(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 693 33 422 T2** 2004.12.16

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 583 559 B1**
(21) Deutsches Aktenzeichen: **693 33 422.3**
(96) Europäisches Aktenzeichen: **93 108 356.2**
(96) Europäischer Anmeldetag: **24.05.1993**
(97) Erstveröffentlichung durch das EPA: **23.02.1994**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **25.02.2004**
(47) Veröffentlichungstag im Patentblatt: **16.12.2004**

(51) Int Cl.[7]: **G06F 17/30**

(30) Unionspriorität:
923203 **31.07.1992** **US**

(73) Patentinhaber:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Teufel, F., Dipl.-Phys., Pat.-Anw., 70569 Stuttgart

(84) Benannte Vertragsstaaten:
AT, BE, CH, DE, ES, FR, GB, IT, LI, NL, SE

(72) Erfinder:
Califano, Andrea, New York, US

(54) Bezeichnung: **Auffindung von Zeichenketten in einer Datenbank von Zeichenketten**

Beschreibung

1. GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft das Gebiet der Suche nach Zeichenketten (Zeichenfolgen) in einer Datenbank. Insbesondere werden gemäß der Erfindung Zeichenketten in einer Datenbank gefunden, die einer vorgegebenen Referenz-Zeichenkette ähnlich oder identisch sind.

2. BESCHREIBUNG DES STANDS DER TECHNIK

[0002] Es gibt zahlreiche herkömmliche Verfahren, um eine in einer Datenbank mit vielen Zeichenketten vorkommende bestimmte Zeichenfolge, die auch als Referenz-Zeichenfolge oder Referenz-Zeichenkette bezeichnet wird, zu finden. (Ein Zeichen ist ein Symbol wie beispielsweise ein Buchstabe, ein Wort, ein Klang, ein Bitmuster oder eine andere Beschreibungsform, die im vorliegenden Fall in einer Folge anderer Zeichen auftreten kann.) Einige dieser Verfahren sind zur Ausführung spezieller Tasks entwickelt worden, z. B. zum Auffinden einer genauen oder ähnlichen Folge spezieller Zeichen wie z. B. von Nukleotiden (oder Aminosäuren) in einer langen Kette von Nukleotiden (oder Aminosäuren), aus denen ein DNA-Molekül (oder Proteinmolekül) besteht. (Zwei Zeichenfolgen sind einander ähnlich, wenn sie durch Insertion, Deletion oder Änderung einer Anzahl von Zeichen identisch gemacht werden können, die kleiner als eine vorgegebene Anzahl von Zeichen in einer der Zeichenfolgen ist.) Einige dieser herkömmlichen Vergleichsverfahren sind: der Needlemann-Wunsch-Algorithmus oder die ursprünglichen Wilbur-Lipman-Algorithmen FASTA, FASTP und BLAST.

[0003] Der Needlemann-Wunsch-Algorithmus ist ein dynamisches Programmierverfahren. Alle in den beiden zu vergleichenden Zeichenfolgen enthaltenen Zeichen werden paarweise untersucht, um alle Zuordnungsmöglichkeiten zwischen den beiden Zeichenketten zu berechnen. Den Deletionen, Insertionen und Änderungen wird ein Aufwandswert zugewiesen. Dann wird diejenige Zuordnung gewählt, welche den geringsten Wert des Gesamtaufwands aufweist. Da die erforderliche Rechenleistung dem Produkt der Längen beider zu vergleichender Zeichenfolgen proportional ist, ist dieses Verfahren sehr aufwändig.

[0004] Der Wilbur-Lipman-Algorithmus vergleicht zusammenhängende Tupel geringer Länge in der Original-Zeichenkette und in der Referenz-Zeichenkette miteinander. Die Tupel werden mittels einer Referenztabelle, die anhand der Referenz-Zeichenkette erstellt wurde, für die beiden Zeichenketten verglichen. Für jeden Vergleich wird eine Bewertungszahl berechnet und dann die beste Bewertungszahl gewählt. Deshalb wird jedes Mal eine neue Referenztabelle erstellt, wenn eine neue Referenz-Zeichenfolge mit der Datenbank verglichen werden soll. Da der gesamte Satz Original-Zeichenketten mit der Referenztabelle verglichen werden muss, verdoppelt sich die zum Vergleich mit einer Datenbank von insgesamt 2N Nukleotiden oder Aminosäuren erforderliche Rechenleistung gegenüber dem Vergleich mit einer Datenbank von lediglich N Nukleotiden oder Aminosäuren. Mit anderen Worten, die Anzahl der Vergleiche mit der Referenztabelle ist mindestens gleich der Gesamtanzahl der in allen Original-Zeichenketten vorhandenen Nukleotide (Aminosäuren).

[0005] Die Algorithmen FASTP und FASTA stellen Verbesserungen des ursprünglichen WILBUR-LIPMAN-Verfahrens dar. Durch eine Austauschmatrix zur Bewertung der Vergleichsoperationen wird eine höhere Empfindlichkeit erreicht. Während der Evolution oft vorkommende Mutationen (Deletionen, Insertionen und Nukleotidaustausch) erhalten eine höhere Bewertungszahl als seltener vorkommende Mutationen. Hierbei handelt es sich jedoch immer noch um einen sequenziellen Ansatz.

[0006] Das BLAST-Verfahren unternimmt erst dann einen kompletten Vergleich zwischen der Original- und der Referenz-Zeichenkette, wenn diese in einem sehr schnell durchzuführenden Vortest eine Mindestähnlichkeit aufweisen. Bei diesem Test wird heuristisch ermittelt, ob die Länge des MSP (maximal segment pair, größtes Segmentpaar) einen bestimmten Schwellenwert überschreitet. Das MSP ist dasjenige Paar identischer Teilzeichenketten in der Referenz-Zeichenkette und der Original-Zeichenkette, welches die beste Mutations-Bewertungszahl aufweist. Wenn dieser Test erfolgreich bestanden wurde, erfolgt mittels der Algorithmen vom Typ FASTP-FASTA eine umfassendere und aufwändigere Ähnlichkeitsanalyse. Hierdurch wird der Rechenaufwand verringert und gleichzeitig in Kauf genommen, dass einige Übereinstimmungstreffer unberücksichtigt bleiben, welche die Ausgangskriterien nicht erfüllen. Etwa 20% der durch den Needleman-Wunsch-Algorithmus gefundenen Übereinstimmungen werden durch BLAST nicht erfasst. Außerdem bleibt dieser Ansatz in seinem Wesen sequenziell, da für jede Zeichenkette aus der Menge der Original-Zeichenketten eine Berechnung durchgeführt werden muss.

p3. BESCHREIBUNG DER PROBLEME NACH DEM STAND DER TECHNIK

[0007] Mit den derzeitigen Verfahren war es möglich, zwei Zeichenketten (wobei es sich bei den Zeichen speziell um Nukleotide oder Aminosäuren handelt) eins zu eins, d. h. sequenziell, ohne großen Aufwand miteinander zu vergleichen. Beim Stand der Technik ist es jedoch oft schwierig, alle oder zumindest die meisten möglichen Übereinstimmungen einer Referenz-Zeichenfolge in einer Datenbank mit Original-Zeichenketten zu finden, ohne Berechnungen an allen oder fast allen Zeichen der Original-Zeichenfolgen durchzuführen. Beim gegenwärtigen Stand der Computertechnik lassen sich diese Aufgaben in sehr großen Datenbanken nur mit einem unvertretbar hohen Zeitaufwand ausführen.

[0008] Deshalb besteht seit langem ein Bedarf an einem indexierten Verfahren zur Ermittlung einer annähernden oder genauen Übereinstimmung zwischen einer Referenz-Zeichenkette und einer Zeichenfolge in einer oder mehreren Original-Zeichenketten in sehr großen Datenbanken. Außerdem wird gefordert, die Lage dieser ähnlichen oder identischen Zeichenfolgen in den Original-Zeichenketten und den Grad ihrer Übereinstimmung mit einer Referenz-Zeichenkette schnell und mit geringem Aufwand zu ermitteln. Insbesondere besteht auf dem Gebiet der Genomkartierung seit langem der Wunsch nach einem Verfahren, welches mittels der neuesten Rechentechnik Übereinstimmungen zwischen den Nukleotidsequenzen in einer Datenbank mit bis zu 4 Milliarden Nukleotiden nachweisen kann.

[0009] Die Verfahren nach dem Stand der Technik sind nicht in der Lage, die Lage von Zeichenfolgen in Original-Zeichenketten großer Datenbanken schnell und ohne großen Aufwand zu ermitteln, wenn sie nach einer Übereinstimmung mit einer Referenz-Zeichenkette suchen. Das liegt daran, dass die Verfahren nach dem Stand der Technik bei der Vergleichsprozedur die gesamte Datenbank der Original-Zeichenketten durchsuchen müssen. Die Verfahren nach dem Stand der Technik müssen auf der Suche nach passenden Zeichenketten die gesamte Datenbank durchsuchen, da sie nicht in der Lage sind, ein Indexierungsverfahren bereitzustellen, welches nur diejenigen Original-Zeichenketten schnell und genau ausfindig macht, in denen mögliche passende Zeichenfolgen enthalten sind.

ZIELE DER ERFINDUNG

[0010] Ein Ziel der vorliegenden Erfindung besteht in der Bereitstellung eines verbesserten Verfahrens zum Auffinden von Zeichenfolgen, die einer Referenz-Zeichenfolge in einer oder mehreren Original-Zeichenketten in einer Datenbank mit einer oder mehreren Original-Zeichenketten identisch oder ähnlich sind.

[0011] Ein weiteres Ziel der vorliegenden Erfindung besteht in der Bereitstellung eines verbesserten Verfahrens zum Auffinden von Zeichenfolgen mittels Indexierungs- und Hashverfahren, die einer Referenz-Zeichenfolge in Original-Zeichenketten in einer Datenbank mit einer oder mehreren Original-Zeichenketten identisch oder ähnlich sind.

[0012] Ein weiteres Ziel der vorliegenden Erfindung besteht in der Bereitstellung eines verbesserten Verfahrens zum Auffinden von Nukleotidsequenzen (oder Aminosäuresequenzen), die mit einer Referenz-Nukleotidsequenz (oder Referenz-Aminosäuresequenz) in einer Datenbank identisch oder dieser ähnlich sind, welche eine Vielzahl von Original-Nukleotidketten (oder Original-Aminosäureketten) enthält, welche ein DNA-Molekül (oder ein Proteinmolekül) darstellen.

[0013] Ferner besteht ein Ziel der vorliegenden Erfindung in der Bereitstellung eines verbesserten Verfahrens zur Spracherkennung durch Auffinden von Phonemfolgen, die einer Referenz-Phonemfolge in einer Datenbank mit Original-Sprachphonemfolgen ähnlich sind.

[0014] Ein weiteres Ziel der vorliegenden Erfindung besteht in der Bereitstellung eines verbesserten Verfahrens zur Musikerkennung durch Auffinden von Notenfolgen, die einer Referenz-Notenfolge in einer Datenbank mit Original-Notenfolgen ähnlich sind.

ÜBERBLICK ÜBER DIE ERFINDUNG

[0015] Die Erfindung gemäß den beiliegenden Ansprüchen stellt ein Verfahren und ein System bereit, welches in der Lage ist, das vorkommen einer Zeichenfolge in einer oder mehreren Original-Zeichenketten in einer Datenbank mit Original-Zeichenketten zu finden, das einer anderen als Referenz-Zeichenfolge bezeichneten Zeichenfolge identisch oder ähnlich ist. Die Erfindung weist eine hohe Wahrscheinlichkeit auf, alle Fälle der Zeichenfolgen in den Original-Zeichenketten schnell und präzise aufzufinden, die mit der Referenz-Zeichenfol-

ge ganz (identisch) oder fast genau übereinstimmen.

[0016] Für jede Original-Zeichenkette wird eine große Anzahl von Indizes erzeugt und zum Speichern eines die Original-Zeichenkette kennzeichnenden Datensatzes in einer Referenztabelle verwendet. Während der Erkennungsphase wird aus einer Referenz-Zeichenkette eine große Anzahl von Indizes gebildet. Diese dienen dazu, die Daten aus der Referenztabelle abzurufen und den Nachweis einer oder mehrerer Original-Zeichenketten abzuspeichern. Auf diese Weise kann die Verarbeitung zum großen Teil vor der Erkennungsphase stattfinden und somit die Erkennungszeit stark verkürzt werden.

[0017] Zur Bildung von Indizes bildet das Verfahren zuerst „Tupel". Hierzu werden aus einer Original-Zeichenfolge in der Datenbank zuerst eine Anzahl aus zusammenhängenden Zeichen bestehender Original-Teilzeichenfolgen ausgewählt. Die Länge jeder Original-Teilzeichenfolge dieses Satzes beträgt nur wenige Zeichen, mindestens jedoch ein Zeichen. Die Länge aller Teilzeichenfolgen des Satzes kann eine fest vorgegebene Anzahl von Zeichen betragen. Die Länge einiger oder aller Teilzeichenfolgen des Satzes kann jedoch eine unterschiedliche Anzahl von Zeichen betragen. Unter Verwendung dieses Satzes von Original-Teilzeichenketten wird ein Satz von Tupeln gebildet. Mindestens ein Tupel dieses Satzes wird durch Zusammenfügen mindestens zweier verschiedener, nicht zusammenhängender Teilzeichenfolgen des Satzes gebildet. (Andere Teilzeichenfolgen dieses Tupels oder in dem Satz von Tupeln können zusammenhängend oder nicht zusammenhängend sein.) Diese Tupel werden auch als „j-Tupel" bezeichnet, wobei j die Anzahl der Original-Zeichenfolgen bedeutet, aus denen das Tupel gebildet wird. Die aus einer Original-Zeichenkette gebildeten Tupel heißen Originaltupel.

[0018] Dann wird ein eindeutiger Index erzeugt und auf Basis der Werte der in dem Tupel enthaltenen Zeichen jedem Tupel zugewiesen. Die den Originaltupeln zugewiesenen Indizes heißen Originalindizes. Üblicherweise werden die Originalindizes mittels eines Algorithmus aus den Werten der in den j-Tupeln enthaltenen Zeichen erzeugt und haben die Form eines Wertes wie zum Beispiel einer Zahl (normalerweise einer ganzen Zahl). Jeder Originalindex ist einer eindeutigen Zelle in einer Speicher-Referenzstruktur zugeordnet, üblicherweise einer Matrix. Der Index dient zur Identifizierung und/oder für den Zugriff auf eine zugehörige eindeutige Zelle in der Referenzstruktur. In der zum Index gehörenden Zelle der Referenzstruktur ist ein Datensatz gespeichert, der eine Original-Zeichenkette beschreibt, aus welcher das Tupel (und der Originalindex) erzeugt wurde. Eine Zelle ist in der Lage, entweder eine fest vorgegebene oder eine beliebige Anzahl dieser Datensätze zu speichern. Zum Beispiel kann ein in der Zelle gespeicherter Datensatz einen Verweis auf die Original-Zeichenkette enthalten. Dieser Verweis, der hier auch als Zeiger bezeichnet wird, stellt ein Mittel zur Kennzeichnung einer der Original-Zeichenketten dar. Man beachte, dass ein Verweis als Zeiger (die Speicheradresse) eines bestimmten Zeichens der Original-Zeichenkette oder als Index in einer Zeigertabelle oder nach einem anderen in der Technik bekannten Verfahren realisiert werden kann. Der Zeiger (Verweis) zeigt diejenige Original-Zeichenkette in der Datenbank an, von welcher das durch den Originalindex eindeutig definierte Originaltupel abgeleitet wurde. Die Referenzstruktur kann auch weitere als Verschiebungswert bezeichnete Daten enthalten, welche die Position des (zur Erzeugung des Indexes verwendeten) Tupels in der Original-Zeichenkette angibt. Zu diesen Positionsdaten können gehören: 1. eine Verschiebung von einer bestimmten Position (Zeichen) der Original-Zeichenkette zum ersten Zeichen der ersten Original-Teilzeichenkette, die zur Bildung des Tupels dient oder 2. eine Verschiebung, die gleich der mittleren Position des ersten Zeichens aller Original-Teilzeichenketten ist, welche das Tupel bilden oder 3. eine andere Verschiebung, die aus der Position mehrerer Zeichen in einer oder mehreren Original-Teilzeichenketten berechnet werden kann.

[0019] Nach der Verarbeitung aller gewünschten Original-Zeichenketten wird die Referenz-Zeichenfolge mit den Original-Zeichenketten verglichen. Hierzu werden mittels der oben beschriebenen oder einer ähnlichen Prozedur aus einer Referenz-Zeichenfolge Referenztupel und deren eindeutig kennzeichnende Referenzindizes gebildet.

[0020] Die Referenzindizes werden unter Verwendung der oben beschriebenen Referenzstruktur mit den Originalindizes verglichen. Ein Referenzindex dient dazu, auf eine Zelle in der Referenzstruktur zu zeigen. Wenn dieser Referenzindex auf eine Zelle zeigt, welche einen oder mehrere Datensätze für Original-Zeichenketten enthält, erhält man eine oder mehrere passende Übereinstimmungen. Wenn der Referenzindex auf eine Zelle ohne Daten zeigt, erhält man keine Übereinstimmung.

[0021] Für jede in der durch den Referenzindex indexierten Zelle gespeicherten Datensatz wird ein Zählwert in einer zweiten Datenstruktur mit der Bezeichnung Nachweis-Sammeltabelle EIT (Evidence Integration Table) gespeichert, die zur Erfassung der Anzahl der Übereinstimmungen für eine bestimmte Original-Zeichenkette dient. Üblicherweise ist dies eine Hash-Tabelle. In dieser zweiten Struktur adressierte Zellen entsprechen Hy-

pothesen über Original-Zeichenketten, die mit der Referenz-Zeichenkette übereinstimmen. Die Zellen enthalten auch einen Wert, der anzeigt, wie oft eine Hypothese gezählt wurde. Zum Beispiel entsprechen die Zählwerteinträge in einer Zählzelle in dieser zweiten Struktur den Zählwerten bestimmter Zeichenketten in der Ausgangs-Datenbank. Je ähnlicher die Referenz-Zeichenfolge einer Zeichenfolge in einer Original-Zeichenkette in der Datenbank ist, desto höher ist die Wahrscheinlichkeit, dass zwischen den Referenzindizes der Referenz-Zeichenfolge und den Originalindizes der Original-Zeichenkette mehr Übereinstimmungen vorkommen. Die Zählzellen in der zweiten Speicherstruktur, welche Original-Zeichenketten entsprechen, die den Referenz-Zeichenketten ähnlich oder identisch sind, weisen eine relativ hohe Anzahl von Zählwerten auf. Umgekehrt ist die Wahrscheinlichkeit der Übereinstimmung zwischen den Referenzindizes und den Originalindizes, die jedem aus der Original-Zeichenkette gebildeten Tupel eindeutig zugeordnet sind, umso geringer, je weniger die Referenz-Zeichenfolge mit den Zeichenfolgen in der Original-Zeichenkette übereinstimmt. Zählzellen in der EIT, die diesen Fällen entsprechen, enthalten nur wenige oder überhaupt keine Zählwerte. Deshalb weist die Anzahl der Zählwerte in jeder Zählzelle der EIT eine direkte Korrelation zum Übereinstimmungsgrad zwischen der durch die Zelle dargestellten Original-Zeichenkette und der Referenz-Zeichenfolge auf.

[0022] Und schließlich dienen die Daten der EIT dazu, die Position derjenigen Zeichenfolgen in einer Original-Zeichenkette in der Datenbank anzugeben, die der Referenz-Zeichenfolge (genau oder annähernd) entsprechen. Zuerst werden Zellen der EIT mit einem Wert ausgewählt, der oberhalb eines vorgegebenen Schwellenwertes liegt. Dann werden mit einem oder mehreren zugehörigen Zeigern eine oder mehrere Original-Zeichenketten in der Datenbank gesucht. Und schließlich wird die übereinstimmende Zeichenfolge der Original-Zeichenkette mittels der zu dem ausgewählten Index bzw. zu den ausgewählten Indizes gehörenden Verschiebungswerte gefunden.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0023] **Fig.** 1 zeigt ein Ablaufdiagramm des Gesamtverfahrens der Erfindung.

[0024] **Fig.** 2 zeigt eine Original-Zeichenkette und eine an der Position p_i der Original-Zeichenkette beginnende Teilzeichenkette mit der Länge von l Zeichen.

[0025] **Fig.** 3 zeigt die bevorzugte Ausführungsart einer Referenzstruktur.

[0026] **Fig.** 4 zeigt eine alternative Ausführungsart einer Referenzstruktur.

[0027] **Fig.** 5 zeigt die bevorzugte Ausführungsart einer Nachweis-Sammeltabelle (EIT).

[0028] **Fig.** 6 ist ein Ablaufdiagramm eines Computerprogramms, das Originaltupel und Originalindizes erzeugt und die zugehörigen Datensätze in der Referenzstruktur speichert.

[0029] **Fig.** 7 ist ein Ablaufdiagramm des Computerprogramms, das Referenztupel und Referenzindizes erzeugt, die Referenzindizes mit den Originalindizes vergleicht und die EIT aktualisiert.

[0030] **Fig.** 8 ist eine Tabelle der verwendeten Symbole.

DETAILLIERTE BESCHREIBUNG DER ERFINDUNG

[0031] Das vorliegende Verfahren ist dazu vorgesehen, in Mehrzweckcomputern angewendet zu werden, die zur Ausführung der durch das Verfahren benötigten Hash- und Referenzfunktionen in der Lage sind. Der Computer benötigt auch ausreichend Speicherkapazität, um alle Zeichen der Original-Zeichenketten und die durch das Verfahren verwendeten Datenstrukturen zu speichern.

[0032] Das allgemeine Verfahren der Erfindung ist im Blockschaltbild von **Fig.** 1 gezeigt. Das Verfahren beginnt in Block **10** durch Auswählen einer Original-Zeichenkette **10** aus einer Datenbank. Dann wird die Zeichenkette in Block **15** in Teilzeichenketten mit zusammenhängenden Zeichen aufgeteilt, von denen mindestens zwei nicht zusammenhängend zusammengefügt sind, um in Block **20** Originaltupel zu bilden. Die Originaltupel dienen der Erzeugung von Originalindizes in Block **25**, die dann in Block **30** zur Speicherung von Daten in einer Zelle der Referenzstruktur dienen, welche zum Index und zur Original-Zeichenkette gehören. Diese Prozedur wird in Block **32** für jede zu untersuchende Original-Zeichenkette der Datenbank wiederholt. Auf ähnliche Weise werden aus der Referenz-Zeichenfolge die Referenzindizes erzeugt. Insbesondere wird die Referenz-Zeichenfolge in Block **35** in Teilzeichenfolgen von zusammenhängenden Zeichen aufgeteilt, von denen

mindestens zwei in Block **40** nicht zusammenhängend zusammengefügt werden, um Referenztupel zu bilden. Mittels desselben Verfahrens wie bei der Erzeugung von Originalindizes aus Originaltupeln werden in Block **45** Referenzindizes aus Referenztupeln erzeugt. Dann werden die Referenzindizes in Block **50** mit den Originalindizes verglichen. Die Anzahl der mit den Originalindizes übereinstimmenden Referenzindizes wird in Block **55** in einer Nachweis-Sammeltabelle (EIT) erfasst. Das Vergleichen und Erfassen kann so lange wiederholt werden, bis alle (oder ein Teil) der Referenzindizes mit den Originalindizes verglichen worden sind. Die Werte der gespeicherten Zahlen für die Übereinstimmung zwischen den Originalindizes und den Referenzindizes werden dann in Block **60** dazu verwendet, um diejenigen Original-Zeichenketten herauszufinden, die der Referenz-Zeichenkette am ähnlichsten sind. Weitere zum Originalindex gehörende Daten dienen in Block **65** zur Suche nach der Position derjenigen Zeichenfolge in der Original-Zeichenkette, welche fast oder ganz genau mit der Referenz-Zeichenfolge übereinstimmt. In Block **70** werden auf dieselbe Weise weitere Referenz-Zeichenketten verarbeitet und mit den Original-Zeichenketten verglichen.

[0033] Das Verfahren beginnt zunächst mit einer Datenbank mit Original-Zeichenketten X mit unterschiedlicher (oder gleicher) Zeichenanzahl. Üblicherweise sind diese Zeichenfolgen in einer zur Speicherung von Zeichenfolgen geeigneten Weise an einem Speicherplatz im Computerspeicher gespeichert. Zum Beispiel wird, wenn in der Datenbank eine große Anzahl von Proteinketten dargestellt werden soll, jedem der möglichen Aminosäurezeichen im Protein ein alphanumerisches Zeichen zugewiesen. Bei diesen Zeichen kann es sich um ASCII-Zeichen handeln. Jede Proteinkette wird dann durch eine ASCII-Zeichenkette in aufeinander folgenden Speicherplätzen des Computers dargestellt. Der Beginn einer Proteinkette kann durch einen Zeiger angezeigt werden, der auf das erste ASCII-Zeichen der Zeichenkette zeigt, und das Ende der Proteinkette kann durch einen Begrenzer wie etwa das Zeichen „0" angezeigt werden. Allgemein wird die Menge X der Original-Zeichenketten in der Datenbank durch $X \equiv \{\chi_i; i = 1, ..., N_\chi\}$ dargestellt.

[0034] Anschließend wird jede Original-Zeichenkette χ in der Datenbank X in zwei oder mehrere Original-Teilzeichenketten mit zusammenhängenden Zeichen μ aufgeteilt. Zur Veranschaulichung zeigt **Fig.** 2 eine Original-Zeichenkette χ **200**, in der jedes Zeichen **205** durch τ bezeichnet wird. Das Zeichen τ_i **207** bezeichnet das Zeichen τ an der Position i **220** der Original-Zeichenkette χ **200**. **Fig.** 2 zeigt ferner, dass durch

$$\mu^{(p_1, l)}$$

eine Original-Teilzeichenkette zusammenhängender und aufeinander folgender Zeichen **210** aus der Original-Zeichenkette bezeichnet wird, die bei dem Zeichen an der Position **220** (p_i) beginnt und eine Länge von l Zeichen **225** hat. Durch Aufteilen der Original-Zeichenkette wird eine Gruppe von Original-Teilzeichenketten **210** gebildet. Diese wird durch

$$M \equiv \mu^{(p_1, l_1)}, \mu^{(p_2, l_2)}, \ldots, \mu^{(p_k, l_k)}$$

dargestellt. Jede Original-Teilzeichenkette beginnt an einer Position p_k der Original-Zeichenkette und hat jeweils eine Länge l_k. Zum Beispiel ist die Original-Teilzeichenkette $\mu^{(5,14)}$ eine Teilzeichenkette aus der Original-Zeichenkette, die an der fünften Position der Original-Zeichenkette beginnt und die folgenden 13 Zeichen der Original-Zeichenkette enthält, wodurch die Teilzeichenkette eine Länge von 14 Zeichen hat. In manchen Fällen ist die Länge l_k einiger oder aller Original-Teilzeichenketten gleich der Länge anderer Original-Teilzeichenketten in dieser Gruppe. Eine Original-Teilzeichenkette muss eine Länge von mindestens einem Zeichen haben.

[0035] Eine Gruppe von K jeweils mit ξ_k bezeichneten Originaltupeln wird durch Zusammenfügen von j Original-Teilzeichenketten gebildet, wobei j gleich zwei oder größer ist. Mindestens eines der Tupel in dieser Gruppe wird durch Zusammenfügen von mindestens zwei nicht zusammenhängenden Original-Teilzeichenketten miteinander gebildet. An dieses Tupel können weitere zusammenhängende Teilzeichenketten angehängt werden. Andere Tupel in dieser Gruppe können auch durch Zusammenfügen zusammenhängender Teilzeichenketten miteinander gebildet werden (siehe Block **20** in **Fig.** 1). Dabei ist zu beachten, dass zwei Zeichenfolgen als zusammenhängend anzusehen sind, wenn das erste Zeichen der zweiten Teilzeichenkette auf das letzte Zeichen der ersten Teilzeichenkette in der Original-Zeichenkette folgt.

[0036] Ein durch eine Anzahl j von Original-Teilzeichenketten gebildetes Originaltupel wird als Original-j-Tupel bezeichnet. Ein j-Tupel der Länge L wird durch das Symbol $\xi^{(j,L)}$ dargestellt. Die Original-Teilzeichenketten im j-Tupel werden hier durch

$$\mu^{(p_1, l_1)}, \mu^{(p_2, l_2)}, \ldots, \mu^{(p_j, l_j)}$$

beschrieben. Man beachte, dass die Länge der Original-Teilzeichenketten in manchen Fällen gleich sein kann.

[0037] Tupel können mittels einer Vielzahl von Algorithmen durch Aneinanderreihung von Teilzeichenketten gebildet werden, diese Algorithmen bilden jedoch im Wesentlichen zwei Gruppen, nämlich probabilistische und deterministische Algorithmen. Die mittels probabilistischer Algorithmen erzeugten Tupel unterliegen eher dem Zufall, d. h. bei Anwendung desselben Algorithmus auf dieselbe Gruppe von Original-Teilzeichenketten erhält man wahrscheinlich jedes Mal einen anderen Satz von Tupeln. Ein deterministischer Algorithmus hingegen erzeugt jedes Mal aus einer bestimmten Gruppe von Original-Teilzeichenketten denselben Satz von Tupeln.

[0038] Zum Beispiel wird ein probabilistischer Algorithmus dazu eingesetzt, um aus einer Gruppe von 17 Teilzeichenketten mit einer Länge von je zwei Zeichen einen Satz von Dreiertupeln (ein Tupel wird durch Zusammenfügen von 3 Teilzeichenketten gebildet) zu erzeugen. Der Algorithmus wählt zufällig eine der 17 Teilzeichenketten aus, dann eine der 16 restlichen Teilzeichenketten, dann eine der 15 restlichen Teilzeichenketten und fügt die drei ausgewählten Teilzeichenketten zusammen, um das erste Dreiertupel des Satzes zu bilden. Die Prozedur wird dann beliebig oft wiederholt, z. B. 100 Mal, bis 100 Dreiertupel gebildet sind. Sehr wahrscheinlich unterscheidet sich der erste Satz von 100 Dreiertupeln, die auf diese Weise aus einer Original-Zeichenkette (Teilzeichenketten) gebildet wurden, vom nächsten Satz von 100 Dreiertupeln, die wiederum mittels desselben Algorithmus aus derselben Original-Teilzeichenkette gebildet wurden. Ungeachtet dessen bildet diese Art von Algorithmus einen Satz von j-Tupeln, die durch die vorliegende Erfindung verwendet werden können.

[0039] Beim vorliegenden Verfahren stellt der deterministische Algorithmus den bevorzugten Algorithmus zur Erzeugung von Tupeln dar. Zum Beispiel wird ein deterministischer Algorithmus zur Erzeugung von Dreiertupeln aus 17 Original-Teilzeichenketten der Länge 2 verwendet, indem er die am 1., 5. und 9. Zeichen der Original-Zeichenkette beginnenden Teilzeichenketten an die am 3., 7, und 11. Zeichen der Original-Zeichenkette beginnenden Teilzeichenketten anfügt. Dieser einfache Algorithmus erzeugt einen Satz von Dreiertupeln, die jedes Mal wieder entstehen, wenn der Algorithmus auf dieselbe Original-Zeichenkette und dieselben Original-Teilzeichenketten angewendet wird. Anhand der vorliegenden Beschreibung kann ein Fachmann eine große Anzahl deterministischer Algorithmen dieser Art entwickeln, die verschieden lange Teilzeichenketten, unterschiedlich viele (zwei oder mehr) zu einem Tupel verknüpfte Teilzeichenketten und verschiedene Verfahren zur Verknüpfung der Teilzeichenketten miteinander verwenden. Die Verwendung dieser vielen Varianten wird in der vorliegenden Erfindung dargelegt.

[0040] Es gibt jedoch einen besonders bevorzugten deterministischen Algorithmus zur Erzeugung von Tupeln für die vorliegende Erfindung, der im Folgenden beschrieben wird. Dieser besonders bevorzugte Algorithmus, der sich durch die begrenzte Anzahl von Tupeln zur Erreichung eines bestimmten Genauigkeitsgrades auszeichnet, wird im Folgenden allgemein beschrieben:

[0041] Für jede Zeichenfolge der Länge L_S, jedes gewünschte j-Tupel der Länge L und jede Ordnung j des Tupels ist eine Reihe von ganzen Zahlen, d. h. Teilzeichenkettenlängen (l_1, l_2, ..., l_j), so zu ermitteln, dass $\Sigma l_m = L$ (mit m = 1 bis j) ist.

[0042] Dann sind alle j-Tupel eines Satzes k zu bilden, der beschrieben wird durch

$$\xi_{k(j,L)} = \mu(p_1, l_1), + \mu(p_2, l_2), + \ldots + \mu(p_j, l_j),$$

wobei für jedes Glied k aus dem Satz von j-Tupeln die ersten Zeichenpositionen (p_1, p_2, ..., p_j) der durch Aufteilung aus der Original-Zeichenkette gebildeten und das Tupel bildenden Teilzeichenkette so ausgewählt werden, dass sie den folgenden Regeln genügen:

1. $\forall$ a und b, sodass $1 \le a < b \le j$ und $p_a < p_b$
2. $\forall$ a, sodass $1 \le a \le j$ und $p_a + l_a \le j$
3. $\forall$ a und b, sodass $1 \le b \le j$ und $\lambda^-_{ab} < p_b - p_a < \lambda^+_{ab}$,

wobei λ^-_{ab} und λ^+_{ab} eine Anzahl a priori festgelegter unterer und oberer Schwellenwerte sind.

[0043] Die bevorzugten Regeln zur deterministischen Erzeugung von Tupeln können alternativ wie folgt beschrieben werden:

1. Gleichung 1 besagt, dass für die Startposition p_a einer der das Tupel bildenden Zeichenketten die das j-te Tupel bildenden nachfolgenden Teilzeichenketten nur eine Startposition größer als p_a haben können, d. h. dass die nachfolgenden Teilzeichenketten des Tupels vom Beginn der Original-Zeichenkette an gerechnet später beginnen müssen. Hierdurch werden die Tupel in ihrer Reihenfolge erzeugt.
2. Gleichung 2 besagt, keine Teilzeichenkette an einer Position beginnen kann, von der aus die Teilzeichenkette über die Gesamtlänge der Original-Zeichenkette hinausragen würde.

3. Gleichung 3 besagt, dass sich beim Vorliegen zweier a priori festgelegter Schwellenwerte λ^-_{ab} und λ^+_{ab}, die als minimaler und maximaler Kohärenzradius bezeichnet werden, die Startpositionen p_a und p_b zweier aufeinander folgender Teilzeichenketten nicht um mehr als λ^+_{ab} und um weniger als λ^-_{ab} unterscheiden können. Gleichung 3 hat die größte Bedeutung für die Begrenzung der Anzahl der durch diesen Algorithmus erzeugten Tupel.

[0044] Dieser besonders bevorzugte deterministische Algorithmus wird durch ein Beispiel veranschaulicht. Eine Original-Zeichenkette hat eine Länge von 18 Zeichen, d. h. $L_s = 18$. Aus einem Satz von 17 Original-Teilzeichenketten mit einer Länge von je 2 Zeichen wird ein Satz von Dreiertupeln gebildet. Alle Tupel des erzeugten Satzes von Originaltupeln haben eine Länge von 6 Zeichen (L = 6) und werden durch Zusammenfügen von 3 Teilzeichenketten gebildet (j = 3). Man beachte, dass es 17 Teilzeichenketten gibt, da die Gruppe der Teilzeichenketten so gebildet wird, dass die erste Teilzeichenkette beginnend an der Zeichenposition p = 1, die zweite Teilzeichenkette beginnend bei p = 2 usw. erzeugt wird, bis die letzte Teilzeichenkette beginnend bei p = 17 erzeugt wird. Mit anderen Worten, die Gruppe M der Original-Teilzeichenketten besteht aus den folgenden 17 Teilzeichenketten: $\mu^{(1,2)}$ $\mu^{(2,2)}$, ..., $\mu^{(17,2)}$. Legt man alle möglichen geordneten Kombinationen von 3 zusammenhängenden bzw. nicht zusammenhängenden Teilzeichenketten diesem Satz von 17 Teilzeichenketten zugrunde, kann man 680 Dreiertupel erzeugen. Mit anderen Worten,

$$\binom{17}{3} = 17!/((17 - 3)!3!) = 680$$

[0045] Wenn man jedoch die drei obigen Regeln anwendet, wird durch Definieren der Kohärenzradien zu $\lambda^-_{12} = \lambda^-_{23} = 0$, $\lambda^-_{12} = \lambda^-_{23} = 3$, $\lambda^-_{13} = 0$ und $\lambda^-_{13} = 20$ ein deterministischer Satz von 42 Dreiertupeln erzeugt. Wendet man die Kriterien dieser 3 Regeln an, wäre ein Dreiertupel $\xi^{(3,6)}_{gut} = \mu^{(5,2)} + \mu^{(7,2)} + \mu^{(8,2)}$ zugelassen, wohingegen das Tupel $\xi^{(3,6)}_{schlecht} = \mu^{(5,2)} + \mu^{(9,2)} + \mu^{(10,2)}$ nicht zugelassen wäre, schlecht da $p_2 - p_1 = 4 \leq \lambda_{12} = 3$ ist. Durch geeignete Auswahl der Werte für die Kohärenzradien kann die Gesamtzahl der erzeugten Tupel und Indizes nach Belieben gewählt werden.

[0046] Im nächsten Schritt, der in Kasten **25** von **Fig.** 1 gezeigt ist, werden für jedes der aus einer Original-Zeichenkette gebildete Tupel eindeutige Originalindizes erzeugt. Gemäß der Beschreibung der vorliegenden Erfindung kann ein erfahrener Programmierer eine Vielzahl von Verfahren entwickeln, die man zur Erzeugung von Originalindizes für jedes dieser Originaltupel einsetzen kann. Bei diesen Verfahren werden üblicherweise Zuordnungsverfahren, Hash-Tabellen oder Algorithmen zum Umwandeln jedes Originaltupels in einen Originalindex verwendet, der eindeutig anzeigt, von welchem Tupel er stammt.

[0047] Bei der bevorzugten Ausführungsart werden eindeutige Indizes γ dadurch erzeugt, indem man zuerst jedem möglichen Zeichenwert in der Original-Zeichenkette einen Wert, z. B. einen numerischen Wert, zuweist. Dadurch entsteht aus der Original-Zeichenkette und somit auch aus den Teilzeichenketten und den Tupeln eine Folge numerischer Werte, welche ihre jeweiligen Zeichen darstellen. Im Allgemeinen wird für jedes Zeichen τ eine eindeutige Zahl τ_j zwischen 0 und n_τ ($0 \leq \tau_j \leq n_\tau$) gewählt. Zum Beispiel werden in einer aus Nukleotiden bestehenden DNA-Sequenz den vier möglichen Nukleotidzeichen (A, C, G, T) die numerischen Werte 0, 1, 2 bzw. 3 zugewiesen. Bei einem anderen Beispiel einer Aminosäuresequenz in einem Protein werden den 20 möglichen Aminosäurezeichen die numerischen Werte 0 bis 19 zugewiesen.

[0048] Dann wird ein Algorithmus zur Indexerzeugung festgelegt, der die Folge numerischer werte (Zeichen) in einen eindeutigen Index umwandelt. Als Beispiel wird ein Algorithmus zur Indexerzeugung vorgestellt, der nicht als Einschränkung anzusehen ist. Anhand der vorliegenden Erfindung kann ein Fachmann viele andere Algorithmen entwickeln. Ein bevorzugter zur Indexerzeugung eingesetzter Algorithmus ist:

$\gamma = \Sigma n_\tau^{(i-1)} \tau_i$, (mit i = 1 bis L).

[0049] Bei dem DNA-Beispiel nimmt der Algorithmus zur Indexerzeugung die Form $\gamma = \Sigma 4_\tau^{(i-1)} \tau_i$ an (mit i = 1 bis L).

[0050] Beim Proteinbeispiel nimmt der Algorithmus zur Indexerzeugung die Form $\gamma = \Sigma 20_\tau^{(i-1)} \tau_i$ an (mit i = 1 bis L).

[0051] Beim DNA-Beispiel würde dann ein j-Tupel wie „AATCGT" in die Zahlenfolge „003123" umgesetzt und den eindeutigen Index $4^0 \times 0 + 4^1 \times 0 + 4^2 \times 3 + 4^3 \times 1 + 4^4 \times 2 + 4^5 \times 3 = 3696$ haben.

[0052] Durch den Algorithmus zur Indexerzeugung wird für jedes Originaltupel im Satz der Originaltupel, die aus den Original-Teilzeichenketten (siehe Kasten **20** in **Fig.** 1) des ausgewählten zu bearbeitenden Originaltupels $\xi_k^{(j,L)}$ gebildet wurden, ein eindeutiger Originalindex erzeugt. Mit demselben Algorithmus und demselben Verfahren werden eindeutige Originalindizes erzeugt, die zu den Originaltupeln jeder anderen ausgewählten Original-Teilzeichenkette in der Datenbank gehören.

[0053] Kasten **30** in **Fig.** 1 beschreibt, wie die zu jedem Originaltupel gehörenden Daten im Computerspeicher (üblicherweise in einem Massenspeicher wie zum Beispiel einer Festplatte) gespeichert werden, damit später mittels des zu dem Tupel gehörenden eindeutigen Originalindexes darauf zugegriffen werden kann.

[0054] **Fig.** 3 zeigt eine Matrix einer in der bevorzugten Ausführungsart verwendeten ersten Daten-Referenzstruktur **300**, in der die zu jedem Originaltupel gehörenden Daten gespeichert werden. In der Technik sind auch andere Indexverfahren zur Speicherung von Daten für den späteren Zugriff bekannt und werden in die Erfindung einbezogen. Hierzu gehören Vektoren mit Zeigern, die auf Listen von Datensätzen zeigen.

[0055] Die in **Fig.** 3 gezeigte Daten-Referenzstruktur **300** besteht aus einer Matrix mit einer Vielzahl üblicher Zellen **310**. Die Matrix der Daten-Referenzstruktur **300** hat mindestens n^L Zellen, wobei L die Länge des längsten verwendeten Tupels und $n = n_t$ die Anzahl der möglichen Zeichenwerte in den Original-Zeichenketten ist. (Im Falle der DNA ist für numerische Darstellungen von vier möglichen Nukleotiden n = 4.) Dadurch wird sichergestellt, dass jedem erzeugten Originalindex mindestens eine Zelle eindeutig zugeordnet ist. Man beachte, dass es bei der vorliegenden Ausführungsart auch viele Zellen gibt, die zu keinem erzeugten Originalindex gehören. Wenn einem erzeugten Index eine Zelle zugeordnet wurde, wird ein Datensatz **314**, üblicherweise ρ, in die Zelle eingefügt oder an sie angehängt, welcher die Original-Zeichenkette und das Tupel charakterisiert, aus denen der Index **312**, üblicherweise γ, erzeugt wurde. Eine Zelle **310** kann mehrere Datensätze **314** und **326** enthalten. Dieser Datensatz enthält zumindest einen Zeiger **315**, der auf die Original-Zeichenkette in der Datenbank zeigt, aus welcher das Tupel erzeugt wurde. Vorzugsweise enthält der Datensatz außerdem auch noch Daten **320** zur Position des Originaltupels in der Original-Zeichenkette. Vorzugsweise enthält die Zelle außerdem auch noch weitere Daten über das Tupel und die Position der zur Erzeugung des Tupels aneinander gehängten Teilzeichenketten. (Nähere Beschreibung siehe unten.) Bei der bevorzugten Ausführungsart bleiben Zellen, die zu keinem erzeugten Index gehören, leer.

[0056] Bei einer besonders bevorzugten Ausführungsart enthält die Zelle **310** außerdem auch noch Platz für mehr als einen Datensatz über mehr als eine Original-Zeichenkette, die identische erzeugte Indizes haben kann. Mit anderen Worten, die Zelle enthält eine Liste **328** mit mehreren Datensätzen, die jeweils einen Zeiger enthalten (**315** und **325**). Jeder Datensatz **326** in der Zellenliste **328**, der einen Zeiger **325** enthält, kann außerdem noch Positionsdaten **330** für den Zeiger **325** enthalten. Wenn mehr als eine Original-Zeichenkette denselben Index erzeugt, werden die Daten für jede den identischen Index **312** erzeugende Original-Zeichenkette an die Liste **328** der Datensätze in der Einzelzelle **310** angehängt, welche dem gemeinsamen Index **312** zugeordnet ist. Die Liste **328** kann statisch oder dynamisch sein. Eine statische Liste enthält Speicherplatz zur Speicherung von Daten, die sich auf eine fest vorgegebene Anzahl von Original-Zeichenketten beziehen, und lässt alle Daten verloren gehen, die den reservierten Speicherplatz überschreiten. Eine dynamische Liste wird erweitert, um Daten über jede Original-Zeichenkette zu speichern, die einen gemeinsamen Index erzeugt.

[0057] Obwohl die Referenzstruktur **300** vom Matrixtyp für eine kleine Anzahl möglicher Indizes ausreicht, wird bei Anwendungen mit einer großen Anzahl von Indizes eine große Anzahl von Zellen **310** der Referenzstruktur benötigt. Ferner beinhalten die meisten dieser Zellen **310** bei vielen Anwendungen keine Daten, da sie keinem erzeugten Index **312** zugeordnet sind. In diesen Fällen werden in der Technik bekannte Verfahren für den Umgang mit leeren Matrizen eingesetzt. Zu diesen Verfahren gehören Hash-Tabellen.

[0058] In der Technik sind viele Ausführungsarten mit Hash-Tabellen bekannt, die in der vorliegenden Erfindung inbegriffen sind. **Fig.** 4 zeigt jedoch als Beispiel eine erste Referenzstruktur, die als bestimmter Typ einer Hash-Tabelle realisiert wurde. Dieses Beispiel beschreibt eine relativ kleine Matrix von n Zellen 0 bis n. Jeder erzeugte Index wird mit einer Primzahl wie zum Beispiel 7 multipliziert und aus diesem Wert durch Modulo n der Wert **405** erhalten. Dann wird der erzeugte Hash-Index **412** der Zelle **410** zugewiesen, deren Identifizierungszahl mit dem Wert der Modulo-Operation übereinstimmt. Der zu diesem Index gehörende Datensatz **414** wird dann in der gewählten Zelle gespeichert. Das Verfahren geht davon aus, dass es in der Hashtabelle genügend Zellen **410** gibt, die jedem erzeugten Index eindeutig zugewiesen werden können. Wenn ein erzeugter Index einer Zelle zugeordnet wird, die bereits einem anderen davon verschiedenen Index zugewiesen wurde, wird dieser Index erneut gehasht und neu zugeordnet. Zum Beispiel wird der Modulo-Wert mit 7 multipliziert und erneut der Modulo-Operation unterzogen. Als Ergebnis soll eine Zahl entstehen, die eine andere Zelle

kennzeichnet. Dies wird so lange wiederholt, bis eine leere Zelle gefunden wurde.

[0059] Im Folgenden werden die in der Zelle der ersten Referenzstruktur gespeicherten Daten beschrieben. Der Datensatz ρ dient dazu, auf einen bestimmten Speicherplatz in einer zweiten Speicherstruktur zuzugreifen, die als Nachweis-Sammeltabelle (Evidence Integration Table, EIT) bezeichnet wird. Ein Datensatz enthält mindestens einen auf die Original-Zeichenkette zeigenden Zeiger α **415** zur Berechnung des Indexes der Zelle **412**. Die bevorzugte Ausführungsart speichert einen Zeiger **415** und einen Verschiebungswert **420**. Der Verschiebungswert δ **420** liefert Daten zur Position der Original-Teilzeichenketten in der Original-Zeichenkette, welche ein der Zelle der Referenzstruktur zugeordnetes j-Tupel gebildet hat.

[0060] Bei Bedarf können in den Zellen auch noch weitere Daten gespeichert werden. Das können andere Datensätze **426** sein, die wiederum Daten zu weiteren Zeigern **425** und Verschiebungswerten **430** enthalten. Diese Datensätze **426** können je nach Bedarf dynamisch in die Zelle eingefügt oder an diese angehängt werden. Es können aber auch noch weitere Daten hinzugefügt werden. Zum Beispiel kann die Zelle **410** Daten zum Abstand zwischen den ein j-Tupel bildenden Teilzeichenketten enthalten.

[0061] Verweise/Zeiger sind in der Rechentechnik bekannt, und in der vorliegenden Erfindung sind alle Zeiger inbegriffen, die an einem Speicherplatz, wie zum Beispiel in der Zelle einer Referenzstruktur, gespeichert und zum Auffinden einer in diesem Speicher gespeicherten Zeichenfolge, die eine Original-Zeichenkette darstellt, verwendet werden können. Ein Verweis/Zeiger kann die Adresse des Speicherplatzes sein, der dasjenige erste (oder ein anderes) Zeichen der Original-Zeichenkette enthält, auf welches gezeigt wird. Alternativ kann der bevorzugte Verweis/Zeiger lediglich eine (ganze) Zahl sein, die einem Index einer Original-Zeichenkette in einer Datenbank von (ganzzahligen) nummerierten Zeichenketten entspricht.

[0062] Der Verschiebungswert beschreibt die mittlere oder genaue Position einer Original-Zeichenkette, in welcher sich die Teilzeichenketten befinden, die zur Erstellung eines Tupels dienten. Zur Abschätzung der Position dieser Teilzeichenketten gibt es viele Wege. Zum Beispiel kann dies ein Verschiebungswert δ_k für ein von der Original-Zeichenkette χ_i abgeleitetes Tupel $\xi_k^{(j,L)}$ sein. Dieser Verschiebungswert δ_i kann der in Zeichen gemessene Abstand (Differenz) zwischen einem bestimmten Zeichen, wie zum Beispiel dem ersten Zeichen τ_i der Original-Zeichenkette χ_i, und einem anderen bestimmten Zeichen, wie zum Beispiel dem ersten (oder zweiten, dritten usw.) Zeichen der ersten Teilzeichenkette, sein, welche zur Erzeugung des Tupels (Index) verwendet wurde, das durch die Zelle in der Speicherstruktur eindeutig gekennzeichnet wird. Bei der besonders bevorzugten Ausführungsart wird der in Zeichen gemessene Mittelwert der Abstände zwischen einem Anfang der Original-Zeichenkette und dem Anfang jeder der Teilzeichenketten zur Erzeugung des Tupels verwendet. Dieser Mittelwert wird als δ_{kmitt} bezeichnet. Dann hat ein j-Tupel einen Mittelwert δ_{kmitt}, der durch Mittelung der in Zeichen gemessenen Abstände zwischen dem ersten Zeichen der Original-Zeichenkette und dem ersten Zeichen in jeder der j Teilzeichenketten berechnet wird.

[0063] Nachdem die Originaltupel und die zugehörigen Originalindizes für die betreffenden Original-Zeichenketten erzeugt und die Datensätze für jeden erzeugten Originalindex in seiner entsprechenden Zelle in der ersten Referenzstruktur gespeichert wurden, werden Referenztupel und deren zugehörige eindeutige Referenzindizes erzeugt. Hierzu siehe Kasten **35**, **40** und **45** in **Fig.** 1. Dies wird erreicht durch: 1. Aufteilen einer bestimmten Referenz-Zeichenkette in zwei oder mehr Referenz-Teilzeichenketten aus zusammenhängenden Zeichen, 2. Bilden mindestens eines Referenztupels durch Zusammenfügen von mindestens zwei nicht zusammenhängenden Referenz-Teilzeichenketten und 3. Erzeugen eines Referenzindexes mittels desselben Indexgenerators, der zur Erzeugung der Indizes in der Referenzstruktur verwendet wurde.

[0064] Eine Referenz-Zeichenkette χ_{ref} ist eine Zeichenfolge, die mit den Original-Zeichenketten in der Datenbank verglichen wird, um festzustellen, ob die Referenz-Zeichenkette genau oder ungefähr mit einem Teil (einer Teilzeichenfolge) einer oder mehrerer Original-Zeichenketten in der Datenbank übereinstimmt.

[0065] Um gemäß Kasten **35** in **Fig.** 1 die Referenztupel ξ_{ref}, d. h. die aus dieser Referenz-Zeichenkette gebildeten Tupel, zu bilden, wird die Referenz-Zeichenkette zuerst in Teilzeichenketten aus zusammenhängenden Zeichen aufgeteilt, die als Referenz-Teilzeichenketten bezeichnet werden. Diese Aufteilung erfolgt nach einem der oben bei der Aufgliederung in Original-Teilzeichenketten erörterten Verfahren. Man beachte, dass die Aufgliederung von Referenz-Zeichenketten in Teilzeichenketten auch mit vielen anderen Verfahren als bei der Aufgliederung der Original-Zeichenketten in Teilzeichenketten erfolgen kann. Bei der bevorzugten Ausführungsart jedoch werden sowohl die Original-Zeichenketten als auch die Referenz-Zeichenketten nach demselben Verfahren aufgegliedert. Hierzu siehe Kasten **40** in **Fig.** 1. Die Referenztupel ξ_{ref} werden nun durch Zusammenfügen von mindestens zwei nicht zusammenhängenden Referenz-Teilzeichenketten gebildet. Dieses Zu-

sammenfügen von Referenz-Teilzeichenketten erfolgt nach einem der oben für das Zusammenfügen von Original-Teilzeichenketten zur Bildung von Originaltupeln erörterten Verfahren. Die Referenz-Teilzeichenketten brauchen jedoch nicht in genau derselben Weise zusammengefügt werden wie die Original-Teilzeichenketten zu Originaltupeln.

[0066] Nun werden gemäß Kasten **45** in **Fig.** 1 aus den Referenztupeln die Referenzindizes γ_{ref} gebildet. Wie zuvor erhält jedes Referenztupel einen eindeutigen Referenzindex. Der eindeutige Referenzindex für jedes Referenztupel sollte möglichst mittels genau desselben Algorithmus zur Indexerzeugung erzeugt werden wie bei der oben beschriebenen Erzeugung der Originalindizes aus den Originaltupeln.

[0067] Auch der Referenz-Verschiebungswert Δ wird berechnet, um die Position der Referenz-Teilzeichenketten in der Referenz-Zeichenkette zu ermitteln, die zur Erzeugung des Referenztupels verwendet wurden. Zur Erzeugung des Positionswertes für die Referenz-Teilzeichenketten kann jedes allgemeine Verfahren, wie zum Beispiel die oben zur Erzeugung des Verschiebungswertes für Original-Zeichenketten eingesetzten Verfahren, verwendet werden. Auch hier sollte das zur Ermittlung des Verschiebungswertes für die Referenztupel verwendete Verfahren möglichst genau dasselbe Verfahren sein, welches zur Ermittlung des Verschiebungswertes für Originaltupel verwendet wurde.

[0068] Nachdem die Referenzindizes erzeugt worden sind, werden sie mit den Originalindizes verglichen, um Übereinstimmungen festzustellen (siehe Kasten **50** in **Fig.** 1). In der vorliegenden Erfindung sind alle in der Technik bekannten Verfahren zum Vergleichen von zwei Indizes (Zahlen) oder von zwei Indexlisten (Zahlenlisten) inbegriffen, mit denen eine Übereinstimmung zwischen den Indizes festgestellt werden kann. Als bevorzugtes Verfahren zum Vergleichen der Indizes mit den Originalindizes wird jedoch die Referenzstruktur verwendet. Hierzu wird jeder Referenzindex zum Zugriff auf diejenige Zelle (üblicherweise die Zelle **310** in **Fig.** 3 oder Zelle **410** in **Fig.** 4) in der Referenzstruktur verwendet, die einem Originalindex (üblicherweise **312** oder **412**) zugeordnet ist, der gleich dem Referenzindex ist. Liegen in der Zelle, auf die zugegriffen wurde, ein oder mehrere Datensätze (**328** oder **428**) vor, kommt es zur Übereinstimmung zwischen dem Referenzindex (**350** oder **450**) und einem oder mehreren Originalindizes (üblicherweise **312** oder **412**) (Tupel), welche der Zelle, auf die zugegriffen wurde (üblicherweise **310** oder **410**), zugeordnet sind. Befindet sich in der Zelle kein Datensatz, kommt es nicht zur Übereinstimmung und für dieses Referenztupel erfolgt keine weitere Verarbeitung.

[0069] In Kasten **55** in **Fig.** 1 wird in einem zweiten Speicherbereich mit der Bezeichnung Nachweis-Sammeltabelle EIT (Evidence Integration Table) jede Übereinstimmung zwischen einem Referenzindex und einem Originalindex erfasst. In **Fig.** 5 wird auf die als Zählzellen bezeichneten Zellen **510** in der EIT **500** mittels eines Zählindexes **512** zugegriffen, der anhand eines Teils oder des gesamten Datensatzes (**314** oder **414**) in der Zelle (**310** oder **410**) der Referenzstruktur erzeugt wird, welche dem mit dem Originalindex übereinstimmenden Referenzindex zugeordnet ist. Im Allgemeinen braucht nur der Verweis/Zeiger (**315** oder **415**) im Datensatz als Zählindex oder zu dessen Berechnung verwendet zu werden. Vorzugsweise werden insbesondere der Verweis/Zeiger und der Verschiebungswert (**320** oder **420**) des Datensatzes und der Verschiebungswert des Referenztupels Δ zur Berechnung eines Zählindexes **512** verwendet. Besonders bevorzugt zur Erzeugung eines Zählindexes **512** für den Zugriff auf eine Zelle **510** in der Zähltabelle **500** ist die Verwendung des Verweises/Zeigers (**315** oder **415**) und eine Optimierung der Differenz zwischen dem im Datensatz (**320** oder **420**) gefundenen Verschiebungswert sowie dem Verschiebungswert des Referenztupels Δ.

[0070] Der Zählindex **512** der bevorzugten Ausführungsart wird aus dem Verweis/Zeiger (**315** oder **415**) α und der Vergleichsdifferenz (berechnet als Differenz zwischen dem Verschiebungswert δ des Datensatzes ρ minus dem Referenz-Verschiebungswert Δ) berechnet. Die Vergleichsdifferenz D_0 entspricht dem in Zeichen gemessenen Abstand zwischen einem bestimmten Zeichen in der Original-Zeichenkette, z. B. dem ersten Zeichen der Original-Zeichenkette, und einem bestimmten Zeichen der Referenz-Zeichenkette, z. B. dem ersten Zeichen der Referenz-Zeichenkette, sodass die übereinstimmenden Zeichenfolgen in der Original-Zeichenkette und der Referenz-Zeichenkette zur Deckung gebracht werden, wenn die Zeichenketten um den Abstand der Vergleichsdifferenz verschoben werden. Mit anderen Worten, die Vergleichsdifferenz (durch $D_0 = \delta_{mitt} - \Delta_{kmitt}$ berechnet) stellt den in Zeichen gemessenen Abstand dar, um den die Original-Zeichenkette gegenüber der Referenz-Zeichenkette verschoben werden muss, um diejenigen übereinstimmenden Zeichenfolgen zur Deckung zu bringen, von denen die entsprechenden übereinstimmenden Originaltupel und Referenztupel (Indizes) abgeleitet wurden.

[0071] Nachdem die Vergleichsdifferenz D_0 ermittelt worden ist, wird mittels eines Algorithmus der Zählindex **512** erzeugt. Es gibt in der Technik viele Verfahren zur Erzeugung solcher Indizes, die in der Erfindung inbegriffen sind. Das bevorzugte Verfahren besteht jedoch darin, den Wert des Verweises/Zeigers **315** oder **415** α

im Datensatz (**328** oder **428**) mit einer bestimmten Konstanten zu multiplizieren, die gleich der größtmöglichen Vergleichsdifferenz ist, und dazu dann den Wert der Vergleichsdifferenz zu addieren. Die Konstante wird so gewählt, dass der Algorithmus einen eindeutigen Zählindex für jede mögliche Kombination von Zeiger und Vergleichsdifferenz erzeugt, die aus den Datensätzen in der Referenzstruktur erzeugt werden kann. Dabei ist zu beachten, dass man aus dem Zählindex die Werte für den Verweis/Zeiger und die Vergleichsdifferenz ermitteln kann, indem man den Algorithmus umkehrt. Jede Zelle in der EIT bezeichnet eindeutig eine Original-Zeichenkette mit einer bestimmten Vergleichsdifferenz gegenüber der Referenz-Zeichenkette.

[0072] Die Zählzellen **510** in der EIT **500**, auf die durch die Zählindizes zugegriffen wird, dienen jedes Mal der Speicherung der „Zählwerte" **515** für eine Original-Zeichenkette mit einer bestimmten Vergleichsdifferenz, wenn mittels der Referenzstruktur und des Referenzindexes in der oben beschriebenen Weise eine entsprechende Übereinstimmung festgestellt wird. Der Wert „c" **515** in jeder Zählzelle **510** der EIT **500** wird jedes Mal aktualisiert, wenn für diese Zelle ein Zählindex erzeugt wird. Kommt es zur Übereinstimmung, d. h. wenn eine Zelle in der Referenzstruktur mindestens einen Datensatz enthält, wird aus dem Verweis/Zeiger im Datensatz und der berechneten Vergleichsdifferenz ein Zählindex **512** erzeugt. Auf die dem Verweis/Zeiger und der Vergleichsdifferenz zugeordnete eindeutige Zählzelle wird durch den Zählindex zugegriffen und ihr Wert c um einen bestimmten Betrag, normalerweise um den ganzzahligen Wert 1, erhöht. Demzufolge stellt der Wert c **515** in jeder Zählzelle **510** einen direkten Hinweis darauf dar, wie oft eine Original-Zeichenkette und eine Referenz-Zeichenkette identische Indizes und denselben Wert der Vergleichsdifferenz erzeugt haben.

[0073] Daher zeigt bei der bevorzugten Ausführungsart der vorliegenden Erfindung eine Zelle in der EIT mit einem hohen Wert c an, dass die entsprechende Original-Zeichenkette der Referenz-Zeichenkette mit hoher Wahrscheinlichkeit ähnlich oder identisch ist, wenn sie um die Vergleichsdifferenz verschoben wird.

[0074] In der Technik sind viele Strukturen bekannt, die als Ausführungsarten für die EIT dienen können, z. B. eine Matrix, ein Vektor oder eine Hash-Tabelle. Matrizen und deren eindimensionale Untereinheit, also Vektoren, müssen bei großen Datenbanken, die die bevorzugte Ausführungsart verwenden, ziemlich groß sein, da für jede mögliche Kombination von Zeiger und Vergleichsdifferenz eine eindeutige Zählzelle benötigt wird. Bei einer typischen Anwendung der vorliegenden Erfindung wird auf die meisten dieser Zellen nicht zugegriffen, da es sehr unwahrscheinlich ist, dass alle diese Zählzellen **510** einer Übereinstimmung zwischen einem Originaltupel und einem Referenztupel (Index) zugeordnet sind. Infolgedessen wird die Matrix (der Vektor) nur in geringem Maße gefüllt und ein großer Teil der Speicherkapazität vergeudet. Die vorliegende Ausführungsart bedient sich einer Hash-Tabelle, um die Anforderungen an das Speichervolumen zu verringern. Hash-Tabellen-Verfahren sind in der Technik bekannt und wurden oben anhand von Beispielen erörtert. Die Hash-Tabelle kann statisch oder dynamisch sein. Eine statische Hash-Tabelle hat eine fest vorgegebene Anzahl von Zählzellen. Eine dynamische Hash-Tabelle hingegen beginnt ohne oder mit nur wenigen Zählzellen und legt beim Auftreten von Übereinstimmungen noch mehr Zählzellen fest und fügt sie zur Tabelle hinzu. Die bevorzugte Ausführungsart verwendet eine dynamische Hash-Tabelle.

[0075] In Kasten **60** in **Fig.** 1 werden alle Zellen in der EIT mit einem Wert c, der einen vorgegebenen Schwellenwert übersteigt, als Hinweis auf übereinstimmende Original-Zeichenketten ausgewählt. Dann kann eine dem Wert c direkt proportionale Ähnlichkeitsbewertung berechnet werden.

[0076] Durch Umkehrung des Wertes des Zählindexes der Zelle in Kasten **65** in **Fig.** 1 können wie oben erläutert der Verweis/Zeiger auf die Original-Zeichenketten und deren zugehörige Vergleichsdifferenzen genau ermittelt werden.

[0077] Zwischen einer Zeichenfolge in einer Original-Zeichenkette χ^0 und einer Zeichenfolge in einer Referenz-Zeichenkette χ^0_{ref} kann es grundsätzlich zu zwei unterschiedlichen Übereinstimmungen kommen, und zwar zu genauen und zu ungefähren Übereinstimmungen.

[0078] Zu genauen Übereinstimmungen kommt es, wenn jedes Zeichen einer Zeichenfolge in einer Referenz-Zeichenkette χ^0_{ref} denselben Wert und dieselbe Reihenfolge aufweist wie jedes Zeichen einer Zeichenfolge in einer Original-Zeichenkette χ^0. Bei einer genauen Übereinstimmung findet jedes Referenztupel (Index) mindestens einen Verweis/Zeiger auf die identische Original-Zeichenkette in der Zelle, welche es bezeichnet. Infolgedessen ist der Wert „c" in der der identischen Original-Zeichenkette entsprechenden Zählzelle sehr groß und mindestens gleich der Anzahl der anhand der übereinstimmenden Zeichen erzeugten Referenztupel.

[0079] Zu annähernden Übereinstimmungen kommt es, wenn die Original-Zeichenkette χ^0 durch Insertion, Deletion oder Änderung von nur wenigen Zeichen genau mit der Referenz-Zeichenkette in Übereinstimmung

gebracht werden kann. Bei annähernden Übereinstimmungen ist es unwahrscheinlich, dass jedes Referenztupel (Index) einen Verweis/Zeiger auf die ähnliche Original-Zeichenkette in der Zelle findet, welche es bezeichnet. Eine größere Ähnlichkeit zwischen der Original-Zeichenkette und der Referenz-Zeichenkette führt jedoch zu einer größeren Anzahl genauer Übereinstimmungen und folglich einer größeren Anzahl von Zählschritten für die Original-Zeichenkette in der EIT. Daher kann man einen Ähnlichkeitsgrad zwischen jeder Original-Zeichenkette und der Referenz-Zeichenkette ermitteln, wenn man die Anzahl der Zählschritte in der EIT vergleicht, welche den verschiedenen Original-Zeichenketten beim Vergleich mit einer vorgegebenen Referenz-Zeichenkette zugewiesen werden. Diejenigen Original-Zeichenketten, die in der EIT eine größere Anzahl von Zählschritten erhalten haben (nach dem Vergleich aller Referenztupel), sind der Referenz-Zeichenkette ähnlicher als die Original-Zeichenketten mit einer geringeren Anzahl von Zählschritten.

[0080] Schließlich wird in der Datenbank gemäß Kasten **60** und **65** in **Fig.** 1 nach den ermittelten genau und ungefähr übereinstimmenden Original-Zeichenketten gesucht. Hierzu werden diejenigen Zellen in der EIT ausgewählt, deren „c"-Werte einen bestimmten Schwellenwert übersteigen. Dann wird der Zählindex dieser ausgewählten Zellen umgekehrt, um die Position (oder andere Ortsbestimmung) der Original-Zeichenkette, welche diesen Zählwert verursacht hat, und den Verschiebungswert (falls vorhanden) zu ermitteln, mittels dessen die Referenz-Zeichenkette und die Original-Zeichenketten zur Deckung gebracht werden, um die übereinstimmenden Zeichenfolgen zur Deckung zu bringen. Ferner kann man anhand des Datensatzes in derjenigen Zelle der Referenztabelle, welche den Zählschritt verursacht hat, nach den mit der Original-Zeichenkette übereinstimmenden Zeichenfolgen suchen. Wenn mehrere Referenz-Zeichenketten verglichen werden müssen (siehe Kasten **70**), startet der Prozess wieder in Kasten **35**.

BEISPIEL 1

[0081] Im Folgenden wird zur Veranschaulichung ein Beispiel einer Zeichenkettensuche vorgestellt, welches das vorliegende Verfahren anwendet und keine Einschränkung der Erfindung bedeutet. Diese Veranschaulichung bezieht sich auf Anwendungen bei Zeichenketten-Erkennungsverfahren wie zum Beispiel in Wörterverzeichnissen und Rechtschreibprüfungen.

[0082] Die Original-Zeichenkette ist „HOTEL" und die Referenz-Zeichenkette „HOSTEL". Die Länge der für beide Zeichenketten gewählten Teilzeichenketten beträgt ein Zeichen und die Länge der Tupel drei Zeichen. Deshalb sind
die Original-Teilzeichenketten: H, O, T, E und L, und
die Referenz-Teilzeichenketten: H, O, S, T, E und L.

[0083] Der zur Erzeugung der Tupel (sowohl Originaltupel als auch Referenztupel) verwendete deterministische Algorithmus besteht in der Auswahl aller möglichen eindeutigen geordneten Kombinationen von drei Teilzeichenketten. Somit sind
die Originaltupel: HOT, HOE, HOL, HTE, HTL, HEL, OTE, OTL und TEL, und
die Referenztupel: HOS, HOT, HOE, HOL, HST, HSE, HSL, HTE, HTL, HEL, OST, OSE, OSL, OTE, OTL, STE, STL und TEL.

[0084] Aus Gründen der Vereinfachung werden aus den Tupeln keine numerischen Indizes gebildet. In den Zellen der Referenztabelle werden Datensätze abgelegt. Zur Vereinfachung enthält ein Datensatz nur einen Zeiger, der auf die Startposition der Original-Zeichenkette HOTEL in der Datenbank zeigt.

[0085] Aus dem Vergleich jedes Referenztupels (Indexes) mit den Originaltupeln ergeben sich 9 Übereinstimmungen. Diese Ergebnisse werden in der EIT gespeichert. In diesem Fall zeigen die 9 Übereinstimmungen einen hohen Korrelationsgrad. Eine Referenz-Zeichenkette wie zum Beispiel „SOLID" würde keine Übereinstimmungen ergeben.

[0086] Die Gesamtzahl von 9 Übereinstimmungen mit dem Wort HOTEL bedeutet eine große Ähnlichkeit. Man beachte, dass man bei Verwendung zusammenhängender Folgen von je drei Zeichen nur eine Übereinstimmung bei TEL erhielte. Jede anschließende Änderung eines Buchstabens (z. B. E zu A) würde im Falle zusammenhängender Zeichenfolgen gar keine und im Falle nicht zusammenhängender Zeichenfolgen drei Übereinstimmungen ergeben.

[0087] **Fig.** 6 und 7 sind Ablaufdiagramme des Computerprogramms, das durch die vorliegende Erfindung zum Erzeugen und Speichern von Originaltupeln (Indizes) und später zum Suchen derjenigen Original-Zeichenketten verwendet wird, welche diesen Tupeln (Indizes) zugeordnet sind, die mit einer Referenz-Zeichen-

kette übereinstimmen. **Fig.** 6 ist ein Ablaufdiagramm des Computerdiagramms, welches Originaltupel, Originalindizes und die zugehörigen Datensätze erzeugt. **Fig.** 7 ist ein Ablaufdiagramm des Computerprogramms, welches Referenztupel und Referenzindizes erzeugt, die Referenzindizes mit den Originalindizes vergleicht und die EIT aktualisiert.

[0088] In Kasten **610** in **Fig.** 6 befinden sich in einer Computer-Datenbank X eine oder mehrere Zeichenfolgen beliebiger Länge L, welche Original-Zeichenketten χ darstellen. Jeder Original-Zeichenkette ist ein Verweiswert α zugeordnet, der ein Zeiger oder der Index für einen Zeiger sein kann. In Kasten **615** werden im Speicher auch minimale und maximale Kohärenzradien gespeichert. In Kasten **620** startet die äußere Schleife einer Reihe verschachtelter Schleifen, die auf Basis der oben angegebenen Regeln aus einer Original-Zeichenkette Originaltupel erzeugt. Beim ersten Mal wird in Kasten **620** das erste Zeichen der betreffenden Original-Zeichenkette ausgewählt. In Kasten **625** wird die Regel 2 angewendet, um festzustellen, ob die Länge der hier startenden Teilzeichenkette die Länge der Original-Zeichenkette überschreitet. Wenn dies der Fall ist, wird das Programm für diese Original-Zeichenkette beendet. Wenn es nicht der Fall ist, geht das Programm weiter zu Kasten **630**, wo die Startposition der zweiten Teilzeichenkette ausgewählt wird. Diese Position startet vom Anfang der Zeichenkette aus gesehen ein Zeichen später als das vorhergehende Startzeichen, in diesem Fall also an Position 2. Die Regel 1 ist so lange eingehalten, wie nachfolgende Zeichenketten vom Anfang der Zeichenkette aus gesehen später als vorhergehende Zeichenketten starten. In diesem Kasten wird die zweite Ebene der verschachtelten Schleifen gestartet. In Kasten **635** wird die Regel 2 auf die zweite ausgewählte Zeichenkette angewendet. Sobald die Regel verletzt wird, springt die Verarbeitung wieder zu Kasten **620** oder zur äußeren verschachtelten Schleife zurück. Wenn die Regel eingehalten wird, erfolgt die Verarbeitung weiterhin in der zweiten Ebene der verschachtelten Schleifen und in Kasten **640** werden die Bedingungen der Regel 3 angewendet. Sobald die Regel 3 verletzt wird, wird für die zweite Teilzeichenkette (immer noch innerhalb der zweiten Ebene der verschachtelten Schleifen) eine andere Startposition gewählt und der Prozess beginnt von vorn. Der Prozess wird so lange in Kasten **645** und **650** wiederholt, bis die Startposition der letzten Original-Teilzeichenkette erreicht ist. Insbesondere bezeichnen Kasten **645** und **650** eine beliebige Anzahl von untergeordneten Schleifen, die zur Erzeugung aller möglichen Kombinationen von j-Tupeln dienen, welche die Regeln für eine bestimmte Anzahl j von Teilzeichenketten im j-Tupel erfüllen. Man beachte, dass bei Verwendung von lediglich zwei Teilzeichenketten, d. h. j = 2, Kasten **645** und **650** entfallen können. Sobald Tupel durch das Starten der letzten Teilzeichenkette in der Original-Zeichenkette erzeugt werden, geht die Prozedur weiter zu Kasten **655**. wenn die ausgewählte Zeichenkette in Kasten **660** die Regel 2 verletzt, geht die Verarbeitung über die nächstäußere Schleife an Kasten **650** über. Wenn über alle Schleifen das Ende der Zeichenkette erreicht wurde, geht die Verarbeitung so lange an die äußeren Schleifen über, bis die Ebene der äußersten Schleife in Kasten **620** erreicht ist, und verlässt dann in Kasten **625** das Programm. Bei der j-ten Teilzeichenkette werden die Bedingungen von Regel 2 und 3 ebenso wie oben beschrieben in Kasten **660** und **665** überprüft und bearbeitet. In Kasten **670** werden alle in den Schleifen ausgewählten j Teilzeichenketten, die die drei Regeln erfüllen, aneinander gefügt, um ein Tupel zu bilden. Der eindeutige Originalindex für das in Kasten **670** gebildete Tupel wird in Kasten **675** mittels der oben beschriebenen Verfahren erzeugt. In Kasten **680** wird wie oben beschrieben der Verschiebungswert für das j-Tupel und in Kasten **685** der Datensatz berechnet. In Kasten **690** wird die Liste der zu dem erzeugten j-Tupel gehörenden Datensätze aus der Referenztabelle abgerufen. Wenn der neue Datensatz dort noch nicht vorhanden ist, wird er an die Liste angehängt und die Liste in der Referenztabelle gespeichert. Die Verarbeitung wird in Kasten **655** fortgesetzt, um auf dieser Ebene der verschachtelten Schleifen das nächste Tupel zu erzeugen. Die gesamte Prozedur des Ablaufdiagramms von **Fig.** 6 wird für jede Original-Zeichenkette der Datenbank wiederholt.

[0089] In Kasten **710** in **Fig.** 7 wird in der Datenbank eine zu untersuchende Referenz-Zeichenkette gespeichert. In Kasten **715** werden eine EIT-Struktur erzeugt und die minimalen und maximalen Kohärenzradien gewählt. In Kasten **720** bis **775** werden in derselben Weise, wie oben bei den Original-Zeichenketten in **Fig.** 6 beschrieben, aus der Referenz-Zeichenkette Referenztupel und Referenzindizes erzeugt. In Kasten **780** werden wie oben beschrieben die Referenz-Verschiebungswerte Δ berechnet. In Kasten **790** wird der erzeugte Referenzindex dazu verwendet, auf einen Datensatz bzw. eine Liste in einer Zelle der Referenzstruktur zuzugreifen, welche durch einen Wert bezeichnet wird, der gleich dem erzeugten Referenzindex ist. In Kasten **795** wird überprüft, ob der Datensatz bzw. die Liste leer ist. Wenn dies der Fall ist, wird die Verarbeitung in Kasten **755** mit der Erzeugung des nächsten Referenzindexes fortgesetzt. Wenn die Zelle hingegen einen Datensatz bzw. eine Liste enthält, wird in Kasten **800** der erste Datensatz der Liste ausgewählt. In Kasten **805** wird wie oben beschrieben ein Zählindex (EIT) berechnet. Dann wird wie oben beschrieben in Kasten **810** mittels des Zählindexes auf eine Zelle in der EIT zugegriffen. Wenn keine Zelle vorhanden ist, wird in Kasten **815** eine Zelle erzeugt und in dieser Zelle ein Wert c = 0 gespeichert. Wenn eine Zelle vorhanden ist, wird in Kasten **820** der Wert c um 1 erhöht. Der Prozess wird in Kasten **825** und **830** so lange wiederholt, bis alle Datensätze in der Liste gezählt wurden. Wenn keine weiteren Datensätze mehr vorliegen, geht die Verarbeitung weiter zu Kasten

755, um das nächste Referenztupel zu erzeugen.

[0090] Man beachte, dass man abgesehen von der Beschreibung der vorliegenden Erfindung viele Varianten dieses Algorithmus entwickeln kann, die in der Erfindung inbegriffen sind.

BEISPIEL 2

[0091] Die besonders bevorzugte Ausführungsart der vorliegenden Erfindung findet speziell Anwendung in der Genomforschung an lebenden Organismen und insbesondere in der menschlichen Genomforschung, um die Positionen und Funktionen der Nukleotidsequenzen sowie weitere biologische Informationen auf den DNA-Strängen zu ermitteln. Mittels dieses Verfahrens können diese Informationen schneller und genauer ermittelt werden als es nach dem Stand der Technik vor Verwendung der Verfahren bisher möglich war. Im Folgenden wird ein Beispiel angegeben, das keine Einschränkung der Erfindung darstellen soll.

[0092] Obwohl noch nicht das gesamte menschliche Genom entschlüsselt worden ist, können bereits Datenbanken käuflich erworben werden, welche Teilsequenzen von DNA-Strängen enthalten, die in Nukleotid-Sequenzen aufgespalten sind. Diese Daten können leicht in einer Computer-Datenbank gespeichert und für die Daten Verweis-/Zeigeradressen bereitgestellt werden. Mittels Algorithmen wie die oben erörterten werden diese DNA-Original-Sequenzen in zusammenhängende Teilsequenzen oder Nukleotide aufgegliedert, welche die drei oben beschriebenen Regeln erfüllen. Auf Basis dieser zusammenhängenden Teilsequenzen wird ein Satz von mindestens einem Originaltupel gebildet. Mindestens ein Originaltupel dieses Satzes wird durch Zusammenfügen von mindestens zwei nicht zusammenhängenden Teilsequenzen oder Nukleotidzeichen erzeugt. Man beachte, dass an dieses Tupel weitere zusammenhängende Teilsequenzen von Nukleotidzeichen angehängt werden können. Dann wird ein der DNA-Original-Sequenz zugehöriger eindeutiger Originalindex erzeugt (siehe oben gezeigtes Beispiel). Dann wird eine Referenz-Nukleotidsequenz ausgewählt. Die aus Nukleotidzeichen bestehenden Referenztupel werden wie oben erläutert erzeugt. Auch hier muss durch Zusammenfügen von mindestens zwei zusammenhängenden Referenz-Teilsequenzen mindestens ein Referenztupel erzeugt werden. Mittels desselben Algorithmus zur Indexerzeugung, mittels dessen die Originalindizes erzeugt wurden, wird ein eindeutiger Referenzindex erzeugt (siehe obiges Beispiel). Die Referenzindizes und die Originalindizes werden genau so wie beim allgemeinen Verfahren unter Verwendung der Referenztabelle miteinander verglichen. Die besonders bevorzugte Referenztabelle ist eine Matrix. Ebenso wie oben werden die Übereinstimmungen in einer zweiten Speicherstruktur EIT erfasst. Die besonders bevorzugte EIT ist eine dynamische Hash-Tabelle. Die DNA-Original-Sequenzen werden mittels der c-Werte der EIT ausgewählt, deren Wert einen vorgegebenen Schwellenwert überschreitet.

[0093] Das Verfahren ist beim Vergleich der Referenz-Sequenzen der Nukleotide beim Genom von E. coli erfolgreich getestet worden, das etwa 4 Millionen Nukleotide enthält.

[0094] Die vorliegende Erfindung besitzt viele weitere Anwendungen, einschließlich des Vergleichs von Aminosäuresequenzen in Proteinen, der Erkennung von Buchstaben-Zeichenketten, der Spracherkennung und der Musikerkennung, ist aber nicht darauf beschränkt.

BEISPIEL 3

[0095] Die Anwendung des Verfahrens auf den Vergleich von Aminosäuresequenzen mit einer aus Aminosäurezeichen bestehenden Original-Proteinsequenz ähnelt dem DNA-Beispiel und ist zum Teil bereits beschrieben worden. Aus Aminosäurezeichen bestehende Original-Proteinsequenzen können entweder als Datenbanken käuflich erworben oder von Hand in den Computerspeicher eingegeben werden. Den **20** möglichen Aminosäurezeichen werden wie oben angegeben stellvertretend alphanumerische Zeichen zugewiesen, um sie eindeutig zu kennzeichnen. Diese Zeichen erhalten einen eindeutigen Zahlenwert, sodass die aus Aminosäurezeichen bestehenden Sequenzen in Zahlenketten umgewandelt werden können. Ebenso wie oben werden Original- und Referenztupel und -indizes erzeugt und miteinander verglichen. Desgleichen werden wie oben die Zahlenwerte der Übereinstimmungen in einer EIT erfasst und die übereinstimmenden Original-Sequenzen ermittelt, indem auf diejenigen Zählzellen zugegriffen wird, deren c-Werte einen vorgegebenen Schwellenwert überschreiten.

BEISPIEL 4

[0096] Die Erkennung von Buchstaben-Zeichenketten erfolgt auf fast dieselbe Weise. Jedem Buchstaben wird ein eindeutiger Zahlenwert zugewiesen. Die Buchstaben-Zeichenketten werden in Zahlenfolgen umge-

wandelt, die mittels des allgemeinen Verfahrens verarbeitet werden, um die Original- und Referenztupel und -indizes zu erzeugen. Die Indizes werden ebenso wie oben miteinander verglichen, zur Deckung gebracht und erfasst. Die Original-Buchstaben-Zeichenketten werden aus der Datenbank ermittelt, indem man nach denjenigen c-Werten in der EIT sucht, die einen vorgegebenen Schwellenwert überschreiten.

BEISPIEL 5

[0097] Die Spracherkennung verläuft ebenso. Zuerst wird die Sprache mittels in der Technik bekannter Verfahren in Phoneme umgewandelt. Aus den Sprachmustern werden dann aus Phonemzeichen bestehende Zeichenketten. Jedem Phonemzeichen wird ein eindeutiger Zahlenwert zugewiesen, und die Original- und Referenz-Phonemzeichenketten werden in Zahlenketten umgewandelt. Dann wird wie oben das allgemeine Verfahren angewendet.

BEISPIEL 6

[0098] Unter Verwendung der vorliegenden Erfindung kann auch eine Musikerkennung durchgeführt werden. Ein Lied oder eine Musikaufzeichnung besteht aus einer Folge von Notenzeichen. Jeder Note wird ein Zahlenwert oder ein alphanumerischer Wert zugewiesen. Zum Beispiel werden die Noten der C-Dur-Tonleiter als „CDEFGAHC" dargestellt. Vorzeichen (B und Kreuz), Pausen und Notenlängen (Viertel- und Achtelnoten usw.) können durch zusätzliche alphanumerische Werte bezeichnet werden. Ausgehend von bekannten Verfahren können weitere Umwandlungsverfahren entwickelt werden. Dann wird aus dem Lied oder der Musikaufzeichnung eine Folge alphanumerischer Zeichen, die unter Anwendung der obigen Beschreibung in eine Zahlenfolge umgewandelt werden können. Dann wird das allgemeine Verfahren angewendet, um eine Referenz-Notenfolge mit Original-Notenfolgen in der Datenbank zu vergleichen.

Patentansprüche

1. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank, welches die folgenden Schritte umfasst:
Erzeugen eines oder mehrerer Originaltupel für jede der Original-Zeichenketten (**200**) in der Datenbank durch:
a. Aufgliedern (**15**) jeder Original-Zeichenkette (**200**) in zwei oder mehr Original-Teilzeichenketten (**210**) von zusammenhängenden Zeichen;
b. Zusammenfügen von zwei oder mehr nicht zusammenhängenden Original-Teilzeichenketten der Original-Zeichenkette (**200**), um ein oder mehr der Original-Zeichenkette (**200**) zugeordnete Originaltupel zu bilden;
Erzeugen (**25**) eines eindeutigen Originalindexes (**312**, **412**) für jedes aus einer Original-Zeichenkette erzeugte Originaltupel, wobei der Originalindex (**312**, **412**) derjenigen Original-Zeichenkette zugeordnet ist (**30**), aus welcher das Originaltupel erzeugt wurde;
Erzeugen eines oder mehrerer Referenztupel aus der Referenz-Zeichenkette durch:
c. Aufgliedern (**35**) der Referenz-Zeichenkette in zwei oder mehr Referenz-Teilzeichenketten von zusammenhängenden Zeichen;
d. Zusammenfügen (**40**) von mindestens zwei nicht zusammenhängenden Referenz-Teilzeichenketten, um mindestens ein Referenztupel zu bilden;
Erzeugen (**45**) eines eindeutigen Referenzindexes (**350**, **450**) für jedes Referenztupel in derselben Weise, wie der Originalindex (**312**, **412**) erzeugt wurde;
Vergleichen (**50**) mindestens eines Referenzindexes (**350**, **450**) mit mindestens einem Originalindex (**312**, **412**);
Erfassen (**55**) der Übereinstimmungen zwischen dem Referenzindex (**350**, **450**) und dem Originalindex (**312**, **412**);
Auswählen (**60**) einer Original-Zeichenkette (**200**) aus der Datenbank anhand der Anzahl der Übereinstimmungen zwischen einem oder mehreren Originalindizes (**312**, **412**) und einem oder mehreren Referenzindezes (**350**, **450**).

2. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem alle Original-Teilzeichenketten (**210**) eine fest vorgegebene Länge (**225**) aufweisen.

3. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem die Original-Teilzeichenketten (**210**) unterschiedlich lang (**225**) sind.

4. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem die gebildeten Originaltupel eine fest vorgegebene Länge (**225**) aufweisen.

5. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem die gebildeten Originaltupel unterschiedlich lang (**225**) sind.

6. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem die Originaltupel unter Anwendung der folgenden Regeln gebildet werden:
Jede das Tupel bildende nachfolgende Teilzeichenkette hat eine in Zeichen gemessene Startposition (**220**), die von einem Anfangszeichen der Original-Zeichenkette weiter entfernt ist als die Startposition (**220**) aller vorangehenden das Tupel bildenden Teilzeichenketten;
keine Teilzeichenkette kann zur Bildung eines Tupels verwendet werden, wenn die Teilzeichenkette infolge ihrer Startposition (**220**) und ihrer in Zeichen gemessenen Länge (**225**) die Länge (**225**) der Original-Zeichenkette überschreitet;
die Startzeichen zweier aufeinander folgender Teilzeichenketten in einer Original-Zeichenkette müssen zwischen einem minimalen und einem maximalen Kohärenzradius der beiden liegen, wobei die Abstandsradien in Zeichen gemessen werden.

7. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem der Originalindex (**312**, **412**) und der Referenzindex (**350**, **450**) mittels desselben Algorithmus erzeugt werden.

8. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem der Originalindex (**312**, **412**) und der Referenzindex (**350**, **450**) Zahlen sind.

9. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 1, bei welchem der Originalindex (**312**, **412**) und der Referenzindex (**350**, **450**) ganze Zahlen sind.

10. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 9, welches ferner die folgenden Schritte umfasst:
Verwenden (**30**) des Originalindexes (**312**, **412**) zum Zeigen auf eine Zelle (**310**, **410**) in einer ersten Speicher-Referenzstruktur und zum Speichern (**30**) eines zu einer Original-Zeichenkette gehörenden Datensatzes (**314**, **414**) in der Zelle (**310**, **410**),
Vergleichen (**50**) mindestens eines Referenzindexes (**350**, **450**) mit mindestens einem Originalindex (**312**, **412**) unter Verwendung der Speicher-Referenzstruktur;
Speichern (**55**) der erfassten Ergebnisse in einer zweiten Speicher-Referenzstruktur (**500**).

11. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die im Datensatz (**314**, **414**) enthaltenen Daten ein Zeiger (**315**, **415**) sind, der zum Suchen derjenigen Original-Zeichenkette in der Datenbank dient, die das Tupel enthält, aus welchem der Originalindex (**312**, **412**) abgeleitet wurde.

12. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem der Datensatz (**314**, **414**) einen Verschiebungswert (**320**, **420**) enthält, der zur Ermittlung der Position der mit der Referenz-Zeichenkette der Original-Zeichenkette übereinstimmenden Zeichenfolge dient.

13. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 11, bei welchem die Zelle (**310**, **410**) eine Liste (**328**, **428**) von Datensätzen (**314**, **414**) enthält, in welchen wiederum ein Verweis auf eine Original-Zeichenkette enthalten ist.

14. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 11, bei welchem die Zelle (**310**, **410**) eine Liste (**328**, **428**) von Datensätzen (**314**, **414**) enthält, in welchen wiederum ein Verweis auf eine Original-Zeichenkette und ein zugehöriger Verschiebungswert (**320**, **420**) enthalten ist.

15. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 12, bei welchem der Verschiebungswert (**320**, **420**) anhand der in Zeichen gemessenen Position von mindestens einer Teilzeichenkette der Original-Zeichenkette berechnet wird, welche zur Bildung des Tupels des Originalindexes (**312**, **412**) dient.

16. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 12, bei welchem der Verschiebungswert (**320**, **420**) gleich dem in Zeichen gemessenen Abstand zwischen dem Mittelwert der Position jeder zur Bildung des Tupels verwendeten Original-Teilzeichenkette und einem bestimmten Zeichen in der Original-Zeichenkette ist.

17. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die Übereinstimmung zwischen der Original-Zeichenkette und der Referenz-Zeichenkette genau ist.

18. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die Übereinstimmung zwischen der Original-Zeichenkette und der Referenz-Zeichenkette ungefähr ist.

19. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die Daten in der zweiten Speicherstruktur (**500**) einen Wert beinhalten, der den Grad der Übereinstimmung zwischen der Original-Zeichenkette und der Referenz-Zeichenkette anzeigt.

20. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die erste Referenzstruktur (**300**) eine Datenstruktur ist, die solche Strukturen wie einen Vektor, eine Matrix und eine Hash-Tabelle beinhaltet.

21. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 20, bei welchem die erste Referenzstruktur (**300**) eine Matrix ist.

22. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die zweite Referenzstruktur (**500**) eine statische Datenstruktur ist, die solche Strukturen wie einen Vektor, eine Matrix und eine Hash-Tabelle beinhaltet.

23. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die zweite Referenzstruktur (**500**) eine dynamische Datenstruktur ist, die solche Strukturen wie einen Vektor, eine Matrix und eine Hash-Tabelle beinhaltet.

24. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 23, bei welchem die zweite Referenzstruktur (**500**) eine dynamische Hash-Tabelle ist.

25. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach Anspruch 10, bei welchem die zweite Referenzstruktur (**500**) jedes Mal aktualisiert wird, wenn ein Referenzindex (**350**, **450**) mit einem Originalindex (**312**, **412**) übereinstimmt.

26. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 25, bei welchem die Referenz-Zeichenketten Nukleotid-Sequenzen darstellen und die Original-Zeichenketten (**200**) DNA-Sequenzen darstellen.

27. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 25, bei welchem die Referenz-Zeichenketten Aminosäuresequenzen und die Original-Zeichenketten (**200**) Proteinsequenzen darstellen.

28. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 25, bei welchem die Referenz-Zeichenketten Buchstaben-Zeichenketten und die Original-Zeichenketten (**200**) ebenfalls Buchstaben-Zeichenketten darstellen.

29. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 25, bei welchem die Referenz-Zeichenketten Pho-

nemfolgen und die Original-Zeichenketten (**200**) ebenfalls Phonemfolgen darstellen.

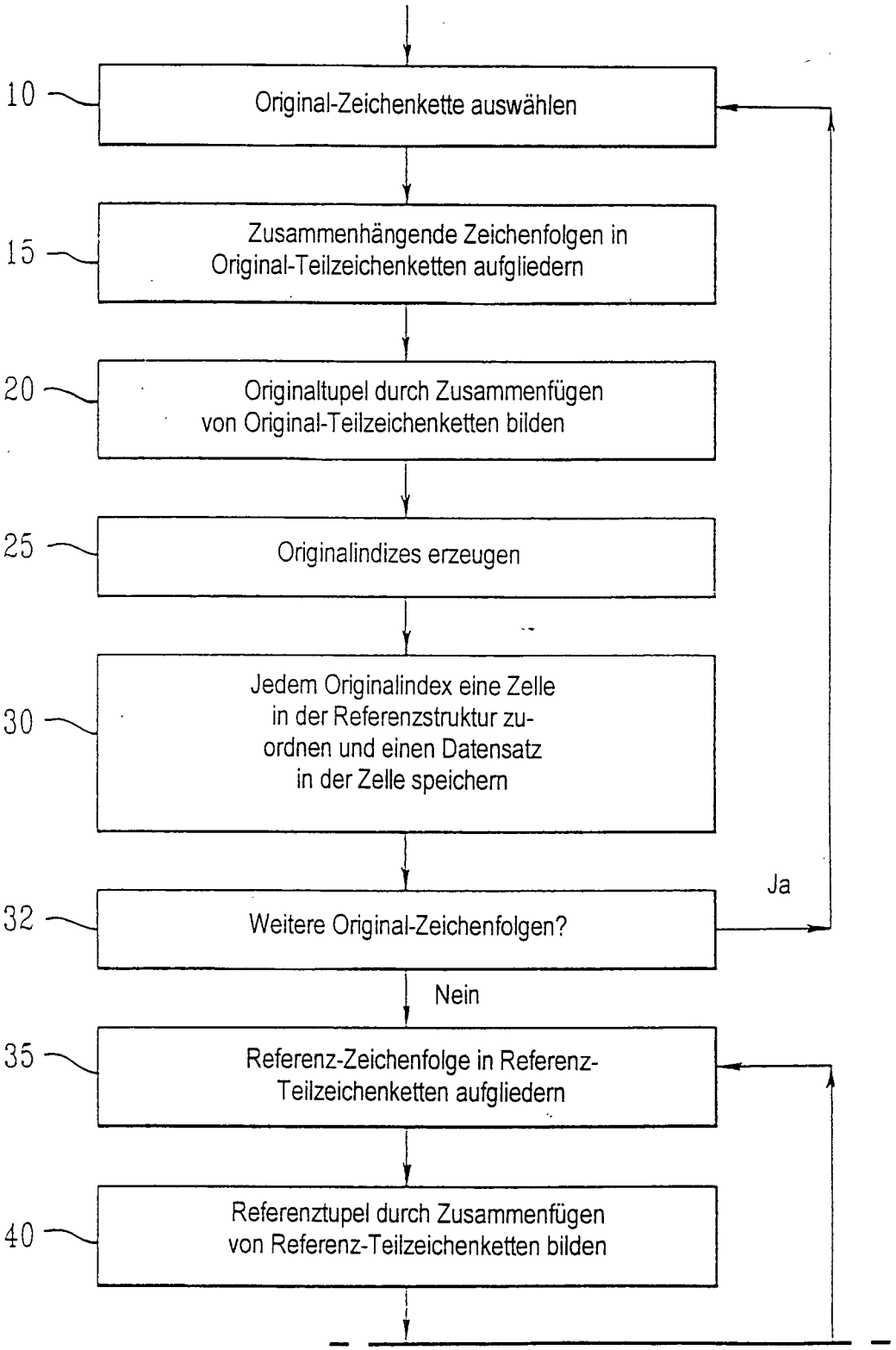
30. Verfahren zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank nach einem der Ansprüche 1 bis 25, bei welchem die Referenz-Zeichenketten Notenfolgen und die Original-Zeichenketten (**200**) ebenfalls Notenfolgen darstellen.

31. Computersystem zum Auffinden einer Referenz-Zeichenkette in einer oder mehreren Original-Zeichenketten (**200**) in einer Datenbank, welches Folgendes umfasst:
eine Datenbank mit einem Satz Original-Zeichenketten (**200**);
Mittel zum Erzeugen mindestens eines Originaltupels für jede der Original-Zeichenketten (**200**) in der Datenbank,
wobei das Tupel gebildet wird durch:
a. Aufgliedern jeder Original-Zeichenkette in zwei oder mehr zusammenhängende Original-Teilzeichenketten (**210**);
b. Bilden mindestens eines zu jeder Original-Zeichenkette gehörenden Originaltupels durch Zusammenfügen von mindestens zwei nicht zusammenhängenden Teilzeichenketten der Original-Zeichenkette;
einen eindeutigen Originalindex (**312**, **412**) für jedes Originaltupel, der durch einen Index-Algorithmus aus der Original-Zeichenkette gebildet wurde, wobei der Originalindex (**312**, **412**) derjenigen Original-Zeichenkette zugeordnet ist, aus welcher das Originaltupel erzeugt wurde;
eine erste Speicher-Referenzstruktur (**300**) mit Zellen (**310**, **410**), wobei auf die Zellen (**310**, **410**) durch den Originalindex (**312**, **412**) zugegriffen wird und die Zellen (**310**, **410**) Daten enthalten, welche zu derjenigen Original-Zeichenkette gehören, aus der das Originaltupel gebildet wurde;
ein oder mehrere Referenztupel, die aus der Referenz-Zeichenkette gebildet wurden durch:
c. Aufgliedern der Referenz-Zeichenkette in zwei oder mehr nicht zusammenhängende Referenz-Teilzeichenketten;
d. Bilden mindestens eines Referenztupels durch Zusammenfügen von mindestens zwei Referenz-Teilzeichenketten;
einen eindeutigen Referenzindex (**350**, **450**) für jedes mittels des Index-Algorithmus erzeugte Referenztupel, wobei der Referenzindex (**350**, **450**) mit mindestens einem Originalindex (**312**, **412**) verglichen wird;
eine zweite Speicher-Referenzstruktur (**500**) zum Erfassen der Übereinstimmungen zwischen dem Referenzindex (**350**, **450**) und dem Originalindex (**312**, **412**) einer Original-Zeichenkette in der Datenbank, wobei die Original-Zeichenkette anhand der Anzahl der Übereinstimmungen zwischen einem oder mehreren Originalindizes (**312**, **412**) und einem oder mehreren Referenzindizes (**350**, **450**) ausgewählt wird.

Es folgen 12 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1A



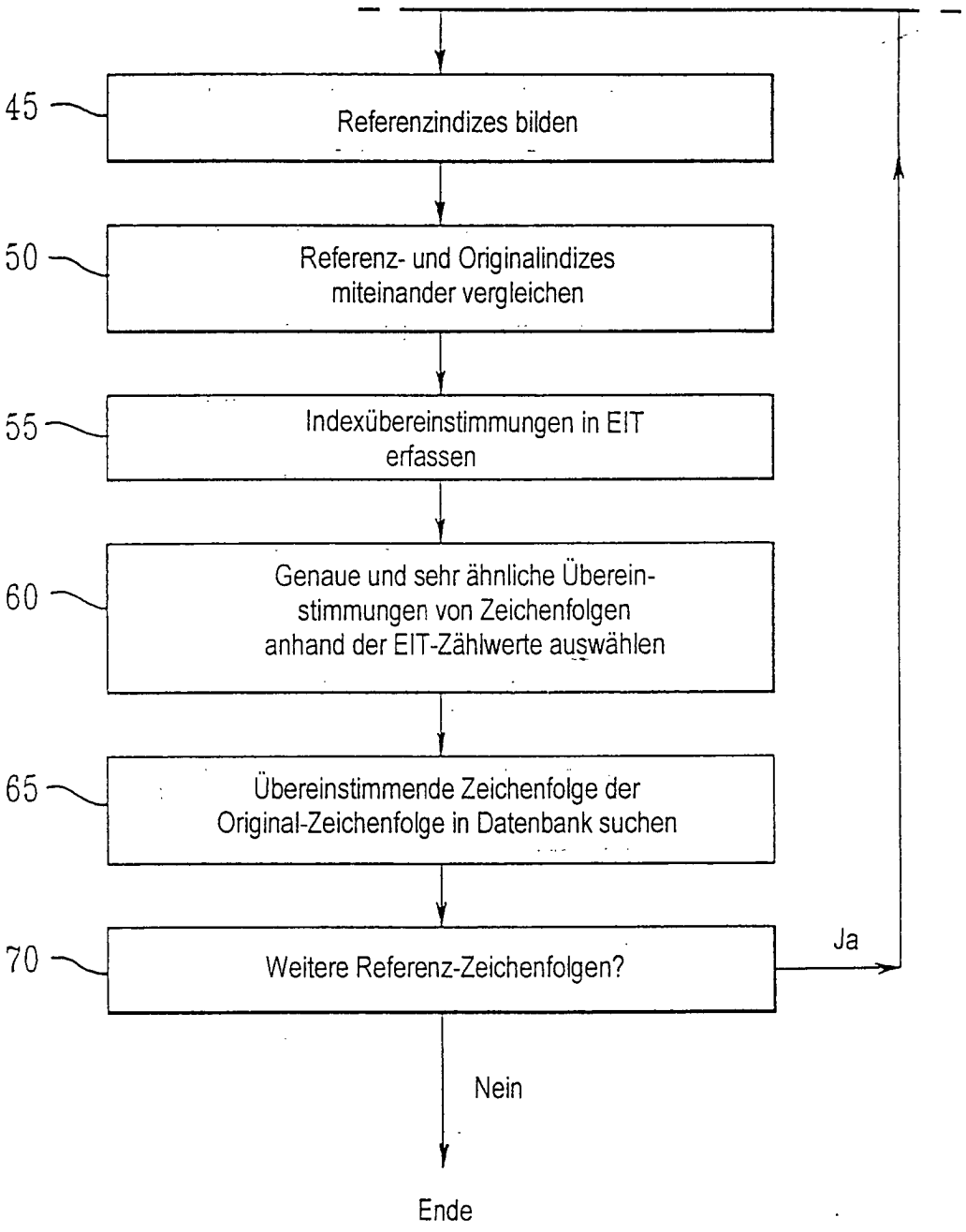

45
Referenzindizes bilden
50
Referenz- und Originalindizes miteinander vergleichen
55
Indexübereinstimmungen in EIT erfassen
60
Genaue und sehr ähnliche Übereinstimmungen von Zeichenfolgen anhand der EIT-Zählwerte auswählen
65
Übereinstimmende Zeichenfolge der Original-Zeichenfolge in Datenbank suchen
70
Weitere Referenz-Zeichenfolgen?
Ja
Nein
Ende

FIG. 1B

FIG. 2

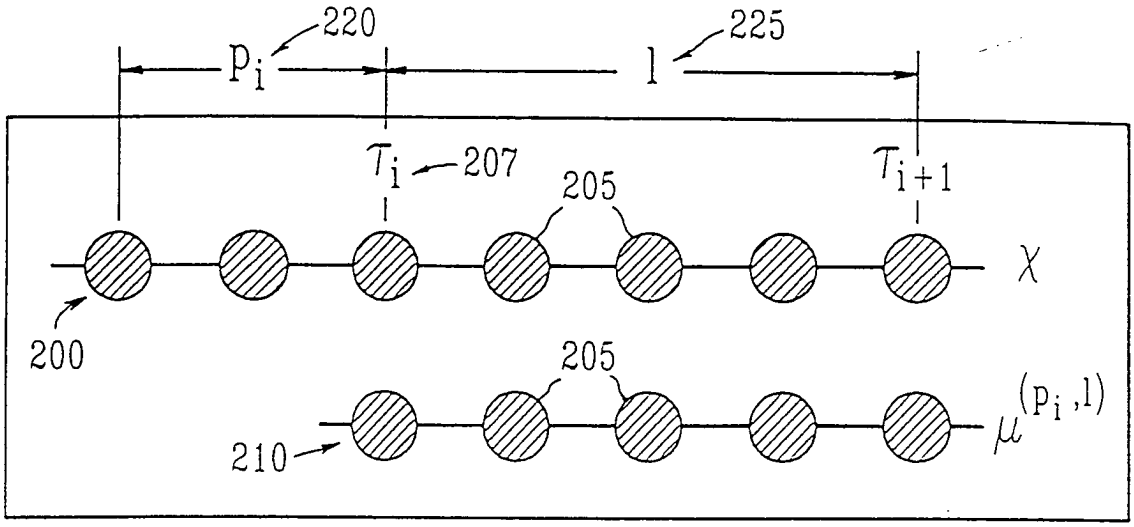
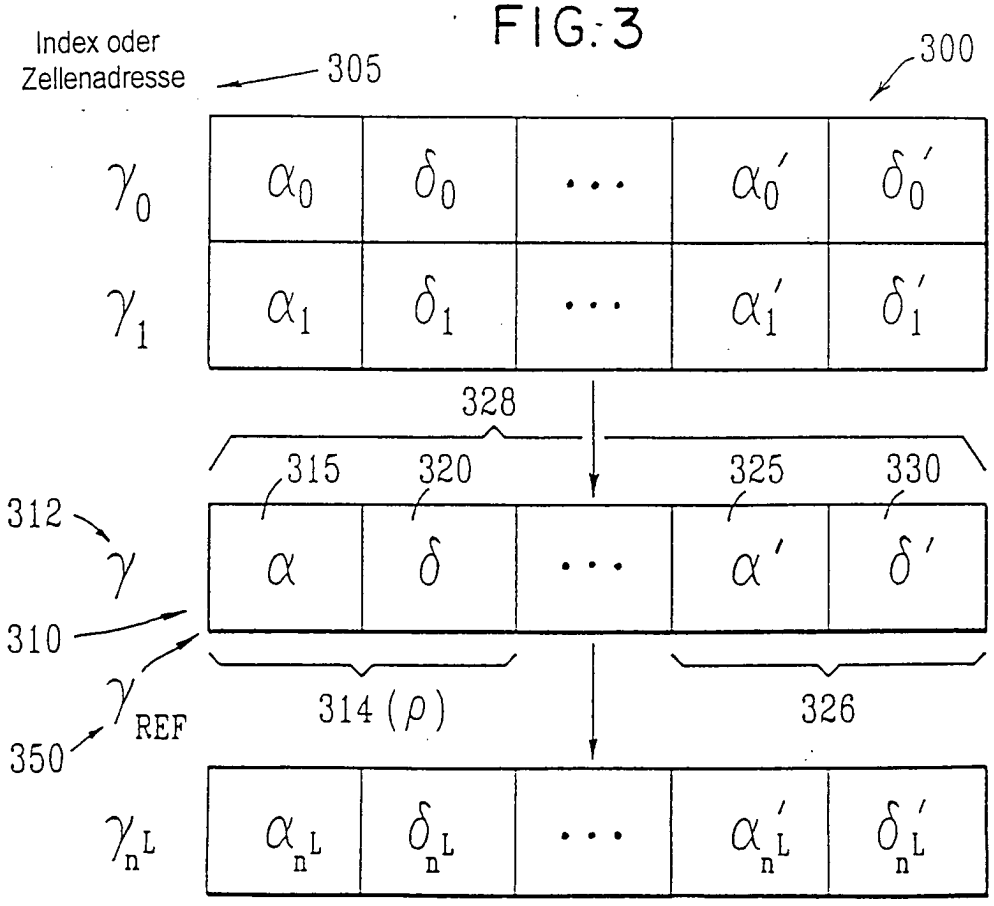

220
225
p_i
l
τ_i
207
205
τ_{i+1}
χ
200
205
$\mu^{(p_i,l)}$
210

FIG. 3


Index oder Zellenadresse
305
300
γ_0
α_0 δ_0 ... α_0' δ_0'
γ_1
α_1 δ_1 ... α_1' δ_1'
328
315
320
325
330
312
γ
α δ ... α' δ'
310
γ_{REF}
350
314 (ρ)
326
γ_{n_L}
α_{n_L} δ_{n_L} ... α_{n_L}' δ_{n_L}'

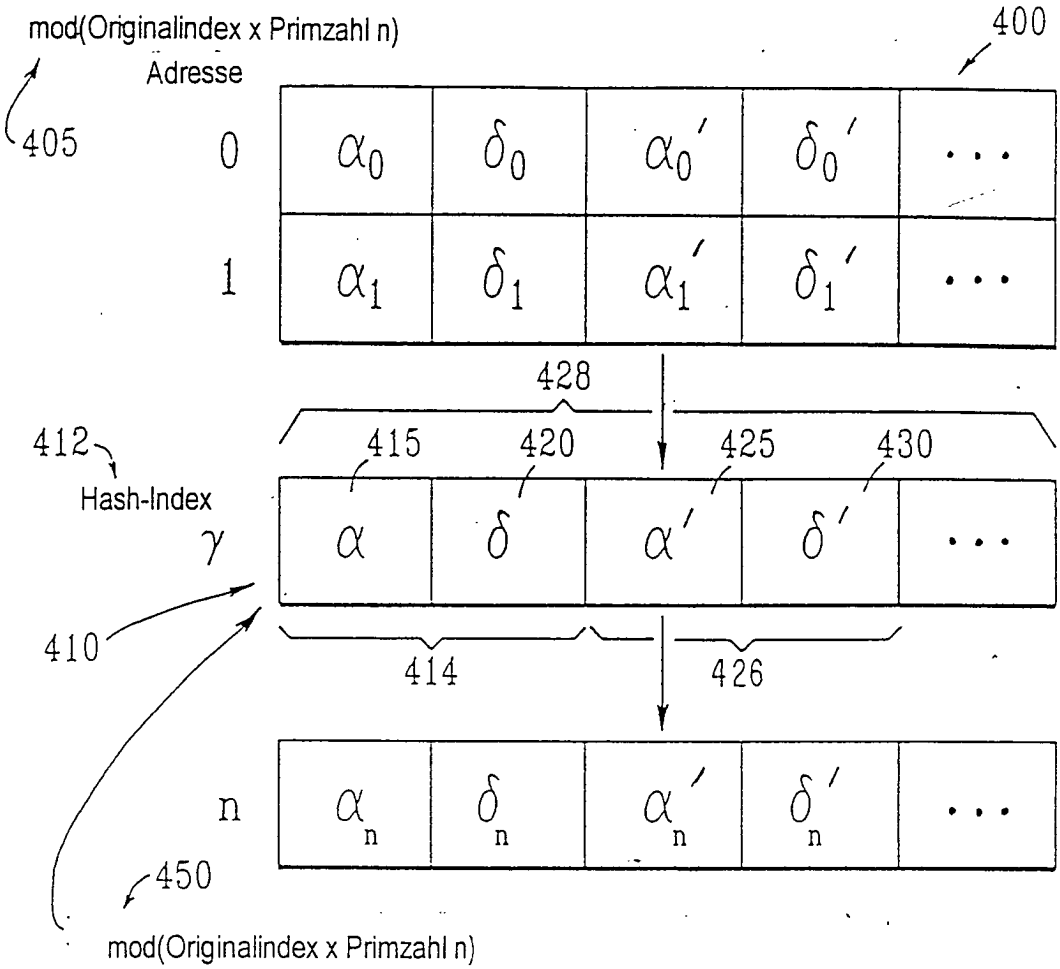

mod(Originalindex x Primzahl n)
Adresse
405
400
0
1
n
α_0 δ_0 α_0' δ_0' ...
α_1 δ_1 α_1' δ_1' ...
α_n δ_n α_n' δ_n' ...
428
412
Hash-Index
γ
415
420
425
430
α δ α' δ' ...
410
414
426
450
mod(Originalindex x Primzahl n)

FIG. 4

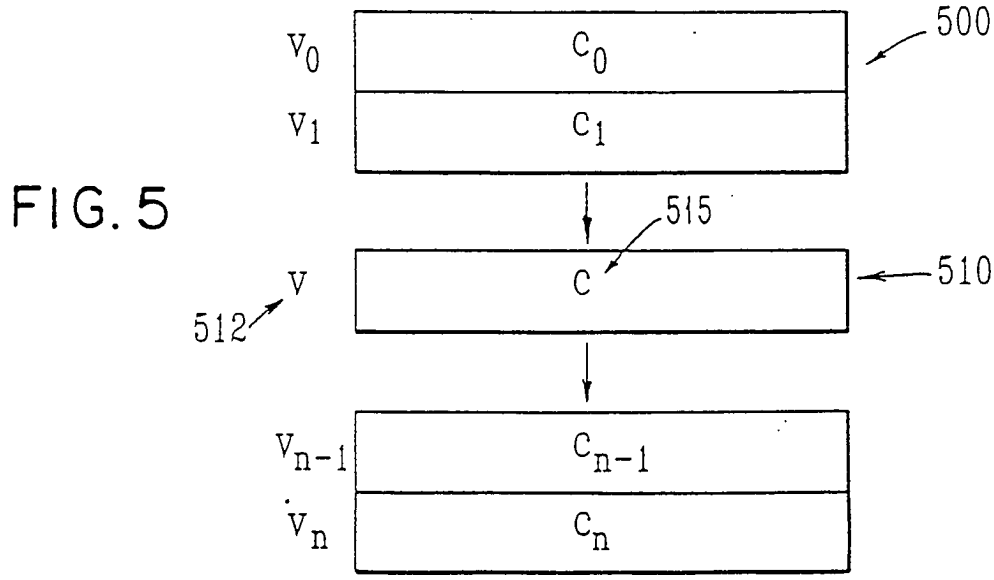

500
v_0 c_0
v_1 c_1
515
v
c
510
512
v_{n-1} c_{n-1}
v_n c_n

FIG. 5

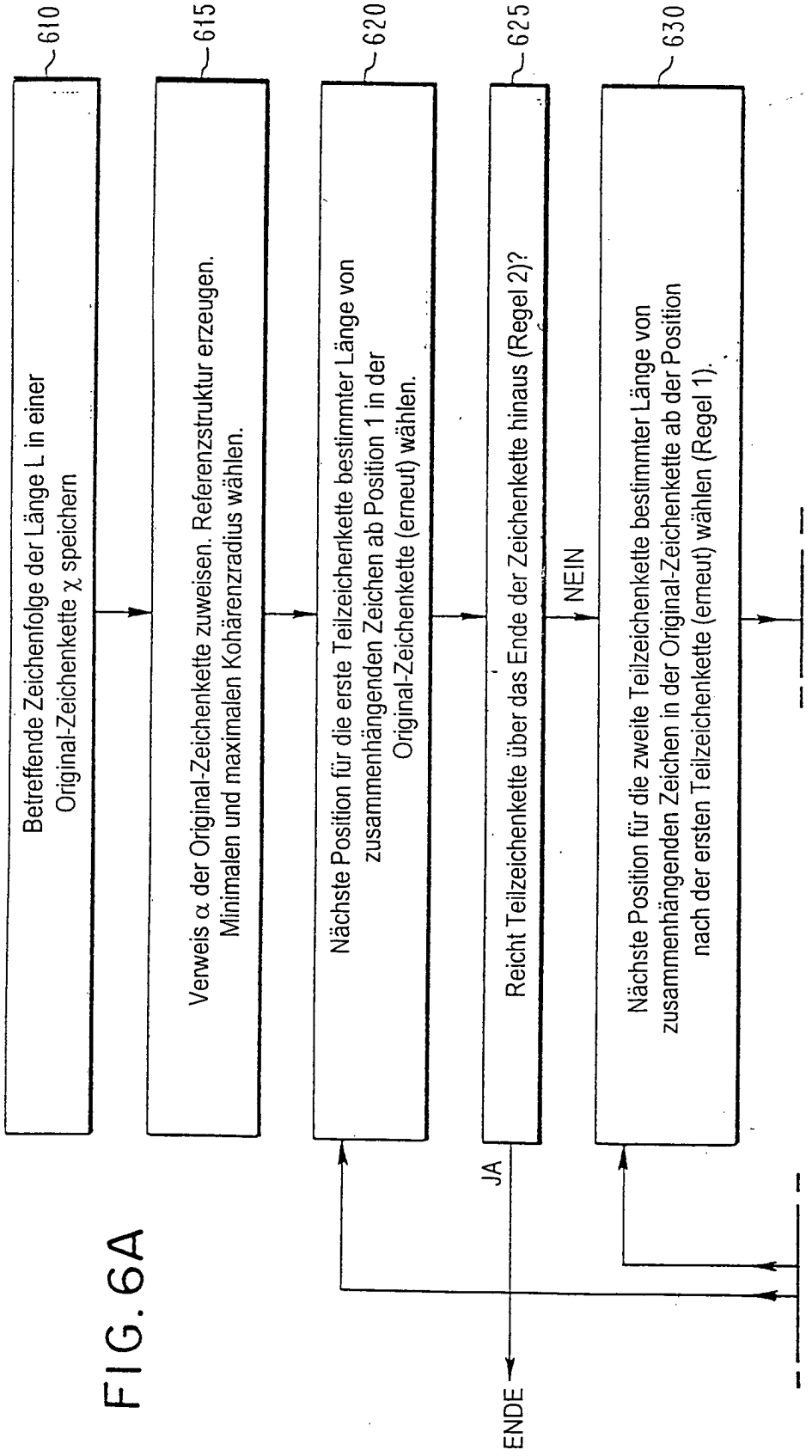

610
Betreffende Zeichenfolge der Länge L in einer Original-Zeichenkette χ speichern
615
Verweis α der Original-Zeichenkette zuweisen. Referenzstruktur erzeugen. Minimalen und maximalen Kohärenzradius wählen.
620
Nächste Position für die erste Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen ab Position 1 in der Original-Zeichenkette (erneut) wählen.
625
Reicht Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
JA
NEIN
630
Nächste Position für die zweite Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen in der Original-Zeichenkette ab der Position nach der ersten Teilzeichenkette (erneut) wählen (Regel 1).
ENDE

FIG. 6A

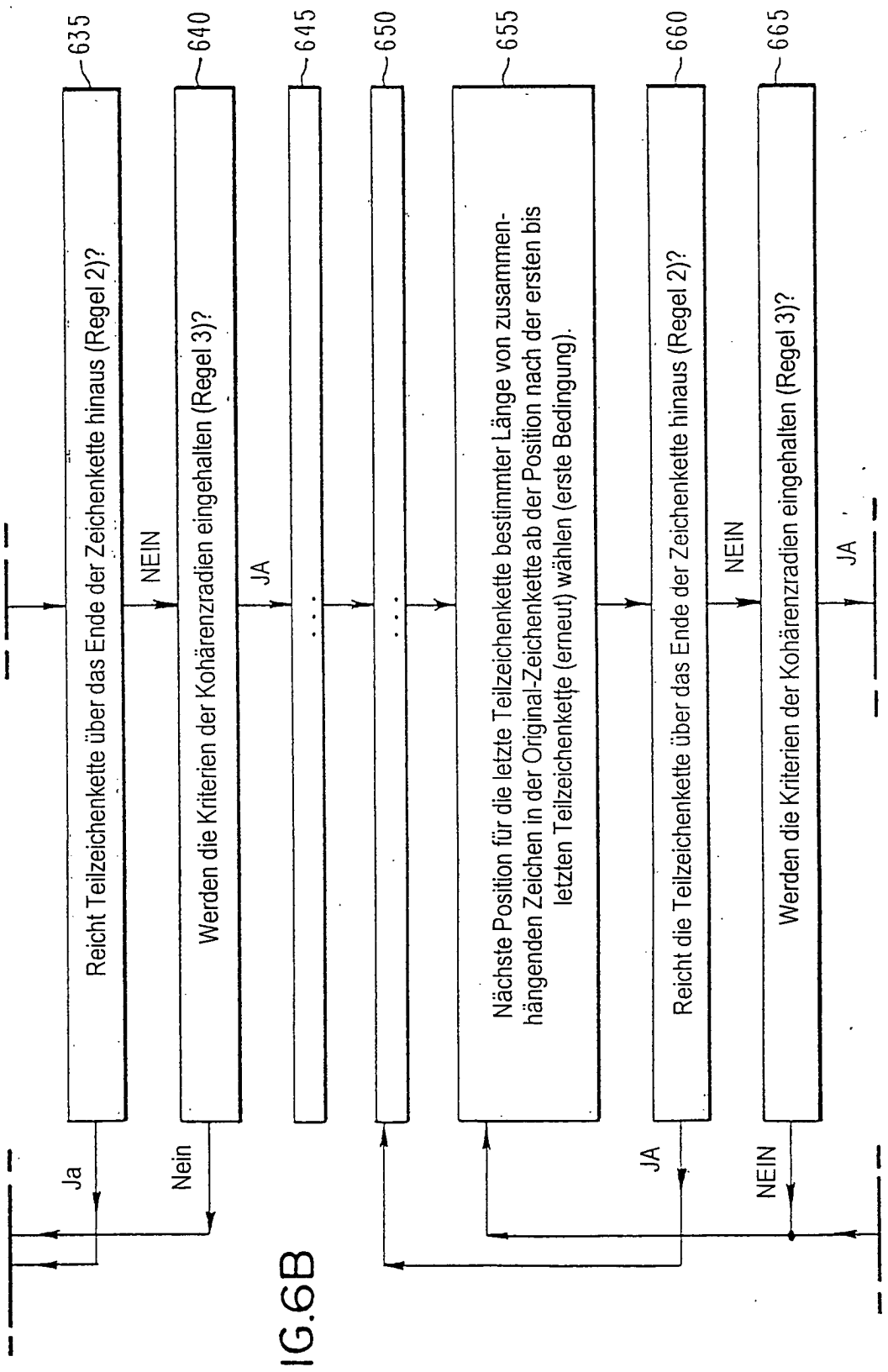

635
Reicht Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
NEIN
640
Werden die Kriterien der Kohärenzradien eingehalten (Regel 3)?
JA
645
650
655
Nächste Position für die letzte Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen in der Original-Zeichenkette ab der Position nach der ersten bis letzten Teilzeichenkette (erneut) wählen (erste Bedingung).
660
Reicht die Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
NEIN
665
Werden die Kriterien der Kohärenzradien eingehalten (Regel 3)?
JA
Ja
Nein
JA
NEIN

FIG.6B

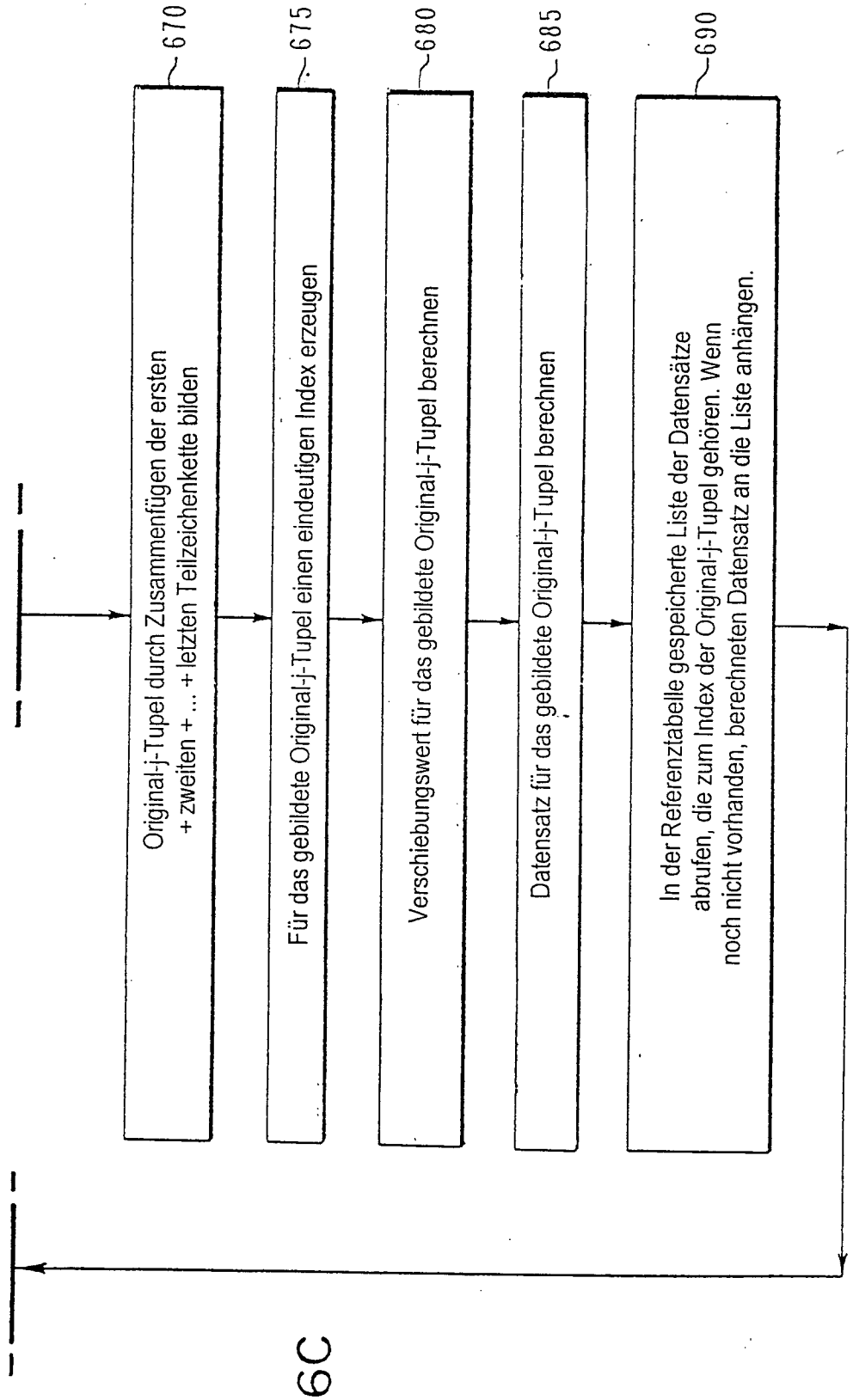

670
Original-j-Tupel durch Zusammenfügen der ersten + zweiten + ... + letzten Teilzeichenkette bilden
675
Für das gebildete Original-j-Tupel einen eindeutigen Index erzeugen
680
Verschiebungswert für das gebildete Original-j-Tupel berechnen
685
Datensatz für das gebildete Original-j-Tupel berechnen
690
In der Referenztabelle gespeicherte Liste der Datensätze abrufen, die zum Index der Original-j-Tupel gehören. Wenn noch nicht vorhanden, berechneten Datensatz an die Liste anhängen.

FIG. 6C

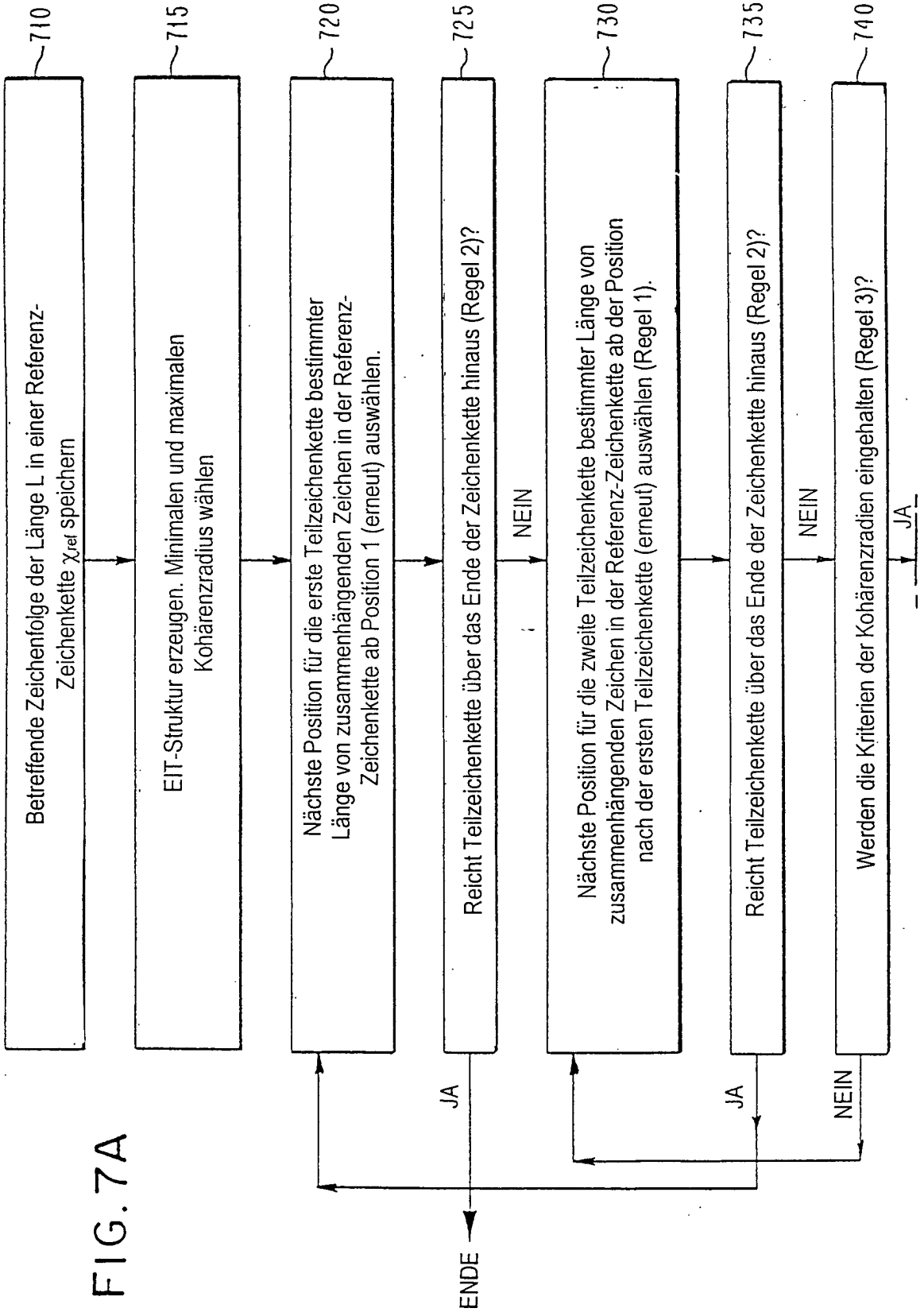

710
Betreffende Zeichenfolge der Länge L in einer Referenz-Zeichenkette χ_{ref} speichern
715
EIT-Struktur erzeugen. Minimalen und maximalen Kohärenzradius wählen
720
Nächste Position für die erste Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen in der Referenz-Zeichenkette ab Position 1 (erneut) auswählen.
725
Reicht Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
NEIN
730
Nächste Position für die zweite Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen in der Referenz-Zeichenkette ab der Position nach der ersten Teilzeichenkette (erneut) auswählen (Regel 1).
735
Reicht Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
NEIN
740
Werden die Kriterien der Kohärenzradien eingehalten (Regel 3)?
JA
JA
JA
NEIN
ENDE

FIG. 7A

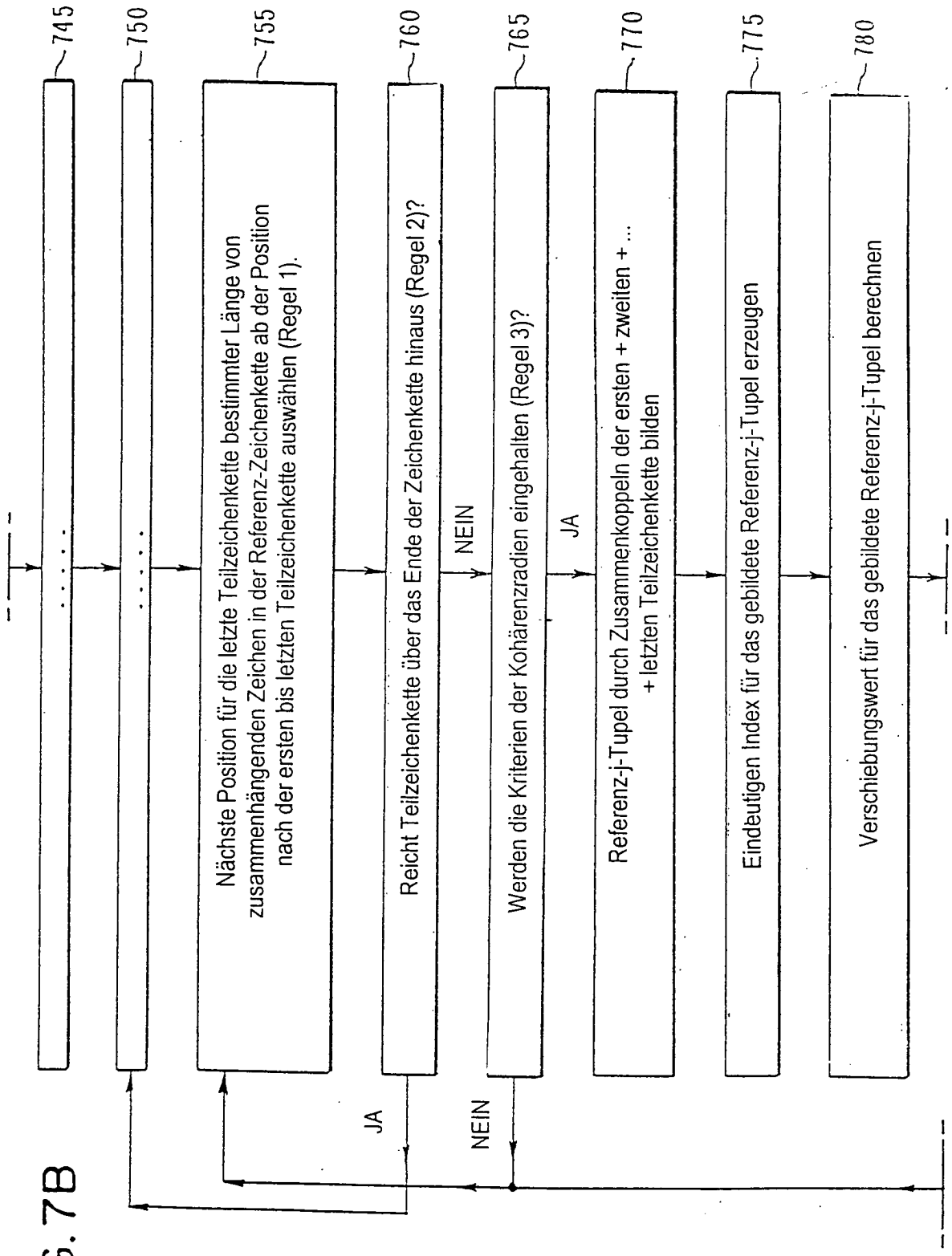

745
750
755
Nächste Position für die letzte Teilzeichenkette bestimmter Länge von zusammenhängenden Zeichen in der Referenz-Zeichenkette ab der Position nach der ersten bis letzten Teilzeichenkette auswählen (Regel 1).
760
Reicht Teilzeichenkette über das Ende der Zeichenkette hinaus (Regel 2)?
JA
NEIN
765
Werden die Kriterien der Kohärenzradien eingehalten (Regel 3)?
NEIN
JA
770
Referenz-j-Tupel durch Zusammenkoppeln der ersten + zweiten + ... + letzten Teilzeichenkette bilden
775
Eindeutigen Index für das gebildete Referenz-j-Tupel erzeugen
780
Verschiebungswert für das gebildete Referenz-j-Tupel berechnen

FIG. 7B

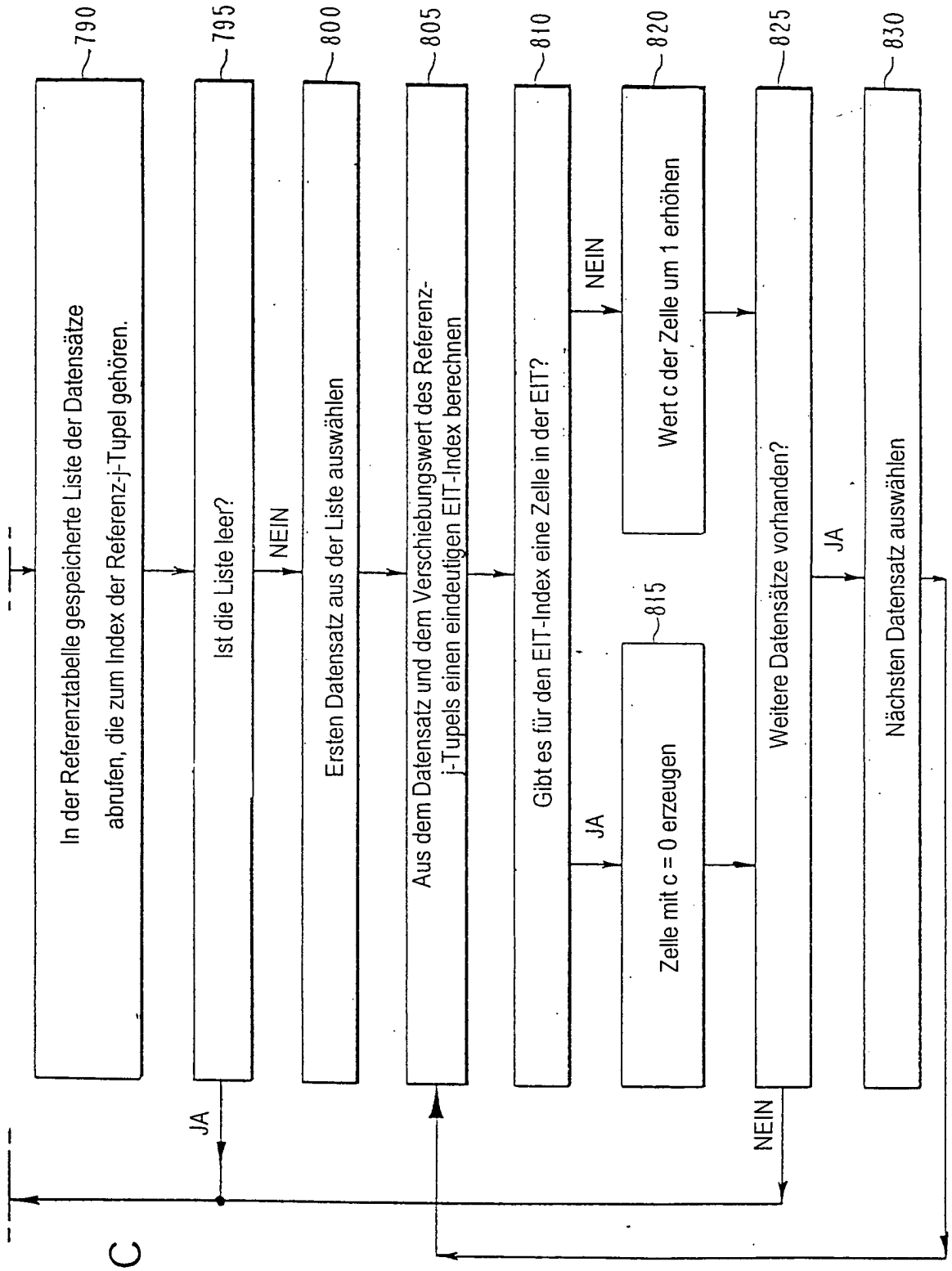

790
In der Referenztabelle gespeicherte Liste der Datensätze abrufen, die zum Index der Referenz-j-Tupel gehören.
795
Ist die Liste leer?
JA
NEIN
800
Ersten Datensatz aus der Liste auswählen
805
Aus dem Datensatz und dem Verschiebungswert des Referenz-j-Tupels einen eindeutigen EIT-Index berechnen
810
Gibt es für den EIT-Index eine Zelle in der EIT?
JA
NEIN
815
Zelle mit c = 0 erzeugen
820
Wert c der Zelle um 1 erhöhen
825
Weitere Datensätze vorhanden?
JA
NEIN
830
Nächsten Datensatz auswählen

FIG. 7C

FIG. 8A

ZEICHENERKLÄRUNG

X	Satz von Original-Zeichenketten, welche eine Datenbank definieren
χ_i	Eine Original-Zeichenkette in der Datenbank von Original-Zeichenketten
χ_{ref}	Eine Referenz-Zeichenkette
χ^0	Eine Zeichenfolge in einer Original-Zeichenkette
χ^0_{ref}	Eine Zeichenfolge in einer Referenz-Zeichenkette
N_χ	Gesamtzahl der Original-Zeichenketten in der Datenbank
μ	Eine Teilzeichenkette
τ	Wert eines Zeichens in einer Zeichenkette oder Teilzeichenkette
τ_i	Wert eines Zeichens an der i-ten Position in einer Zeichenkette oder Teilzeichenkette
n_τ	Anzahl möglicher Zeichenwerte
p_i	Die i-te Position in einer Zeichenkette oder Teilzeichenkette
$\mu_{(p,l)}$	Eine Teilzeichenkette der Länge von l Zeichen ab der Position p einer längeren Zeichenkette
K	Anzahl der durch Aufgliedern einer längeren Zeichenkette erhaltenen j-Tupel in einem Satz

FIG. 8B

ZEICHENERKLÄRUNG

M	Satz von Teilzeichenketten, die durch Aufgliederung gebildet wurden
ξ	Ein Originaltupel
ξ_{ref}	Ein Referenztupel
j-Tupel	Ein aus j Teilzeichenketten gebildetes Tupel, von denen mindestens zwei nicht zusammenhängend sind
$\xi^{(j,L)}$	Ein j-Tupel der Länge L und der Tupelordnung j
$\xi_k^{(j,L)}$	k-tes j-Tupel der Gesamtlänge L
L_S	In Zeichen gemessene Länge einer gegebenen Zeichenkette
γ	Ein Originalindex
γ_{ref}	Ein Referenzindex
ρ	Ein Datensatz in einer Nachschlagestruktur
α	Ein Zeiger in einem Datensatz
δ	Verschiebungswert für ein Originaltupel
Δ	Verschiebungswert für ein Referenztupel