se determina si $k + 1 > q$. Si es así, la segunda porción de texto considerada en último lugar (que tiene k líneas) ya ha sido toda de la tercera porción de texto, y termina el proceso. De lo contrario, k se incrementa en uno, en la etapa 56, y se restablece j con el valor 1. El proceso continúa en la etapa 51 que define una nueva segunda porción de texto TP2.

REIVINDICACIONES

1. Procedimiento ejecutado informáticamente para determinar automáticamente si una primera porción de texto (1) debe ser considerada como incluida o no en una segunda porción de texto (2), en la que puede estar incluida en una forma no modificada o modificada, estando ambas porciones de texto codificadas electrónicamente y estructuradas en un número respectivo de una o más líneas (4, 5),

el procedimiento **caracterizado por** las etapas de

seleccionar una línea L_i de la primera porción de texto y una línea Z_j de la segunda porción de texto, incluyendo la línea L_i al menos una, dos, tres o más palabra/s (3) e incluyendo la línea Z_j al menos una, dos, tres o más palabra/s;

aplicar un algoritmo *Levenshtein* modificado a la tupla que se compone de las líneas L_i y Z_j, comparando el algoritmo *Levenshtein* cada palabra de la línea L_i con una o algunas o cada palabra/s de la línea Z_j para determinar si las palabras comparadas son iguales o no, y calcular, utilizando el si las palabras comparadas son iguales o no, al menos un valor de resultado;

determinar, utilizando el al menos un valor de resultado, si la primera porción de texto ha de ser considerada o no como incluida en la segunda porción de texto; y

si la línea L_i se considera como incluida en la línea Z_j comparar la línea que sucede a L_i de la primera porción de texto sólo con las líneas que suceden a la línea Z_j de la segunda porción de texto.

2. Procedimiento de la reivindicación 1, en el que la determinación de si la primera porción de texto ha de ser considerada o no como incluida en la segunda porción de texto incluye la etapa de

determinar (31) si la línea L_i ha de ser considerada como similar a la línea Z_j, que

incluye comparar (23) el al menos un valor de resultado con un valor de umbral y determinar si las líneas L_i y Z_j han de ser consideradas o no como similares en base al resultado de la comparación,

en el que el procedimiento preferiblemente incluye además la etapa de recibir una selección del valor de umbral.

3. Procedimiento de la reivindicación 2, en el que la segunda porción de texto incluye varias líneas, y el procedimiento incluye además las etapas de

seleccionar, en el caso de que la línea L_i no se ha de considerar como similar a la línea Z_j, una línea Z_k diferente de Z_j e incluida en la segunda porción de texto, comprendiendo la línea Z_k al menos una, dos, tres o más palabra/s,

y aplicar el algoritmo predeterminado a la tupla que se compone de la línea L_i y la línea Z_k, comparando el algoritmo cada palabra de la línea L_i con una o algunas o cada palabra/s de la línea Z_k para determinar si las palabras comparadas son iguales o no, y calcular, utilizando el si las palabras comparadas son iguales o no, al menos un valor de resultado adicional, y

en el que la determinación de si la primera porción de texto ha de ser considerada o no como incluida en la segunda porción de texto se basa además en el al menos un valor de resultado adicional.

4. Procedimiento de las reivindicaciones 2 ó 3, en el que la primera porción de texto y la segunda porción de texto ambas incluyen varias líneas, y el procedimiento incluye además las etapas de

seleccionar, en el caso de que la línea L_i se ha de considerar como similar a la línea Z_j, una línea L_p incluida en la primera porción de texto y diferente de L_i, y una línea Z_k incluida en la segunda porción de texto y diferente de Z_j, comprendiendo cada una de las líneas L_p y Z_k al menos una, dos, tres o más palabra/s, y

aplicar el algoritmo predeterminado a la tupla que se compone de la línea L_p y la línea Z_k, comparando el algoritmo cada palabra de la línea L_p con una o algunas o cada palabra/s de la línea Z_k para determinar si las palabras

comparadas son iguales o no, y calcular por parte del algoritmo, utilizando el si las palabras comparadas son iguales o no, al menos un valor de resultado adicional, y

en el que la determinación de si la primera porción de texto ha de ser considerada o no como incluida en la segunda porción de texto se basa además en el al menos un valor de resultado adicional.

5. Procedimiento de una de las reivindicaciones 2 a 4,

en el que el procedimiento comprende además

determinar el número S de líneas de la primera porción de texto que han de ser consideradas como similares a una línea respectiva de la segunda porción de texto, y

calcular la ratio r del número S dividido por uno de los siguientes números

el número total de líneas de la primera porción de texto, o

el número total de líneas de la segunda porción de texto, o

la suma del número total de líneas de la primera porción de texto y el número total de líneas de la segunda porción de texto; y

en el que la determinación de si la primera porción de texto ha de ser considerada o no como incluida en la segunda porción de texto incluye la comparación (38) de la ratio r con un valor de umbral, y

en el que el procedimiento preferiblemente comprende además el paso de recibir una selección del valor de umbral.

6. El procedimiento de una de las reivindicaciones 1 a 5, en el que la segunda porción de texto se compone de un conjunto de una, dos, tres o más líneas incluidas en una tercera porción de texto (8), comprendiendo el procedimiento además las etapas de

seleccionar, si la primera porción de texto se ha determinado como que no ha de ser considerada como incluida en la segunda porción de texto (2a; 2b; 2c), una nueva segunda porción de texto (2a'; 2b'; 2a"; 2b") que es otro conjunto de una o más líneas incluidas en la tercera porción de texto, y

determinar, mediante la aplicación del procedimiento de una de las reivindicaciones 1 a 5 a la primera y la nueva segunda porción de texto, si la primera porción de texto ha de ser considerada o no como incluida en la nueva segunda porción de texto.

7. El procedimiento de la reivindicación 6, en el que el número de líneas de la segunda porción de texto es igual al número de líneas de la primera porción de texto, y la nueva segunda porción de texto

tiene el mismo número de líneas que la segunda porción de texto

o tiene al menos una línea más que la segunda porción de texto.

8. El procedimiento de una de las reivindicaciones 1 a 7, que comprende además la aplicación de una identificación, como por ejemplo resaltándolas en una representación en una pantalla, a la primera porción de texto y segunda porción de texto, si se determina que la primera porción de texto ha de considerarse incluida en el segunda porción de texto,

o, cuando el procedimiento incluye la característica de la reivindicación 2, si una o más líneas de la primera porción de texto se determinan como que son similares a una o más líneas respectivas de la segunda porción de texto, la aplicación de una identificación a la una o más líneas de cada una de la primera y segunda porciones de texto, como por ejemplo, resaltándolas en una representación en una pantalla.

9. El procedimiento de una de las reivindicaciones 1 a 8, en el que la primera porción de texto es una porción de un primer documento electrónico (6) y la segunda porción de texto es una porción de un segundo documento electrónico (7),

en el que el primer y el segundo documentos electrónicos ambos son preferiblemente una respectiva versión modificada del mismo documento electrónico original, tal como un documento de código fuente de software.

10. El procedimiento de la reivindicación 9, en el que

la primera y segunda porciones de texto se determinan, mediante un algoritmo de cálculo de diferencias aplicado al primer y segundo documento electrónico, como regiones de diferencias de los documentos electrónicos,

o, cuando se incluye la característica de la reivindicación 6, en el que la primera y tercera porciones de texto se determinan, mediante un algoritmo de cálculo de diferencias aplicado al primer y segundo documento electrónico, como regiones de diferencias de los documentos electrónicos.

11. Procedimiento para la detección de una operación de movimiento de código en un proceso de edición o combinación que involucra a unas versiones modificadas A y B de un mismo documento de base, incluyendo el procedimiento las etapas de

aplicar un algoritmo de cálculo de diferencias a las versiones A y B, para determinar regiones de diferencias en estas versiones,

elegir una región de diferencias de cada una de las versiones A y B,

utilizar una región de diferencias elegida de una versión como una primera porción de texto y una región de diferencias elegida de la otra versión como la segunda porción de texto en la aplicación de uno de los procedimientos de las reivindicaciones 1 a 10;

o usar la región de diferencias elegida de una versión como una primera porción de texto, la región de diferencias elegida de la otra versión como la tercera porción de texto y una porción seleccionada de la tercera porción como la segunda porción de texto en la aplicación de uno de los procedimientos de las reivindicaciones 6 a 10 que incluye la característica de la reivindicación 6.

12. El procedimiento de la reivindicación 11, que comprende además

determinar el número de líneas de cada una de las regiones de diferencias elegidas,

determinar que una primera de las regiones de diferencias elegidas tiene un número de líneas más pequeño o igual que el número de líneas de la otra región de diferencias elegida, y

tener en cuenta la primera región de diferencias elegida como la primera porción de texto.

13. El procedimiento de una de las reivindicaciones 11 ó 12, que comprende además

determinar que una región de diferencias incluye al menos un número de líneas mínimo predeterminado,

incluyendo el procedimiento preferiblemente además recibir la selección del número de líneas mínimo.

14. Medio legible informáticamente que tiene almacenadas en el mismo unas instrucciones que realizan, cuando se ejecutan en un ordenador, uno o más de los procedimientos de las reivindicaciones 1 a 13.

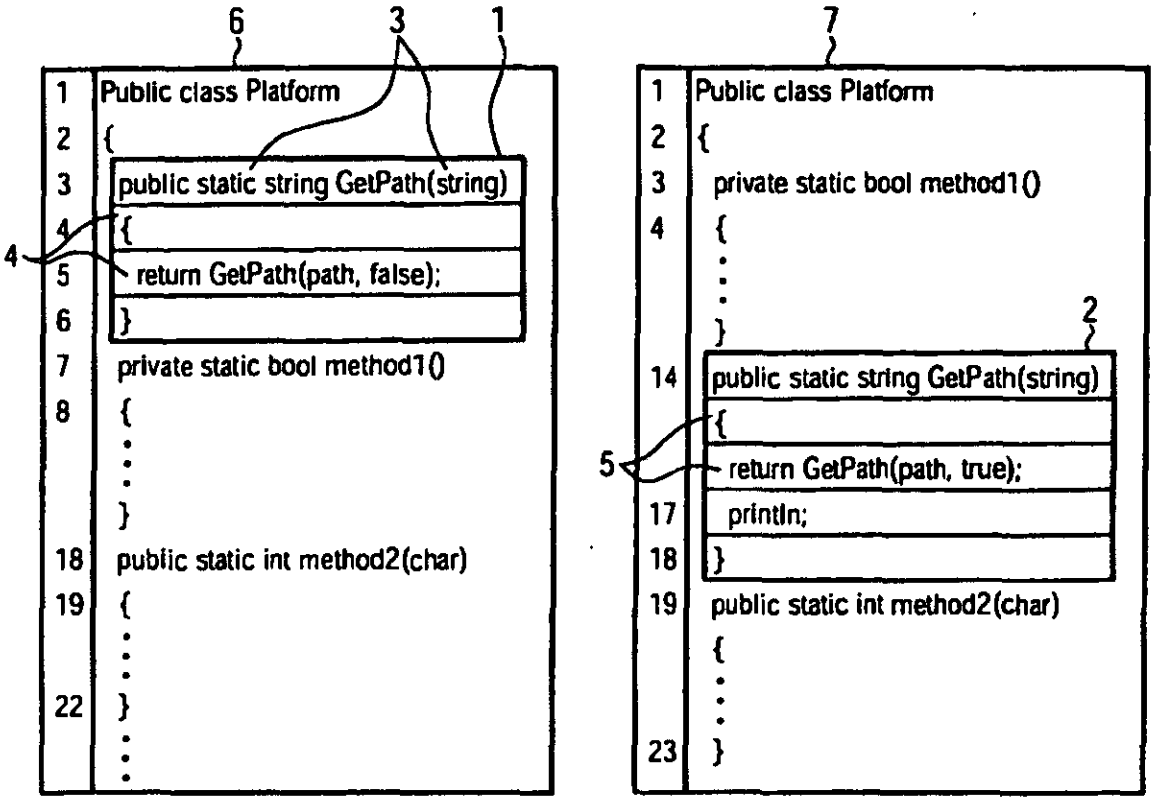
6
3
1
7
1 Public class Platform
2 {
3 public static string GetPath(string)
4 {
5 return GetPath(path, false);
6 }
7 private static bool method1()
8 {
}
18 public static int method2(char)
19 {
22 }
4
1 Public class Platform
2 {
3 private static bool method1()
4 {
}
2
14 public static string GetPath(string)
{
return GetPath(path, true);
17 println;
18 }
19 public static int method2(char)
{
23 }
5

FIG. 1

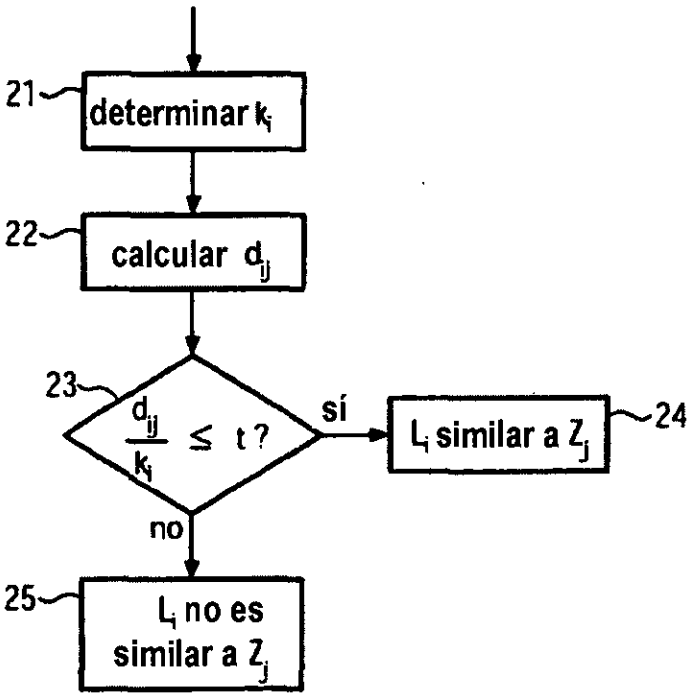
21
determinar k_i
22
calcular d_{ij}
23
$\frac{d_{ij}}{k_i} \leq t$?
sí
L_i similar a Z_j
24
no
25
L_i no es similar a Z_j

FIG. 2

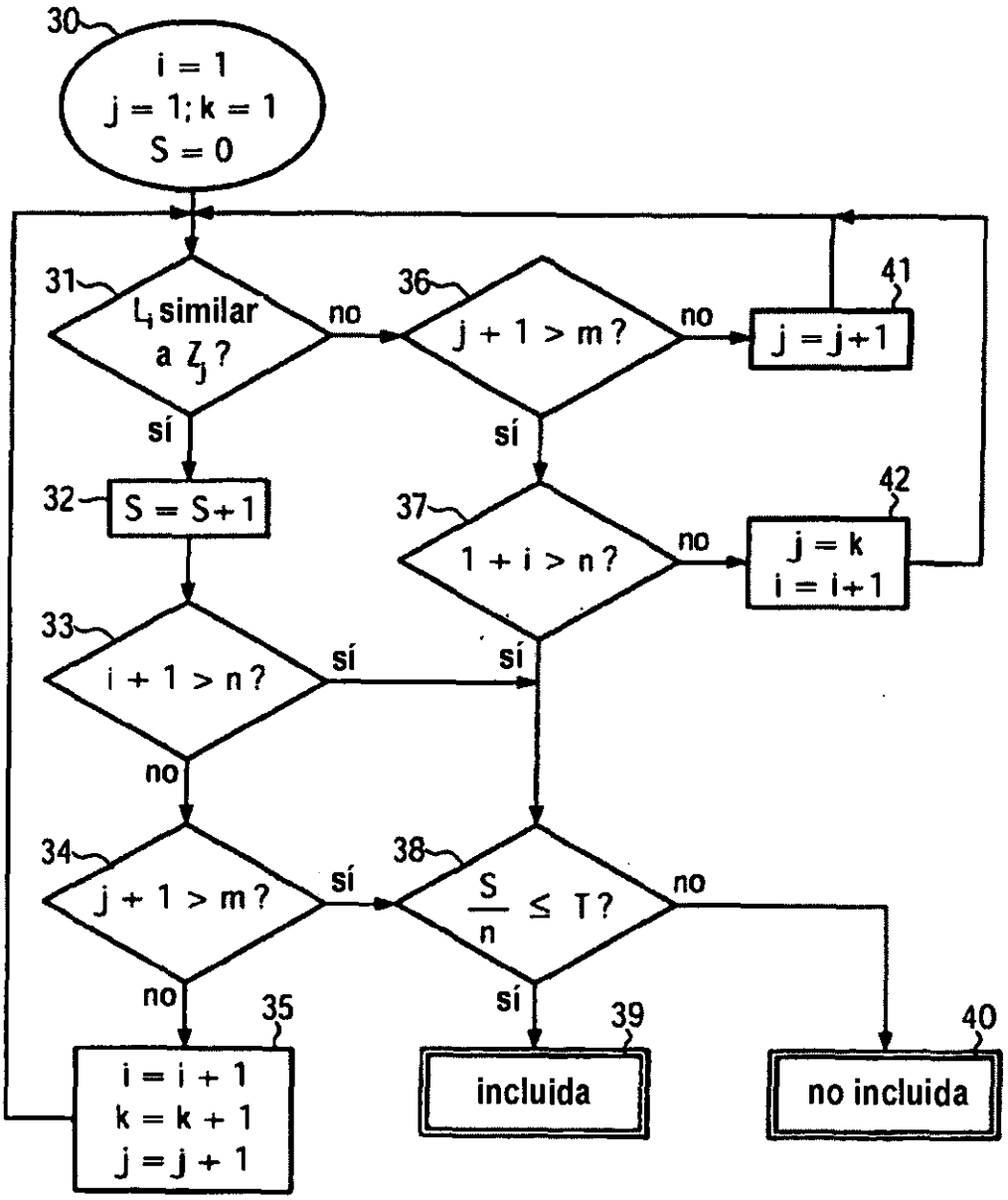
30
i = 1
j = 1; k = 1
S = 0
31
L_i similar a Z_j?
no
sí
36
j + 1 > m ?
no
sí
41
j = j+1
32
S = S+1
37
1 + i > n ?
no
sí
42
j = k
i = i+1
33
i + 1 > n ?
sí
sí
no
34
j + 1 > m ?
sí
38
$\frac{S}{n} \leq T$?
no
sí
no
35
i = i + 1
k = k + 1
j = j + 1
39
incluida
40
no incluida

FIG. 3

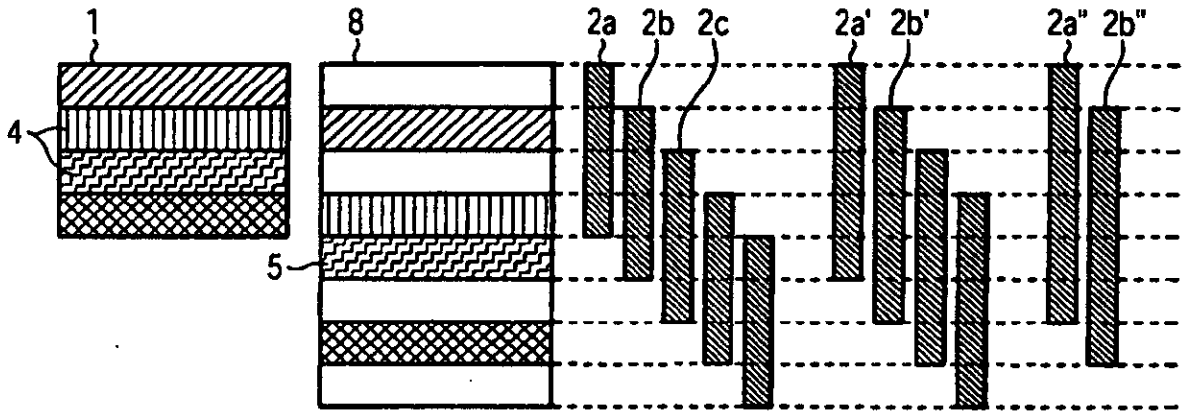

FIG. 4

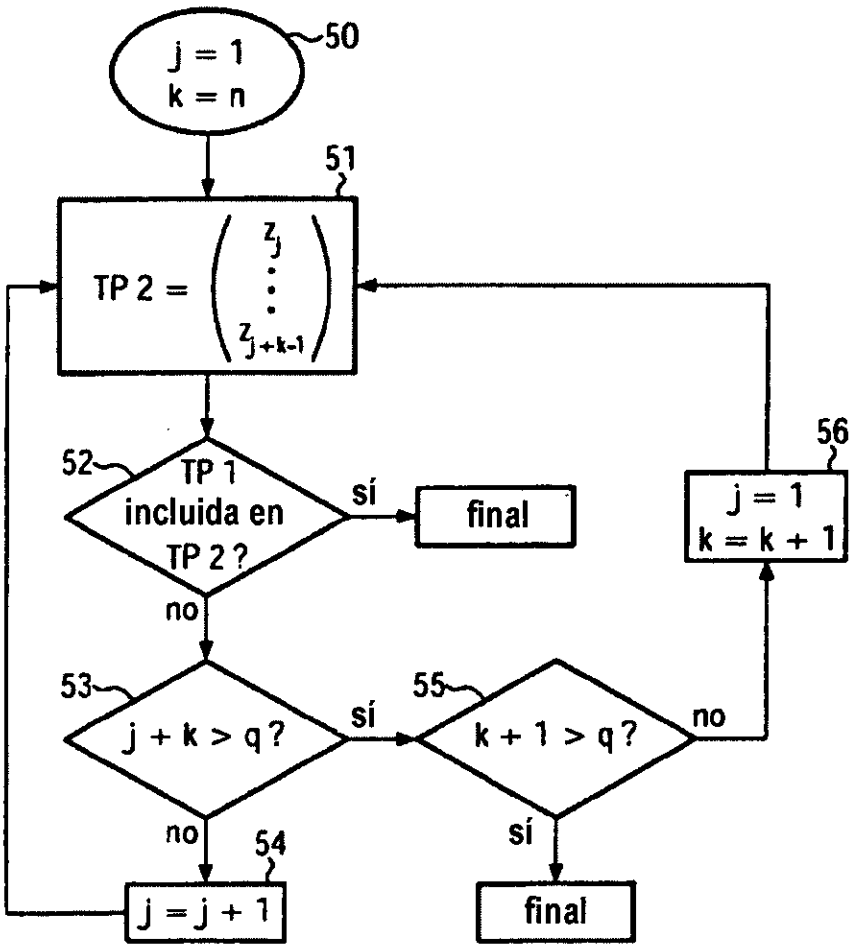
j = 1
k = n
50
51
TP 2 =
z_j
z_{j+k-1}
52
TP 1
incluida en
TP 2 ?
sí
final
no
53
j + k > q ?
sí
55
k + 1 > q ?
no
56
j = 1
k = k + 1
no
54
j = j + 1
sí
final

FIG. 5