



MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA LOTTA ALLA CONTRAFFAZIONE
UFFICIO ITALIANO BREVETTI E MARCHI

DOMANDA DI INVENZIONE NUMERO	102010901879846
Data Deposito	12/10/2010
Data Pubblicazione	12/04/2012

Classifiche IPC

Titolo

METODO PER ESEGUIRE IL CONFRONTO AUTOMATICO DI DOCUMENTI TESTUALI.

TITOLO

METODO PER ESEGUIRE IL CONFRONTO AUTOMATICO DI DOCUMENTI TESTUALI

SETTORE TECNICO

La presente invenzione concerne un metodo per il confronto di documenti di testo e, in modo specifico ma non esclusivo, per il confronto di codici sorgente di programmi per elaboratori.

STATO DELL'ARTE

Il confronto di documenti testuali editabili tramite editors di testi in elaboratori elettronici ha rilevante importanza in vari settori e per svariati motivi. Tra i principali tipi di documenti testuali per i quali risulta particolarmente importante poter avere a disposizione strumenti per eseguire un confronto efficace ve ne sono due tipologie: i documenti di testo formattati per essere letti da un essere umano, quali ad esempio quelli realizzati tramite l'applicativo Microsoft Word o mediante applicazioni sostanzialmente equivalenti; ed i documenti testuali che costituiscono codici sorgente di programmi per elaboratore, vale a dire serie di istruzioni scritte in specifici linguaggi che letti da un elaboratore elettronico gli consentono di eseguire specifiche operazioni. Per il primo tipo di documenti testuali un applicativo efficace per il confronto di due documenti è quello che permette di rendere immediatamente visibile all'utente, in maniera chiara ed immediata, le differenze tra due documenti, solitamente tra un documento ed una sua revisione. Per i programmi per elaboratore, un confronto efficace, ad esempio tra documenti che costituiscono due successive fasi di sviluppo di un programma, è sotto specifici aspetti ancor più importante in quanto può permettere anche di ottimizzare la produzione del software e di valutare l'operato e la produttività del programmatore ed il costo dei progetti software.

Esistono numerosi applicativi che consentono di eseguire il confronto di documenti testuali. Ad esempio esistono applicazioni per il confronto di documenti testuali scritti in specifici linguaggi di programmazione quale il Cobol. Inoltre, in ambiente Windows esiste una utility che permette il confronto di documenti all'interno degli ambienti di sviluppo Visual Studio. Anche all'interno del programma Word di Microsoft, così come in analoghi programmi di scrittura, esiste una specifica funzione che permette di confrontare due documenti o, preferibilmente due diverse versioni di uno stesso documento.

La maggior parte degli applicativi per il confronto di documenti testuali esegue il confronto per righe di testo e, laddove le righe non siano esattamente individuabili in virtù della formattazione del testo, creano delle righe fittizie prendendo in lettura un certo numero di caratteri dal documento per poter eseguire il confronto. In alcuni casi, come nel caso di Word, il confronto avviene per singole parole.

Una delle principali lacune degli applicativi per il confronto di documenti testuali ad oggi noti è la limitazione ad operare nell'ambito di uno specifico sistema operativo e, in molti casi addirittura all'interno di uno specifico applicativo (il caso di Word). Inoltre, anche l'efficacia e la correttezza del confronto possono essere migliorate. Infatti, negli applicativi per il confronto dei codici sorgente di programmi per elaboratore esiste, ad esempio, il problema dell'esistenza di righe uguali: gli algoritmi di confronto di questo tipo di documenti, una volta trovato che una certa riga compare in uno dei due documenti e non nell'altro, scorrono l'altro file fino al raggiungimento di quella riga contabilizzando le opportune righe come aggiunte o eliminate. Chiaramente se tale riga è una riga gemella, la sua identificazione come la riga stessa che veniva cercata

provoca errori di attribuzione, contabilizzando come aggiunte o eliminate righe che in realtà sono uguali nei due file.

SINTESI DELL'INVENZIONE

Scopo della presente invenzione è allora quello di proporre un metodo per il confronto di documenti testuali dalla efficienza migliorata.

In particolare uno scopo della presente invenzione è proporre un metodo per il confronto di documenti testuali che sia indipendente dal sistema operativo installato.

Un altro scopo della presente invenzione è proporre un metodo per il confronto di codici sorgente di programmi di elaboratore che consenta di valutare in termini quantitativi e qualitativi, le modifiche apportate ad un certo codice sorgente rispetto ad una sua versione precedente.

Secondo un aspetto della presente invenzione, gli scopi suddetti sono raggiunti per mezzo di un metodo per il confronto di documenti testuali in cui il metodo comprende:

la generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

l'ordinamento alfa-numerico degli elementi all'interno di detto primo vettore e di detto secondo vettore;

l'eliminazione delle ripetizioni di elementi che si presentano più volte nel vettore con conteggio delle relative occorrenze,

la comparazione dell'ottenuto primo vettore con l'ottenuto secondo vettore per verificare le corrispondenze con elementi del secondo vettore.

Vantaggiosamente il suddetto metodo comprende ulteriori preventive fasi di:

generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

la prima comparazione del primo vettore con il secondo vettore per verificare le corrispondenze di elementi del primo vettore con elementi del secondo vettore,

la verifica se nel primo vettore e nel secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un nuovo primo vettore e di un nuovo secondo vettore,

la ripetizione della suddetta prima comparazione eseguita con detto nuovo primo vettore e detto nuovo secondo vettore,

la verifica se nel nuovo primo vettore e nel nuovo secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un ulteriore nuovo primo vettore e di un ulteriore nuovo secondo vettore,

una seconda comparazione dell'ulteriore nuovo primo vettore con l'ulteriore nuovo secondo vettore,

la scrittura dei risultati del suddetto confronto;
in cui la generazione del suddetto ulteriore nuovo primo vettore avviene eliminando le ripetizioni degli elementi ripetuti di detto nuovo primo vettore e annotandone le occorrenze, e la generazione del suddetto ulteriore nuovo secondo vettore avviene eliminando le ripetizioni degli elementi ripetuti di detto nuovo secondo vettore e annotandone le occorrenze.

Ancora vantaggiosamente nel metodo della presente invenzione la generazione di detto nuovo primo vettore e di detto nuovo secondo vettore avvengono scomponendo detto primo e secondo documento

testuale, rispettivamente, in parole a cui sono associati i relativi articoli e preposizioni.

Alternativamente la generazione di detto nuovo primo vettore e di detto nuovo secondo vettore avvengono scomponendo detto primo e secondo documento testuale, rispettivamente, in righe di codice sorgente, ad ognuna di dette righe essendo associato il nome di una routine in cui è contenuta.

Vantaggiosamente i documenti testuali di cui avviene il confronto con il metodo della presente invenzione sono due versioni del codice sorgente di un programma per elaboratore ed i risultati del confronto vengono incrociati con banche dati contenenti informazioni su elementi costitutivi del linguaggio di programmazione in cui sono scritti detti codici sorgente, ad ognuno di detti elementi costitutivi essendo associato un peso o punteggio, il risultato del suddetto incrocio essendo costituito da una valorizzazione delle differenze tra dette due versioni di codice sorgente.

Vantaggiosamente il primo vettore viene generato da un codice sorgente di un software ed il secondo vettore viene generato in seguito a: la realizzazione di un documento a) comprendente detto codice sorgente, ad ogni riga di detto documento essendo abbinato un numero progressivo; l'esecuzione del suddetto software mediante un kit di test di software e, ad ogni esecuzione, la generazione di un documento c) in cui vengono memorizzate informazioni relative alle righe del codice sorgente del software eseguite ed il relativo numero progressivo; l'unione di tutti i documenti c) generati in un unico documento d); la generazione di detto secondo vettore da detto documento d).

Secondo un altro aspetto della presente invenzione gli scopi suddetti sono raggiunti tramite un mezzo che può essere letto da un elaboratore elettronico, detto mezzo essendo caratterizzato dal fatto di

contenere un codice che quando eseguito in un elaboratore elettronico dà origine ad un processo che comprende:

la generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

l'ordinamento alfa-numerico degli elementi all'interno di detto primo vettore e di detto secondo vettore;

l'eliminazione delle ripetizioni di elementi che si presentano più volte nel vettore con conteggio delle relative occorrenze,

la comparazione dell'ottenuto primo vettore con l'ottenuto secondo vettore per verificare le corrispondenze con elementi del secondo vettore.

Vantaggiosamente il processo generato dal codice memorizzato nel mezzo dell'invenzione comprende ulteriori preventive fasi di:

generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

la prima comparazione del primo vettore con il secondo vettore per verificare le corrispondenze di elementi del primo vettore con elementi del secondo vettore,

la verifica se nel primo vettore e nel secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un nuovo primo vettore e di un nuovo secondo vettore,

la ripetizione della suddetta prima comparazione eseguita con detto nuovo primo vettore e detto nuovo secondo vettore,

la verifica se nel nuovo primo vettore e nel nuovo secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un ulteriore nuovo primo vettore e di un ulteriore nuovo secondo vettore,

una seconda comparazione dell'ulteriore nuovo primo vettore con l'ulteriore nuovo secondo vettore,

la scrittura dei risultati del suddetto confronto;

in cui la generazione del suddetto ulteriore nuovo primo vettore avviene eliminando le ripetizioni degli elementi ripetuti di detto nuovo primo vettore e annotandone le occorrenze, e la generazione del suddetto ulteriore nuovo secondo vettore avviene eliminando le ripetizioni degli elementi ripetuti di detto nuovo secondo vettore e annotandone le occorrenze.

BREVE DESCRIZIONE DEI DISEGNI

Queste ed altre caratteristiche dell'invenzione risulteranno più facilmente comprensibili dalla seguente descrizione di una forma realizzativa preferita dell'invenzione, fornita come esempio non limitativo, con riferimento alle figure allegate nelle quali:

- la figura 1 mostra uno schematico diagramma di flusso che illustra in modo molto generale le modalità di funzionamento del metodo dell'invenzione in diverse tipologie di documenti di testo;
- la figura 2 mostra un diagramma di flusso che illustra la modalità di funzionamento generale di un metodo di confronto secondo la presente invenzione;
- la figura 3 mostra un diagramma di flusso che illustra in dettaglio le fasi del metodo di fig.2.

DESCRIZIONE DELLE FORME REALIZZATIVE PREFERITE

Con riferimento alla figura 1 è indicato complessivamente con 100 un diagramma di flusso che illustra in modo molto generale le modalità di funzionamento del metodo della presente invenzione ed i possibili campi di applicazione. Con 101 è indicato il confronto tra due documenti testuali eseguito riga per riga secondo il metodo della presente invenzione. Nell'eseguire il confronto secondo il metodo della presente invenzione si possono presentare diversi casi che vengono gestiti in modo diverso. Quando il documento di testo è un documento di dati, 103, solitamente non si incontrano problemi e dal confronto riga per riga si ottiene un risultato corretto che può essere immediatamente prodotto. Quando il documento di testo è un documento scritto ad esempio in italiano o altra lingua per la lettura da parte di esseri umani si può incontrare il problema della individuazione delle righe da confrontare, 102, in questo caso il metodo della presente invenzione utilizza una particolare modalità di scomposizione parola per parola, 105, che permette di produrre un risultato esente da errori. Come ulteriore sviluppo, 107, in seguito all'utilizzo del metodo dell'invenzione possono essere individuate nel documento di partenza ed in quello di arrivo le parole o diciture più numerose, per esempio per individuare l'argomento di cui tratta il documento. Quando il documento di testo è un codice sorgente di un programma per elaboratore, 104, si può presentare il problema della presenza di righe uguali ripetute in parti diverse del documento. In questo caso con il metodo dell'invenzione le righe di codice uguali vengono distinte assegnando loro ulteriori attributi, 106, così da poter operare correttamente il confronto. In questo caso, oltre a produrre un risultato corretto il confronto eseguito con il metodo dell'invenzione può portare alla possibilità di implementare ulteriori funzioni particolarmente interessanti quali l'inserimento di istruzioni che

eseguono la tracciatura delle varie routine/funzioni del codice sorgente, 108, particolarmente utili in fase di test del programma, oppure funzioni di automatizzazione della valutazione quantitativa e qualitativa delle modifiche apportate al programma rispetto ad una versione precedente dello stesso, 109.

Il metodo di confronto della presente invenzione è definito nelle sue linee generali con riferimento al diagramma di flusso, 200, di fig. 2. In seguito alla selezione di due documenti testuali da confrontare, 201, i documenti vengono convertiti in vettori e si avanza alternativamente nei due documenti eseguendo un confronto riga per riga, 202. In seguito a questa operazione tramite il metodo si cercano di individuare righe eliminate e righe aggiunte del documento di partenza rispetto a quello di arrivo, 203. Se nel documento di partenza o nel documento di arrivo non ci sono righe ripetute, 204, allora è possibile produrre direttamente il risultato del confronto, 206. Se invece ci sono righe che si ripetono nel documento di partenza o nel documento di arrivo, 205, si presentano due possibilità: o le righe non sono ben definite, 208, oppure siamo di fronte ad un codice sorgente in cui una stessa riga ritorna più volte, 209. Nel primo caso, secondo il metodo dell'invenzione si procede a suddividere il documento per singole parole, accorpendo però nella parola articoli e preposizioni, 207. In questo modo vengono definiti nuovi vettori con i quali operare il confronto riga per riga. Se all'esecuzione del confronto non ci sono righe ripetute, 212 si produce il risultato, 206. Se invece si riscontra ancora il problema della presenza di righe uguali nel vettore di partenza o in quello di arrivo si esegue un ordinamento delle righe eliminando in ogni vettore le righe ripetute e conteggiando le occorrenze delle stesse, 213. Viene quindi eseguito nuovamente il confronto tra i due nuovi vettori ottenuti e viene prodotto il risultato, 214. Se i due documenti sono invece due versioni

del codice sorgente di un programma per elaboratore si esegue una diversa suddivisione dei due vettori da confrontare applicando ad ogni riga un ulteriore identificativo che rappresenta il nome della routine a cui tale riga appartiene, 211. I due nuovi vettori ottenuti vengono confrontati e se non ci sono righe ripetute si produce il risultato, 214. Se invece ci sono ancora righe ripetute nell'uno o nell'altro vettore si esegue un ordinamento con conteggio delle occorrenze come sopra delineato, 213, e quindi si procede nuovamente all'esecuzione del confronto producendo il risultato, 214.

Con il metodo dell'invenzione si ottiene un risultato esente da errori dovuti a righe ripetute in uno dei due documenti semplicemente eseguendo al massimo tre volte la scomposizione in vettori sui quali eseguire il confronto: la prima volta secondo l'immediata e convenzionale scomposizione in righe di testo, la seconda volta scomponendo o in parole con i relativi articoli e preposizioni oppure in righe provviste di un ulteriore attributo che rappresenta la routine in cui sono inserite, la terza volta ordinando le righe, eliminando quelle ripetute e conteggiando le occorrenze. Con un numero massimo di tre passaggi si ha la certezza che il confronto tra i due documenti non generi risultati falsi dovuti alla presenza di righe ripetute nell'uno o nell'altro documento.

Sulla base del metodo di confronto sopra delineato e le cui modalità di funzionamento verranno descritte con maggiore dettaglio nel seguito è possibile creare una molteplicità di applicativi ed utilities che possono essere anche riuniti in pacchetti in funzione del campo di utilizzo. Ad esempio alcune specifiche procedure del metodo di confronto possono essere utilizzate per eseguire ricerche di una o più parole all'interno di un documento o di una cartella e per controllare gli standard di programmazione. Altre procedure, in particolare quella di ordinamento, possono essere utilizzate

individualmente per eseguire l'ordinamento degli elementi di un documento con eliminazione e conteggio delle occorrenze delle righe ripetute. In questo caso gli elementi con un numero maggiore di occorrenze costituiranno presumibilmente l'argomento principale del documento. Con semplici aggiunte di specifiche procedure possono essere create particolari utilities per i creatori di programmi per elaboratori. Ad esempio può essere eseguita la tracciatura dei documenti inserendo nel codice delle righe inserite automaticamente. Dopo quasi tutte le righe del codice sorgente possono essere inserite righe di codice che valorizzano una opportuna variabile che contiene il testo dell'ultima istruzione "vera" precedente. Nel costrutto di gestione dell'errore viene quindi fatto scrivere su un documento di testo le informazioni relative al documento, al blocco di codice e alle ultime istruzioni. Nel caso che avvenga una generazione massiccia di righe di report può essere utilizzato il metodo "sort", descritto di seguito, per effettuare una eliminazione selettiva delle informazioni meno importanti. La suddetta metodologia di tracciatura può essere utilizzata per eseguire il debug del programma in modo rapido ed efficiente. Ancora a vantaggio di programmatori, possono essere creati kit di test particolarmente efficienti. Inoltre, per tutti i documenti di una cartella contenenti codice sorgente possono essere analizzati i punti salienti del linguaggio, scelte logiche ed elaborazioni cicliche, riportandole in reports che permettono all'analista di verificare se il programma è conforme alle specifiche e agli standard di programmazione. È possibile inoltre creare strumenti per valutare lo sviluppo di progetti software. Infatti, possono essere analizzate le caratteristiche intrinseche del software quali scelte logiche, elaborazioni cicliche, numero di righe del documento e dei blocchi di codice di documenti di input o di output, etc, indipendentemente dal linguaggio di programmazione del progetto. In

questo modo possono essere creati report, ad esempio giornalieri che danno una misura dello sviluppo del progetto. Inoltre, nell'esecuzione del confronto di codici software ogni riga di codice può essere suddivisa in due parti: da un lato le parole chiave, cioè parole standard del linguaggio utilizzato, la cui modifica comporta un cambiamento delle funzionalità, dall'altra parte i nomi di variabili e di altri elementi scritti a discrezione dell'utente, la cui modifica può essere massiva senza effetti complessivamente rilevanti. In questo modo si ottiene anche una corretta valutazione a prescindere dalle eventuali modifiche massive eseguite sul codice sorgente. Addirittura, utilizzando il metodo dell'invenzione può essere realizzato uno strumento per valutare la completezza dei test del codice sorgente e per l'individuazione di parti di programma non testate o che addirittura non verranno mai raggiunte. Uno strumento come sopra delineato può essere realizzato come segue. Si realizza un documento a) con il codice sorgente del software da testare abbinato ad un numero progressivo che indica la posizione della riga. Si realizza un codice sorgente alternativo b) che per ogni riga di programma fa scrivere in un documento, quando il programma viene eseguito, la riga eseguita e il relativo numero progressivo. Il codice sorgente alternativo deve ovviamente aggiungere questa funzionalità di scrittura riga per riga senza alterare la funzionalità del software originale. Si effettuano quindi tutti i test possibili del software con il programma alternativo generando una certa quantità di documenti c). Si esegue l'unione dei suddetti documenti. Si ordina il documento assemblato d) riunendo insieme le righe uguali per numero e contenuto e conteggiando quante volte tali righe sono state trovate. Si confrontano i documenti a) e d) e si ottiene che le righe che compaiono nel documento a) (listato del programma) e non nel documento d) (righe effettivamente eseguite) sono evidentemente

righe che il programma non ha elaborato o perché non sono raggiungibili o perché i test non sono completi.

Questi ed altri applicativi possono essere generati applicando in tutto o in parte il metodo di confronto della presente invenzione di cui nel seguito viene descritta in dettaglio procedure di confronto.

Con riferimento alla fig. 3, è indicata complessivamente con 300 una procedura di confronto tra due vettori. I vettori sono ottenuti convertendo opportunamente due documenti testuali da confrontare. Tramite la definizione di procedure cicliche che verranno nel seguito descritte si esegue il confronto tra elementi del primo vettore ed elementi del secondo vettore avanzando opportunamente nei due vettori in modo da ottenere un confronto efficace ed esente da errori.

La procedura di confronto comprende l'esecuzione di un ciclo sul primo vettore, 305, che consiste nell'eseguire su ognuno degli elementi del primo vettore una determinata procedura. Tale procedura è a sua volta un ciclo sul secondo vettore, 310, interno al primo, in cui vengono eseguite specifiche procedure su elementi del secondo vettore. Il ciclo 310 viene realizzato sugli elementi del secondo vettore che vanno dall'elemento successivo all'ultimo elemento del secondo vettore di cui sia stata già stabilita l'uguaglianza o meno con un corrispondente elemento del primo vettore fino all'ultimo elemento del secondo vettore. Le suddette procedure interne al ciclo 310 comprendono il confronto, 315, tra un elemento del primo vettore con un elemento del secondo vettore selezionati in funzione dell'avanzamento nei suddetti cicli 305 e 310. Se dal confronto 315 risulta che i due elementi non sono uguali e se è stato confrontato l'ultimo elemento del secondo vettore allora si registra l'elemento del primo vettore come elemento eliminato, 320, altrimenti si avanza all'elemento successivo del secondo vettore proseguendo il ciclo 310. Se, invece, dal confronto 315 risulta che i due elementi sono uguali si

esce dal ciclo 310 memorizzando quale elemento del secondo vettore risulta uguale all'elemento del primo vettore, 325. A questo punto, se non sono stati trovati precedentemente, nell'esecuzione del ciclo 310, elementi non corrispondenti, si memorizzano gli elementi confrontati come "uguali", 330, e si avanza all'elemento successivo del primo vettore, altrimenti il valore di una certa variabile X di tipo "true or false", 335, che, per chiarezza, potremmo definire come variabile di conferma di aver trovato nel secondo vettore l'elemento corrispondente ad un certo elemento del primo vettore viene impostata su un valore che indica la non conferma. Una volta impostato il valore della suddetta variabile la procedura prevede di eseguire nuovamente i cicli di confronto sul primo vettore e al suo interno sul secondo vettore in cui però i suddetti cicli comprendono gli elementi dei due vettori determinati come descritto di seguito. Se la lunghezza del primo vettore è superiore a ciò che è già stato esplorato nel secondo vettore (determinato dall'indice del secondo vettore precedentemente memorizzato in 325) allora si imposta la fine del ciclo interno sul primo vettore, 350, all'indice del punto già esplorato nel secondo vettore, 340, altrimenti imposto di percorrere il primo vettore fino all'ultimo elemento, 345. Il ciclo interno sul secondo vettore, 355, avviene invece dall'elemento successivo all'ultimo già confermato in 230 fino all'ultimo elemento esplorato il cui indice è stato memorizzato in 325. Nell'ambito di questi cicli la procedura prevede di volta in volta il confronto, 360, degli elementi selezionati dei due vettori. Se dal confronto 360 risulta che i due elementi non sono uguali e se è stato confrontato l'ultimo elemento del secondo vettore allora si prosegue il ciclo interno 350 sul primo vettore avanzando di un elemento, altrimenti si avanza all'elemento successivo del secondo vettore proseguendo il ciclo 355. Se, invece, dal confronto 360 risulta che i due elementi sono uguali si esce dal

ciclo interno 350 e la suddetta variabile X viene impostata sul valore di conferma, 365. Infine, terminato il ciclo interno 350, viene eseguita una procedura che prende in considerazione il valore della variabile X. Se il valore della variabile è di non conferma (335) allora viene eseguito un ciclo di memorizzazione/scrittura degli elementi del vettore due su cui è stato eseguito il ciclo interno 355 come “elementi aggiunti”, 370. Se invece il valore della variabile X è di conferma (365) allora viene eseguito un ciclo di memorizzazione/scrittura relativo al primo vettore in cui gli elementi del primo vettore da quello successivo all’ultimo già determinato all’ultimo esplorato vengono contrassegnati come elementi eliminati, 380. In seguito il valore della variabile X viene impostato sul valore non confermato, 380, e quindi si torna all’esecuzione del ciclo principale sul primo vettore (305). Nel caso, invece, che il ciclo principale sia terminato si procede a memorizzare/scrivere, 385, relativamente al secondo vettore, che gli elementi da quello successivo all’ultimo confermato fino all’ultimo elemento del vettore sono elementi inseriti.

La procedura sopra descritta consente di identificare in modo efficace gli elementi uguali dei due vettori, gli elementi eventualmente eliminati nel primo vettore e gli elementi eventualmente aggiunti nel secondo vettore. Tuttavia, la procedura suddetta potrebbe risultare non accurata nel caso che all’interno di almeno uno dei due vettori ci fossero elementi che si ripetono identici. In questo caso, come già accennato, per superare tale problema il metodo della presente invenzione prevede che si proceda a eseguire secondo una modalità diversa la scomposizione in vettori dei due documenti di cui deve essere effettuato il confronto. In ultima istanza, qualora il problema delle righe ripetute si presentasse anche dopo aver trasformato i documenti in vettori con la seconda modalità di scomposizione si esegue per la terza volta la procedura di

confronto dopo aver però eseguito un cosiddetto “sort”. Il sort, secondo il metodo dell’invenzione, consiste nel trasformare i due documenti in vettori secondo la prima o la seconda modalità di trasformazione, nell’ordinare quindi, ad esempio in ordine alfa-numerico, gli elementi di ciascun vettore, e nell’eliminare le ripetizioni degli elementi uguali annotando separatamente per ogni elemento quante volte si presenta nel vettore. Dopo aver eseguito la suddetta operazione di sort viene di nuovo eseguita una procedura di confronto sostanzialmente uguale alla procedura 300 sopra descritta. Le differenze rispetto alla suddetta procedura riguardano la fase 320, in cui quando si memorizza/scrive l’elemento del primo vettore come eliminato se ne conteggia anche il numero di occorrenze e nella la fase 330, in cui prima di poter memorizzare/scrivere i due elementi come uguali la procedura prevede una fase di verifica della corrispondenza tra le occorrenze dei due elementi: se il numero di occorrenze nel primo vettore è maggiore del numero di occorrenze nel secondo vettore si hanno degli elementi eliminati nel primo vettore, se il numero di occorrenze nel primo vettore è inferiore al numero di occorrenze nel secondo vettore si hanno degli elementi aggiunti nel secondo vettore, se, infine, il numero di occorrenze è uguale allora anche i due elementi sono uguali.

Secondo un aspetto particolarmente vantaggioso della presente invenzione il confronto secondo le procedure sopra descritte viene eseguito tra i codici sorgente di un programma per elaboratore, ad esempio tra due versioni successive del codice sorgente in fase di sviluppo. Nel caso del codice sorgente è possibile creare delle banche dati in cui vengono memorizzati e classificati gli elementi fondamentali di un certo linguaggio di programmazione. In tal modo, i risultati del confronto eseguito secondo il metodo dell’invenzione possono essere incrociati con le informazioni contenute nelle suddette

banche dati in modo da verificare quale tipologia di modifiche è stata apportata al codice sorgente tra una versione e la successiva. Ad esempio, si possono conteggiare il numero di scelte logiche, di elaborazioni cicliche o di altri elementi costitutivi del linguaggio che sono stati modificati, mentre si può non tener conto di modifiche, anche massive, di poca importanza quali il cambiamento di nome di una variabile o altro. Ai vari elementi del linguaggio di programmazione viene inoltre assegnato un peso che ne stabilisce la rilevanza. In tal modo si ottiene un vero e proprio punteggio delle modifiche apportate.

Ovviamente varianti e modifiche possono essere apportate alle procedure di confronto di documenti testuali ed ulteriori applicazioni delle stesse possono essere elaborate, pur sempre rimanendo all'interno dell'ambito di protezione definito dalle rivendicazioni seguenti.

RIVENDICAZIONI

1. Metodo per il confronto di documenti testuali in cui il metodo comprende:

la generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

l'ordinamento alfa-numerico degli elementi all'interno di detto primo vettore e di detto secondo vettore;

l'eliminazione delle ripetizioni di elementi che si presentano più volte nel vettore con conteggio delle relative occorrenze,

la comparazione dell'ottenuto primo vettore con l'ottenuto secondo vettore per verificare le corrispondenze dei relativi elementi.

2. Metodo per il confronto di documenti testuali secondo la rivendicazione 1 in cui il metodo comprende ulteriori preventive fasi di:

generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

la prima comparazione del primo vettore con il secondo vettore per verificare le corrispondenze di elementi del primo vettore con elementi del secondo vettore,

la verifica se nel primo vettore e nel secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un nuovo primo vettore e di un nuovo secondo vettore,

la ripetizione della suddetta prima comparazione eseguita con detto nuovo primo vettore e detto nuovo secondo vettore,

la verifica se nel nuovo primo vettore e nel nuovo secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un ulteriore nuovo primo vettore e di un ulteriore nuovo secondo vettore,

una seconda comparazione dell'ulteriore nuovo primo vettore con l'ulteriore nuovo secondo vettore,

la scrittura dei risultati del suddetto confronto;

in cui la generazione del suddetto ulteriore nuovo primo vettore ed ulteriore nuovo secondo vettore avvengono eliminando le ripetizioni degli elementi ripetuti di detto nuovo primo vettore e di detto nuovo secondo vettore ed annotandone le occorrenze.

3. Metodo secondo la rivendicazione precedente in cui detta generazione di detto nuovo primo vettore e di detto nuovo secondo vettore avvengono scomponendo detto primo e secondo documento testuale, rispettivamente, in parole a cui sono associati i relativi articoli e preposizioni.

4. Metodo secondo la rivendicazione 2 in cui detta generazione di detto nuovo primo vettore e di detto nuovo secondo vettore avvengono scomponendo detto primo e secondo documento testuale, rispettivamente, in righe di codice sorgente, ad ognuna di dette righe essendo associato il nome di una routine in cui è contenuta.

5. Metodo secondo una delle rivendicazioni precedenti in cui detti documenti testuali sono due versioni del codice sorgente di un programma per elaboratore ed in cui i risultati del metodo di confronto vengono combinati con banche dati contenenti informazioni su elementi costitutivi del linguaggio di programmazione in cui sono scritti detti codici sorgente, ad ognuno di detti elementi costitutivi essendo associato un peso o punteggio, il risultato del suddetto incrocio essendo costituito da una

valorizzazione delle differenze tra dette due versioni di codice sorgente.

6. Metodo secondo la rivendicazione 1 in cui detta generazione di un primo vettore di elementi testuali e di un secondo vettore di elementi testuali avviene con detto primo vettore ottenuto da un codice sorgente di un software e detto secondo vettore ottenuto in seguito a: la realizzazione di un documento a) comprendente detto codice sorgente, ad ogni riga di detto documento a) essendo abbinato un numero progressivo; l'esecuzione del suddetto codice sorgente mediante un kit di test di software e, ad ogni esecuzione, la generazione di un documento c) in cui vengono memorizzate informazioni relative alle righe del codice sorgente del software eseguite ed il relativo numero progressivo; l'unione di tutti i documenti c) generati in un unico documento d); la generazione di detto secondo vettore da detto documento d).

7. Mezzo che può essere letto da un elaboratore elettronico, detto mezzo essendo caratterizzato dal fatto di contenere un codice che quando eseguito in un elaboratore elettronico dà origine ad un processo che comprende:

la generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

l'ordinamento alfa-numerico degli elementi all'interno di detto primo vettore e di detto secondo vettore;

l'eliminazione delle ripetizioni di elementi che si presentano più volte nel vettore con conteggio delle relative occorrenze,

la comparazione dell'ottenuto primo vettore con l'ottenuto secondo vettore per verificare le corrispondenze con elementi del secondo vettore.

8. Mezzo che può essere letto da un elaboratore elettronico, detto mezzo essendo caratterizzato dal fatto di contenere un codice che quando eseguito in un elaboratore elettronico dà origine ad un processo secondo la rivendicazione precedente che comprende ulteriori preventive fasi di:

generazione di un primo vettore di elementi da un primo documento testuale e di un secondo vettore di elementi da un secondo documento testuale,

la prima comparazione del primo vettore con il secondo vettore per verificare le corrispondenze di elementi del primo vettore con elementi del secondo vettore,

la verifica se nel primo vettore e nel secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un nuovo primo vettore e di un nuovo secondo vettore,

la ripetizione della suddetta prima comparazione eseguita con detto nuovo primo vettore e detto nuovo secondo vettore,

la verifica se nel nuovo primo vettore e nel nuovo secondo vettore sono presenti elementi ripetuti nello stesso vettore, e se sono presenti elementi ripetuti:

la generazione di un ulteriore nuovo primo vettore e di un ulteriore nuovo secondo vettore,

una seconda comparazione dell'ulteriore nuovo primo vettore con l'ulteriore nuovo secondo vettore,

la scrittura dei risultati del suddetto confronto;

in cui la generazione del suddetto ulteriore nuovo primo vettore avviene eliminando le ripetizioni degli elementi ripetuti di detto nuovo primo vettore e annotandone le occorrenze, e la generazione del suddetto ulteriore nuovo secondo vettore avviene eliminando le

ripetizioni degli elementi ripetuti di detto nuovo secondo vettore e annotandone le occorrenze.

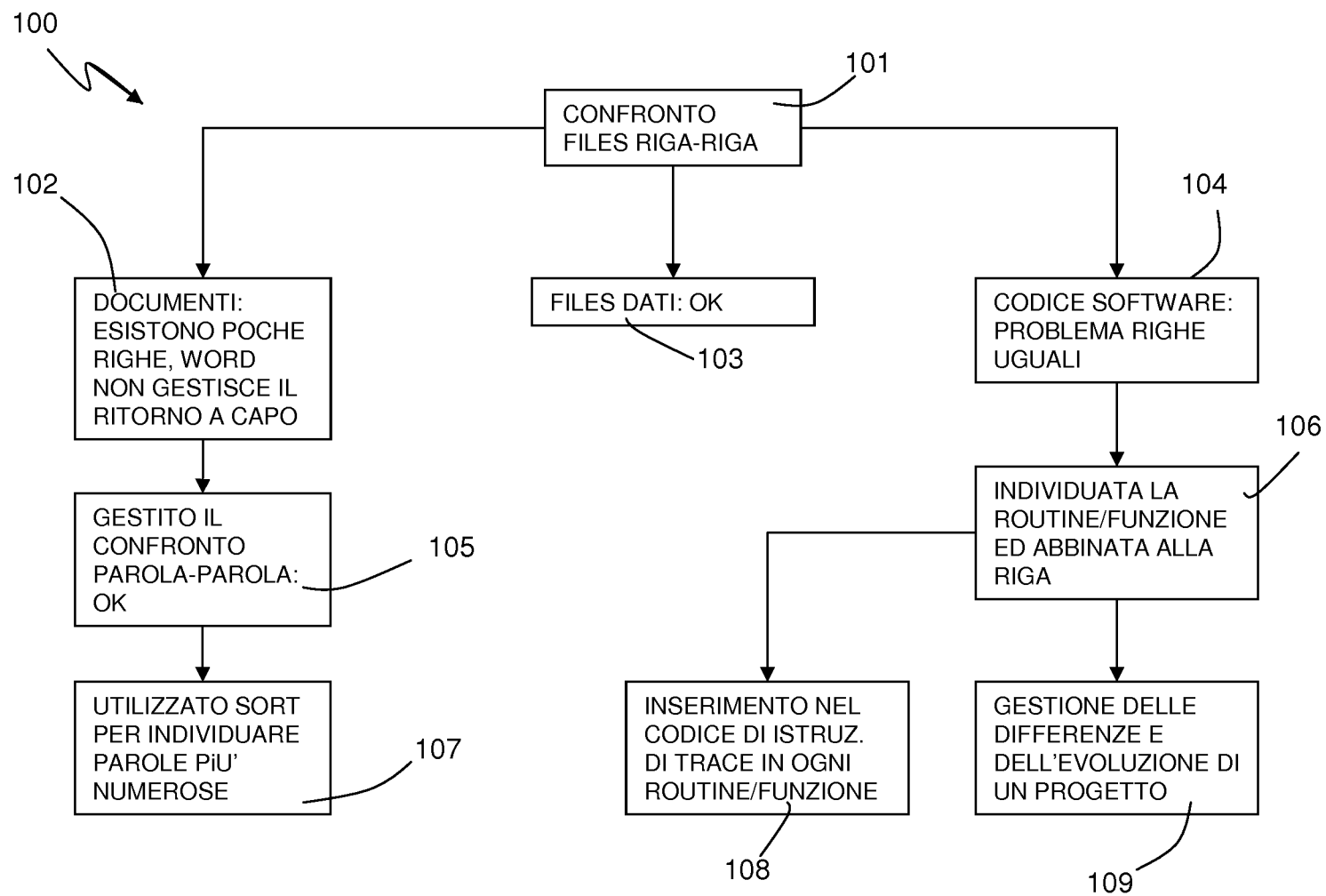


FIG. 1

