(19) Deutsches Patent- und Markenamt

(10) **DE 11 2008 003 826 B4** 2015.08.20

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2008 003 826.0**
(86) PCT-Aktenzeichen: **PCT/US2008/061576**
(87) PCT-Veröffentlichungs-Nr.: **WO 2009/131585**
(86) PCT-Anmeldetag: **25.04.2008**
(87) PCT-Veröffentlichungstag: **29.10.2009**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **14.04.2011**
(45) Veröffentlichungstag der Patenterteilung: **20.08.2015**

(51) Int Cl.: ***G06F 9/06*** (2006.01)
G06F 9/00 (2006.01)
G06F 17/30 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
Hewlett-Packard Development Company, L.P., Houston, Tex., US

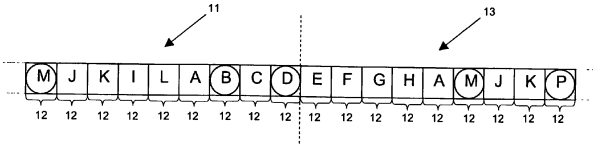
(74) Vertreter:
Samson & Partner Patentanwälte mbB, 80538 München, DE

(72) Erfinder:
Lillibridge, Mark, Palo Alto, Calif., US; Deolalikar, Vinay, Palo Alto, Calif., US

(56) Ermittelter Stand der Technik:
siehe Folgeseiten

(54) Bezeichnung: **Datenverarbeitungsvorrichtung und Verfahren zur Datenverarbeitung**

(57) Hauptanspruch: Datenverarbeitungsvorrichtung zum entduplizierten Speichern . von Eingabedaten, Folgendes umfassend:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Eingabedatensegmente repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst,
einen dünn besetzten Abschnittindex, der Einträge zu nur einigen im Abschnittspeicher gespeicherten Musterdatenabschnitten und Verweise auf Verzeichnisse im Verzeichnisspeicher enthält, die jeweils wenigstens einen Verweis auf die im dünn besetzten Abschnittindex eingetragenen und im Abschnittspeicher gespeicherten Musterdatenabschnitte enthalten,
wobei die Verarbeitungsvorrichtung dafür eingerichtet ist,
Eingabedaten in mehrere Eingabedatensegmente zu verarbeiten, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
ein erstes Eingabedatensegment zu verarbeiten, umfassend:
– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen,
– Identifizieren eines ersten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der im dünn besetzten Abschnittindex eingetragenen Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte des ersten Eingabedatensegments entspricht;
– Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen, und dies unter Verwendung des identifizierten ersten Satzes von Verzeichnissen,
ein zweites Eingabedatensegment zu verarbeiten, umfassend:
– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen,
– Identifizieren eines zweiten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des zweiten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte des zweiten Eingabedatensegments entspricht,
– Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen, und dies unter Verwendung des identifizierten zweiten Satzes von Verzeichnissen und wenigstens eines ausgewählten priorisierten Verzeichnisses des ersten Satzes von Verzeichnissen, wobei die ...



(56) Ermittelter Stand der Technik:

DE	**11 2007 003 645**	**T5**
DE	**11 2007 003 678**	**T5**
US	**2010 / 0 198 832**	**A1**
US	**2010 / 0 235 372**	**A1**
US	**5 990 810**	**A**
US	**6 122 626**	**A**

BIGELOW, S. J.: Data Deduplication Explained. 2007. Storage Magazine. URL: http://cdn.ttgtmedia.com/searchStorage/downloads/StorageExtra_DataDupe.pdf [abgerufen am 03.06.2013]

ComputerWeekly.com: How data deduplication works. 06.11.2007. URL: http://www.computerweekly.com/feature/How-data-deduplication-works [abgerufen am 03.06.2013]

FREEMAN, L.: Looking Beyond the Hype: Evaluating Data Deduplication Solutions. Netapp White Paper. 2007. URL: http://storage-brain.com/wp-content/uploads/papers/Evaluating_Data_Deduplication_Solutions.pdf [abgerufen am 04.06.2013]

LLEWELLYN, M.: COP 3530C - Computer Science III (Vorlesungsmanuskript). 2002. URL: http://www.cs.ucf.edu/courses/cop3530-su02/ [abgerufen am 03.06.2013]

MANEGOLD, S.; BONCZ, P., KERSTEN, M.: Databases Techniek. 2005. URL: http://homepages.cwi.nl/~manegold/teaching/DBtech/, Archiviert in http://www.archive.org am 16.09.2005 [abgerufen am 03.06.2013]

ZAIANE, O. R.; KOPERSKI, K.: CMPT 354 Database Systems and Structures (Vorlesungsmanuskript). 1998. URL: http://www.cs.sfu.ca/CourseCentral/354/zaiane/ [abgerufen am 03.06.2013]

ZHU, B.; LI, K.; PATTERSON, H.: Avoiding the disk bottleneck in the data domain deduplication file system. In: Proceedings of the 6th USENIX Conference on File and Storage Technologies, 29.02.2008, S. 269-282. http://usenix.org/events/fast08/tech/full_papers/zhu/zhu.pdf [abgerufen am 03.06.2013]

Beschreibung

HINTERGRUND DER ERFINDUNG

[0001] Daten, die auf einem primären Datenträger gehalten werden, können auch auf einen sekundären Datenträger gesichert (engl. backed-up) werden. Der sekundäre Datenträger kann sich an einem anderen Ort als der primäre Datenträger befinden. Sollte ein auch nur teilweiser Datenverlust auf dem primären Datenträger auftreten, können die Daten mittels des sekundären Datenträgers wiederhergestellt werden. Der sekundäre Datenträger kann auch eine zeitliche Entwicklung der auf dem primären Datenträger gespeicherten Daten über einen Zeitraum enthalten. Auf Anfrage durch einen Benutzer kann der sekundäre Datenträger dem Benutzer die Daten bereitstellen, die auf dem primären Datenträger zu einem bestimmten Zeitpunkt gespeichert waren.

[0002] Datensicherungsvorgänge können wöchentlich, täglich, stündlich oder in anderen Intervallen ausgeführt werden. Die Daten können inkrementell gesichert werden, wobei nur die Änderungen, die an den Daten auf dem primären Datenträger seit der letzten Sicherung vorgenommen wurden, auf den sekundären Datenträger übertragen werden. Es können auch Vollsicherungen ausgeführt werden, bei denen der gesamte Inhalt des primären Datenträgers auf den sekundären Datenträger kopiert wird. Es existieren viele weitere Sicherungsstrategien.

[0003] Beim Erstellen von Sicherungskopien von Daten kann ein bestimmter Teil der zu sichernden Daten bereits vorher auf dem primären Datenträger gespeichert sein, was insbesondere dann der Fall sein kann, wenn Vollsicherungen ausgeführt werden. Das mehrfache Speichern derselben Daten stellt eine ineffiziente Nutzung eines Datenträgers dar.

[0004] Backup-und-Recovery-Systeme sind etwa aus den nachveröffentlichten Druckschriften DE 11 2007 003 678 T5 und DE 11 2007 003 645 T5 bekannt. Diese beschreiben Mechanismen zur effizienten Speicherung von Eingabedatensegmenten mit jeweils mehreren Eingabedatenabschnitten. Vor der Speicherung von Eingabedatenabschnitten wird zunächst geprüft, ob und welche diesen entsprechende Musterdatenabschnitte schon in einem Abschnittspeicher gespeichert sind. Hierzu wird ein dünn besetzter Abschnittindex nach Einträgen entsprechender Musterdatenabschnitte geprüft. Enthält der dünn besetzte Abschnittindex einen Eintrag eines Musterdatenabschnitts, der einem der Eingabedatenabschnitte des Eingabedatensegments entspricht, so wird ein Verzeichnissegment, welches Meta-Daten zu dem im Abschnittindex eingetragenen Musterdatenabschnitt enthält, auf weitere Verweise zu weiteren Musterdatenabschnitten untersucht, die weiteren Eingabedatenabschnitten des Eingabedatensegments entsprechen.

[0005] Aus der US 5990810 A ist ein Datei-System mit geringer Redundanz (”lowredundancy file system”) bekannt. Hierbei werden Dateien in sog. Sub-Blöcke unterteilt. Gleiche Sub-Blöcke werden jeweils nur einmal auf der Festplatte gespeichert. Hierzu wird ein Verzeichnis in Form einer indizierten Hash-Tabelle verwendet, welche auf die gespeicherten Sub-Blöcke verweist. Der Index der Hash-Tabelle kann bspw. in Form von Teilwerten kryptografischer Hash-Werte der referenzierten Sub-Blöcke erfolgen. Beim Schreiben einer Datei werden die entsprechenden zu schreibenden Sub-Blöcke gehasht und in der indizierten Verzeichnis-Hash-Tabelle nachgeschlagen. Ist ein Sub-Block in der Hash-Tabelle eingetragen, so wird er nicht nochmals geschrieben. Ist der in der Hash-Tabelle nicht eingetragen, so wird er geschrieben und die Hash-Tabelle entsprechend aktualisiert. Für Lesezugriffe steht zudem eine Datei-Tabelle zur Verfügung, die angibt, welche Indexe von Sub-Blöcken zu welcher Datei gehören.

[0006] Weitere Ansätze zur ent-duplizierten Datenhaltung werden in folgenden Veröffentlichungen beschrieben:

– Zhu, B.; Li, K.; Patterson, H.: Avoiding the disk bottleneck in the data domain deduplication file system. In: Proceedings of the 6th USENIX Conference an File and Storatge Technologies, 29.02.2008, S. 269–282;

– ComputerWeekly.com: How data dedpulication works. 06.11.2007.

– Bigelow, S. J.: Data Deduplication Explained. 2007. Storage Magazine.

– Freeman, L.: Looking Beyond the Hype: Evaluating Data Deduplication Solutions. Netapp White Paper. 2007.

[0007] Zudem ist aus dem Vorlesungsskript ”CMPT 354 Database Systems and Structures” der Simon Fraser University aus dem Sommersemester 1998 eine Teil-Indizierung von Tabellen bekannt. Diese benötigt weniger Speicherplatz als eine Voll-Indizierung. Jedoch ist nach dem Einspruch in eine teil-indizierte Tabelle ein weiteres Durchsuchen selbiger notwendig, bis ein nicht-inzidierter Tabelleneintrag gefunden wird.

ZUSAMMENFASSUNG DER ERFINDUNG

[0008] Dementsprechend stellt die vorliegende Erfindung eine Datenverarbeitungsvorrichtung zum entduplizierten Speichern von Eingabedaten gemäß Patentanspruch 1 bereit.

[0009] Die vorliegende Erfindung stellt außerdem ein Verfahren zum entduplizierten Speichern von Eingabedaten nach Patentanspruch 11 bereit.

[0010] Weitere Ausgestaltungen ergeben sich aus den abhängigen Patentansprüchen.

KURZE BESCHREIBUNG DER ZEICHNUNG

[0011] Es werden nun Ausführungsformen der Erfindung beschrieben, wobei diese nur beispielhaft und mit Bezug auf die beigefügte Zeichnung beschrieben sind, in der:

[0012] Fig. 1 eine schematische Darstellung eines Datensegments zeigt;

[0013] Fig. 2 eine schematische Darstellung einer Datenverarbeitungsvorrichtung zeigt, die eine Ausführungsform der vorliegenden Erfindung ist;

[0014] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, wenigstens eines der priorisierten Verzeichnisse auszuwählen, um Musterdatenabschnitte zu identifizieren, die den weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen.

[0015] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, den identifizierten ersten Satz von Verzeichnissen und das wenigstens eine Verzeichnis, das beim Verarbeiten vorhergehender Daten identifiziert wurde, in nachfolgenden Operationen neu zu priorisieren.

[0016] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, jedes ausgewählte priorisierte Verzeichnis in absteigender Reihenfolge seiner Priorisierung zu verarbeiten, bis eine vorgegebene Bedingung zutrifft.

[0017] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, jedes ausgewählte priorisierte Verzeichnis in absteigender Reihenfolge seiner Priorisierung zu verarbeiten, bis eine vorgegebene Bedingung für das gerade in Verarbeitung befindliche Verzeichnis zutrifft.

[0018] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, einen Eingabedatenabschnitt im Abschnittspeicher als Musterdatenabschnitt zu speichern, falls die Vorrichtung nicht feststellen kann, dass ein Musterdatenabschnitt, der diesem Eingabedatenabschnitt entspricht, im Abschnittspeicher vorhanden ist.

[0019] Die vorliegende Erfindung stellt außerdem eine Datenverarbeitungsvorrichtung bereit, die Folgendes umfasst:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Daten repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst,
einen dünn besetzten Abschnittindex, der Information über nur einige Musterdatenabschnitte enthält,
wobei die Verarbeitungsvorrichtung dafür eingerichtet ist,
Eingabedaten in mehrere Eingabedatensegmente zu verarbeiten, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
einen ersten Satz von Verzeichnissen zu identifizieren, wobei jedes Verzeichnis des ersten Satzes Verweise auf Musterdatenabschnitte hat, die Eingabedatenabschnitten eines ersten Eingabedatensegments entsprechen und über die Information im dünn besetzten Abschnittindex enthalten ist;
Verzeichnisse abzurufen, die Verweise auf Musterdatenabschnitte haben, die wenigstens einem Eingabedatenabschnitt zuvor verarbeiteter Daten entsprechen;
die identifizierten und abgerufenen Verzeichnisse zu verwenden, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen.

[0020] In einer Ausführungsform ist der wenigstens eine Eingabedatenabschnitt aus zuvor verarbeiteten Daten ein vorgegebener Teil von Eingabedatenabschnitten eines Eingabedatensegments aus zuvor verarbeiteten Daten.

[0021] In einer Ausführungsform geht das Eingabedatensegment zuvor verarbeiteter Daten dem ersten Eingabedatensegment in den Eingabedaten unmittelbar voraus.

[0022] In einer Ausführungsform umfasst der vorgegebene Teil eines Eingabedatensegments zuvor verarbeiteter Daten die Eingabedatenabschnitte, die dem ersten Eingabedatensegment in den Eingabedaten unmittelbar vorausgehen.

[0023] Die vorliegende Erfindung stellt außerdem einen Datenprozessor bereit, der für Folgendes eingerichtet ist:
Verarbeiten von Eingabedaten in Eingabedatenabschnitte, wobei die Eingabedatenabschnitte in Eingabedatensegmenten angeordnet sind;
für ein gegebenes Eingabedatensegment: Auswählen wenigstens einiger der Eingabedatenabschnitte des Eingabedatensegments, die eine vorgegebene Eigenschaft haben,
Zusammenstellen einer Liste potentieller Verzeichnisse aus einem Verzeichnisspeicher, wobei die Liste Folgendes umfasst:
wenigstens ein Verzeichnis, das einen Verweis auf einen Musterdatenabschnitt hat, der wenigstens einem der ausgewählten Eingabedatenabschnitte entspricht; und

wenigstens ein Verzeichnis, das bei der Verarbeitung wenigstens eines weiteren Segments von Eingabedaten identifiziert wurde; und
Priorisieren und Verarbeiten der potentiellen Verzeichnisse, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des gerade in Verarbeitung befindlichen Eingabedatensegments entsprechen.

[0024] Die vorliegende Erfindung stellt außerdem ein Verfahren zur Verarbeitung von Daten bereit, das Folgendes verwendet:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Daten repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst; und
einen dünn besetzten Abschnittindex, der Information über nur einige Musterdatenabschnitte enthält,
wobei das Verfahren Folgendes umfasst:
Verarbeiten von Eingabedaten in mehrere Eingabedatensegmente, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
Identifizieren eines ersten Satzes von Verzeichnissen, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte eines ersten Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist; und
Verwenden des identifizierten ersten Satzes von Verzeichnissen und wenigstens eines Verzeichnisses, das beim Verarbeiten vorhergehender Eingabedaten identifiziert wurde, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen.

[0025] In einer Ausführungsform umfasst das Verfahren Folgendes:
Vergleichen der Eingabedatenabschnitte des ersten Eingabedatensegments mit den Musterdatenabschnitten, auf die wenigstens eines aus dem identifizierten ersten Satz von Verzeichnissen oder wenigstens ein Verzeichnis, das beim Verarbeiten vorhergehender Eingabedaten identifiziert wurde, verweisen.

[0026] In einer Ausführungsform umfasst das Verfahren Folgendes: Priorisieren des identifizierten ersten Satzes von Verzeichnissen und des wenigstens eine Verzeichnisses, das beim Verarbeiten vorhergehender Daten identifiziert wurde, basierend auf vorgegebenen Kriterien.

[0027] Die vorliegende Erfindung stellt außerdem eine Datenverarbeitungsvorrichtung bereit, die Folgendes umfasst:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Daten repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst,
einen dünn besetzten Abschnittindex, der Information über nur einige Musterdatenabschnitte enthält,
wobei die Verarbeitungsvorrichtung für Folgendes eingerichtet ist:
für ein erstes Eingabedatensegment, Identifizieren von Verzeichnissen, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Eingabedatenabschnitte des ersten Eingabedatensegments entspricht, und über den Information im dünn besetzten Abschnittindex enthalten ist;
Verwenden wenigstens eines der identifizierten Verzeichnisse bei der Verarbeitung eines zweiten Eingabedatensegments, um Musterdatenabschnitte zu identifizieren, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen.

KURZE BESCHREIBUNG DER ZEICHNUNG

[0028] Es werden nun Ausführungsformen der Erfindung beschrieben, wobei diese nur beispielhaft und mit Bezug auf die beigefügte Zeichnung beschrieben sind, in der:

[0029] **Fig.** 1 eine schematische Darstellung eines Datensegments zeigt;

[0030] **Fig.** 2 eine schematische Darstellung einer Datenverarbeitungsvorrichtung zeigt, die eine Ausführungsform der vorliegenden Erfindung ist;

[0031] **Fig.** 3 eine schematische Darstellung der Datenverarbeitungsvorrichtung aus **Fig.** 2 im Einsatz zeigt;

[0032] **Fig.** 4 eine schematische Darstellung zweier weiterer Datensegmente zeigt;

[0033] **Fig.** 5 eine schematische Darstellung der Datensegmente aus den **Fig.** 1 und **Fig.** 4 zeigt;

[0034] **Fig.** 6 ein Flussdiagramm eines Verfahrens zeigt, das eine Ausführungsform der vorliegenden Erfindung ist;

DETAILLIERTE BESCHREIBUNG

[0035] **Fig.** 1 zeigt eine schematische Darstellung eines Datensegments **1**. Ein Datensegment **1** kann kürzer oder länger als das in **Fig.** 1 dargestellte sein. Ein Datensegment **1** enthält eine Datenmenge, die in der Größenordnung von 10 Byte, 1000 Byte, 10 KB oder vielen Megabyte oder Terabyte liegen kann. Ein Datensegment kann wenigstens einen Teil der Daten für einen gegebenen Sicherungsvorgang repräsentiert

sentieren. Ein Datensegment kann eines von vielen in einem Datensatz sein.

[0036] Ein Sicherungsdatensatz kann einen kontinuierlichen Datenstrom oder einen diskontinuierlichen Datenstrom umfassen. Unabhängig davon kann der Datensatz verschiedene einzelne Dateien oder Teile von Dateien enthalten. Der Datensatz muss nicht in die einzelnen Dateien, die er enthält, aufgeteilt sein. Der Datensatz kann eingebettete Informationen enthalten, die Verweise auf die Grenzen der einzelnen im Datensatz enthaltenen Dateien umfassen. Der Datensatz kann dann gegebenenfalls leichter in seine ihn bildenden Komponenten zerlegt werden. Die Größe der eingebetteten Informationen kann einen erheblichen Anteil der gesamten Daten darstellen. Das Sichern von Daten mit eingebetteten Dateiinformationen erhöht die erforderliche Kapazität des Datenträgers.

[0037] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, ist dafür eingerichtet, Eingabedaten in einen oder mehrere Eingabedatenabschnitte zu verarbeiten. Ein Eingabedatensatz kann in mehrere, wie oben dargestellte Eingabedatensegmente aufgeteilt werden. Von den Segmenten wird jedes in mehrere Eingabedatenabschnitte aufgeteilt. Jeder Eingabedatenabschnitt kann eine einzelne Datei, einen Teil einer einzelnen Datei, eine Gruppe einzelner Dateien innerhalb des Eingabedatensatzes oder mehrere einzelne Dateien und zusätzlich Teile mehrerer anderer repräsentieren. Die Verarbeitung des Datensatz in Eingabedatenabschnitte kann auf Eigenschaften der Eingabedaten als Ganzes basieren, ohne oder mit geringer Berücksichtigung der einzelnen dann enthaltenen Dateien. Die Grenzen der Datenabschnitte können mit Dateigrenzen zusammenfallen oder auch nicht. Die Datenabschnitte können von gleicher Größe sein oder sie können in der Größe variieren. Analog kann der Datensatz basierend auf Eigenschaften des Eingabedatensatzes als Ganzes oder basierend auf Eigenschaften und/oder der Zahl der Eingabedatenabschnitte in Segmente verarbeitet werden. Die Segmente können ebenfalls von gleicher Größe sein oder sie können in der Größe variieren. Der Datensatz kann zuerst segmentiert werden, bevor jedes Segment anschließend in Datenabschnitte verarbeitet wird, oder umgekehrt.

[0038] **Fig.** 1 zeigt eine schematische Darstellung eines Eingabedatensegments **1**, das in Eingabedatenabschnitte **2** verarbeitet ist. Aus Gründen der Zweckmäßigkeit ist jeder eindeutige Eingabedatenabschnitt in **Fig.** 1 von A bis H bezeichnet. Für den Zweck dieser Anwendung, wird zur Bestimmung, ob ein Abschnitt eindeutig ist nur sein Inhalt berücksichtigt (das heißt die Bytefolge in dem Datensegment **1** aus dem er erzeugt wurde) und nicht seine Position oder Stellung in einem Eingabedatensegment oder Datensatz. Es ist zu bemerken, dass für dieses Beispiel der erste und vierte Abschnitt im Datensegment **1** denselben Inhalt haben und ihnen deshalb dieselbe Bezeichnung zugeordnet wird. Duplizierte Daten im Datensegment **1** können zu wiederholten Eingabeabschnitten **2** führen. Das Eingabedatensegment **1** kann in mehr oder andere Eingabedatenabschnitte **2**, als die in **Fig.** 1 dargestellten, aufgeteilt werden. Ein Eingabedatensegment **1** kann eine Größe von vielen Terabyte haben und in Milliarden von Eingabedatenabschnitten verarbeitet werden. Dem Fachmann sind spezielle Verfahren bekannt, um zu bestimmen, wie das Eingabedatensegment **1** in Eingabedatenabschnitte **2** verarbeitet wird und welche Information jeder Eingabedatenabschnitt **2** enthält.

[0039] **Fig.** 2 zeigt die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist. Die Datenverarbeitungsvorrichtung **3** umfasst einen Abschnittspeicher **4**, einen Verzeichnisspeicher **5** und einen dünn besetzten Abschnittindex **8**. Der Verzeichnisspeicher **5** kann separat, und getrennt, vom Abschnittspeicher **4** sein, es können jedoch auch beide Speicher **4**, **5** sich auf einem gemeinsamen Datenträger oder Speichergerät befinden. Bei dem in **Fig.** 2 dargestellten Beispiel enthalten der Abschnittspeicher **4**, der Verzeichnisspeicher **5** und der dünn besetzte Abschnittindex **8** noch keine Daten oder Information. Es wird nun beschrieben, wie die Daten und die Information jeweils in den Abschnittspeicher **4**, den Verzeichnisspeicher **5** und den dünn besetzten Abschnittindex **8** gefüllt werden.

[0040] Bei der Verarbeitung eines Eingabedatensegments **1** durch die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, wird jeder Eingabedatenabschnitt **2** im Abschnittspeicher **4** als ein Musterdatenabschnitt **6** gespeichert falls die Datenverarbeitungsvorrichtung **3** feststellt, dass ein entsprechender Musterdatenabschnitt **6** nicht bereits vorhanden ist. **Fig.** 3 zeigt eine schematische Darstellung der Datenverarbeitungsvorrichtung **3** nachdem diese das Eingabedatensegment aus **Fig.** 1 verarbeitet hat. Es ist in diesem Beispiel zu erkennen, dass, da das Eingabedatensegment **1** das Erste zu verarbeitende ist, alle eindeutigen Eingabedatenabschnitte dem Abschnittspeicher **4** als Musterdatenabschnitte **6** hinzugefügt werden (das heißt von jedem der B, C, D, E, F, G und H einer, jedoch nur eines der beiden Auftreten von Eingabedatenabschnitt A). In nachfolgenden Vorgängen kann jedoch festgestellt werden, dass ein Eingabedatenabschnitt identisch zu einem Musterdatenabschnitt **6** ist, der bereits im Abschnittspeicher **4** gespeichert ist; wobei in diesem Fall keine neuen Hinzufügungen zum Abschnittspeicher **4** vorgenommen werden. Dies ist das Prinzip der Entduplizierung.

[0041] Ein Musterdatenabschnitt **6** kann ein Duplikat eines Eingabedatenabschnitts **2** sein. Alternativ kann

ein Musterdatenabschnitt **6** eine transformierte Kopie des entsprechenden Eingabedatenabschnitts **2** sein; beispielsweise kann er eine verschlüsselte oder komprimierte Version des Eingabedatenabschnitts **2** sein oder es können ihm zusätzliche Header oder Metadaten beigefügt werden. Ein Eingabedatenabschnitt **2** und ein Musterdatenabschnitt **6** können als einander entsprechend angenommen werden, wenn sie denselben Inhalt enthalten. (Der Inhalt eines verschlüsselten Abschnitts sind dabei die entsprechenden unverschlüsselten Daten.)

[0042] Es ist zu bemerken, dass obwohl zwei Eingabedatenabschnitte mit dem Inhalt A vorhanden sind (der erste und der vierte), in **Fig.** 3 nur ein Musterdatenabschnitt **6** mit dem Inhalt A als im Abschnittspeicher **4** gespeichert dargestellt ist. Dies ist so, da wir in diesem Beispiel angenommen haben, dass die Datenverarbeitungsvorrichtung, wenn sie zum vierten Abschnitt des Eingabedatensegments **1** kommt, feststellt, dass sie bereits einen entsprechenden Musterdatenabschnitt **6** im Abschnittspeicher **4** hat (der hinzugefügt wurde als der erste Abschnitt des Eingabedatensegments **1** verarbeitet wurde). Der Feststellungsvorgang kann gelegentlich Fehler machen, wobei er dann feststellt, dass ein Abschnitt nicht vorhanden ist, wenn er es jedoch tatsächlich ist, was dazu führt, dass einige Musterdatenabschnitte **6** dem Abschnittspeicher **4** mehrfach hinzugefügt werden. Eine gelegentliche Duplizierung kann zugelassen werden. Der Abschnittspeicher **4** kann viele Musterdatenabschnitte **6** speichern.

[0043] In einer Ausführungsform werden sowohl der Abschnittspeicher **4** als auch der Verzeichnisspeicher **5** in nicht-flüchtigem Speicher mit langer Latenzzeit, wie etwa einer Festplatte, gespeichert. Der dünn besetzte Abschnittindex **8** kann in flüchtigem Speicher mit kurzer Latenzzeit, wie etwa RAM, gespeichert werden.

[0044] Wenn ein Eingabedatenabschnitt **2** verarbeitet wird, wird ein Verzeichnis **7** zusammengestellt. Ein Verzeichnis **7** ist eine Repräsentation eines Eingabedatensegments **1**. Das Verzeichnis **7** umfasst Verweise auf Musterdatenabschnitte **6** im Abschnittspeicher **4**, die den Eingabedatenabschnitten **2** entsprechen, die im Eingabedatensegment **1** enthalten sind. Somit können die Verweise des Verzeichnisses **7** als Metadaten für Musterdatenabschnitte **6** angesehen werden. Falls die Verweise auf Musterdatenabschnitte **6** eines gegebenen Verzeichnisses **7** von geringerer Größe sind als die Musterdatenabschnitte **6**, auf die im Verzeichnis **7** verwiesen wird, dann ist zu erkennen, dass ein Verzeichnis **7** von geringerer Größe sein kann als das Eingabedatensegment **1**, das es repräsentiert. Werden dem Verzeichnisspeicher **5** immer weitere Verzeichnisse hinzugefügt, wobei diese Verzeichnisse auf Musterdatenabschnitte verweisen, die bereits im Abschnittspeicher **4** gespeichert sind, dann kann die kombinierte Gesamtgröße der Verzeichnisse und Musterdatenabschnitte kleiner sein als die kombinierte Gesamtgröße der Datensegmente, die die Verzeichnisse repräsentieren; dies ist so, da duplizierte Eingabeabschnitte gegebenenfalls nur jeweils einmal gespeichert werden.

[0045] Wenn ein Eingabedatensegment **1** in Eingabedatenabschnitte **2** verarbeitet und ein Verzeichnis **7**, das das Eingabedatensegment **1** repräsentiert, zusammengestellt wurde, dann wird das Verzeichnis **7** im Verzeichnisspeicher **5** gespeichert, wie dies schematisch in **Fig.** 3 dargestellt ist. Dort wird ein Verweis auf einen Musterdatenabschnitt **6** mit Inhalt X (es wird für gewöhnlich nur einen geben) unter Verwendung des entsprechenden Kleinbuchstabens x dargestellt. Die die Buchstaben umgebenden Kreise werden weiter unten beschrieben.

[0046] Falls ein Benutzer der Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, die Daten eines gegebenen Eingabedatensegments **1** wiederherstellen will – das zu einem Backup gehören kann, das zu einem bestimmten Zeitpunkt erstellt wurde – liest die Vorrichtung das entsprechende Verzeichnis **7** aus dem Verzeichnisspeicher **5**. Es wird dann jeder Verweis im Verzeichnis **7** auf Musterdatenabschnitte **6** im Abschnittspeicher **4** verwendet, um das originale Datensegment **1** zu rekonstruieren.

BEFÜLLEN DES DÜNN BESETZTEN ABSCHNITTINDEX: ABSCHNITTKENNUNGEN

[0047] Bei der Verarbeitung jedes der Eingabedatenabschnitte **2** kann der dünn besetzte Abschnittindex **8** mit Information über lediglich einige der Musterdatenabschnitte **6**, die Eingabedatenabschnitten **2** entsprechen, befüllt werden. In einer Ausführungsform können die ‚einigen' Musterdatenabschnitte danach ausgewählt werden, ob sie eine vorgegebene Eigenschaft haben. Für eine gegebene Zahl von Musterdatenabschnitten im Abschnittspeicher könnte gegebenenfalls im dünn besetzten Abschnittindex Information mit Bezug auf nur einige davon sein, die eine vorgegebene Eigenschaft haben. In einer weiteren Ausführungsform gilt, dass wenn keiner dieser Musterdatenabschnitte **6** die vorgegebene Eigenschaft hat, dem dünn besetzten Abschnittindex **8** keine Information hinzugefügt wird. Das ‚dünne Besetztsein' des dünn besetzten Abschnittindex **8** resultiert daraus, dass der Index Information über nur einige Musterdatenabschnitte **6** enthält (in einer Ausführungsform diejenigen, die die vorgegebene Eigenschaft haben) und keine Information über andere Musterdatenabschnitte **6** (in einer Ausführungsform diejenigen, die die vorgegebene Eigenschaft nicht haben). Für eine gegebene Zahl von im Abschnittspeicher **4** gespeicherten Musterdatenabschnitten **6**, wird es normalerweise eine kleinere Zahl von Musterdatenabschnitten

6 geben, über die der dünn besetzte Abschnittindex **8** Information enthält.

[0048] In einer Ausführungsform ist die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, dafür eingerichtet, eine Abschnittkennung für einen Eingabedatenabschnitt zu erzeugen. Eine Abschnittkennung kann ein digitaler Fingerabdruck des Datenabschnitts sein, auf den er sich bezieht. Die Abschnittkennung kann eine eindeutige Abschnittkennung sein, die für einen bestimmten Datenabschnitt eindeutig ist. Der Algorithmus zum Erzeugen von Abschnittkennungen kann so gewählt werden, dass er dazu fähig ist, für eine vorgegebene Zahl von Datenabschnitten eindeutige Abschnittkennungen zu erzeugen. In einer Ausführungsform wird die Abschnittkennung unter Verwendung des SHA1 Hash-Algorithmus erzeugt. Es können auch andere Hash-Algorithmen wie etwa SHA2 oder MDA5 verwendet werden. In einer Ausführungsform wird der Hash-Algorithmus so gewählt und konfiguriert, dass es im Wesentlichen numerisch unmöglich ist, zwei verschiedene Musterdatenabschnitte zu finden, die dieselbe Abschnittkennung erzeugen würden. Bei einer gegebenen Zahl von Musterdatenabschnitten **6**, die in der Praxis aufgrund von Größenbegrenzungen des Abschnittspeichers **4** dem Abschnittspeicher **4** gegebenenfalls hinzugefügt werden können, kann es extrem unwahrscheinlich sein, dass zwei der hinzugefügten Abschnitte **6** dieselbe Abschnittkennung teilen.

[0049] In einer Ausführungsform ist die Abschnittkennung eines Eingabedatenabschnitts dieselbe wie die Abschnittkennung des entsprechenden Musterdatenabschnitts **6**. Dies kann erreicht werden, indem die Abschnittkennung nur vom Inhalt des gegebenen Abschnitts abhängt. In einer Ausführungsform enthält der dünn besetzte Abschnittindex **8** nur Information über die Musterdatenabschnitte **6**, die eine Abschnittkennung mit einer vorgegebenen Eigenschaft haben. In einem Beispiel kann die vorgegebene Eigenschaft sein, dass N benachbarte Bits der Abschnittkennung einen vorgegebene Wert haben.

[0050] In einer Ausführungsform wird der Algorithmus zum Erzeugen von Abschnittkennungen so gewählt, dass er eine eindeutige Abschnittkennung für jeden möglichen Musterdatenabschnitt erzeugt, für den es wahrscheinlich ist, dass er dem Abschnittspeicher **4** hinzugefügt wird. So sollte eine 4-Bit Abschnittkennung, die nur 16 mögliche Werte hat, nicht dort gewählt werden, wo wahrscheinlich mehr als 16 eindeutige Musterdatenabschnitte dem Abschnittspeicher **4** hinzugefügt werden. Andernfalls könnte zwei verschiedenen Musterdatenabschnitten dieselbe Abschnittkennung zugeordnet werden. In einer Ausführungsform ist die Zahl der möglichen Abschnittkennungswerte viel größer als die wahrscheinliche Zahl eindeutiger Musterdatenabschnitte, die im Abschnittspeicher **4** gespeichert werden sollen. In dieser Ausführungsform kann das Risiko einer Kollision (bei der dieselbe Abschnittkennung von zwei verschiedenen Musterdatenabschnitten erzeugt wird) verringert sein.

[0051] Die oben genannte Ausführungsform trifft eine Auswahl der Musterdatenabschnitte **6** gestützt auf eine Eigenschaft von deren Abschnittkennungen und nicht direkt auf den Musterdatenabschnitten **6** selber. Bei einer Ausführungsform, bei der die Abschnittkennungen nur vom Inhalt des gegebenen Abschnitts abhängen, bedeutet dies, dass die Position des Musterdatenabschnitts **6** im Abschnittspeicher **4** oder die Reihenfolge, in der die Musterdatenabschnitte **6** dem Abschnittspeicher hinzugefügt wurden nicht berücksichtigt werden. Es ist daher reproduzierbar, ob ein gegebener Eingabedatenabschnitt die vorgegebene Eigenschaft hat, unabhängig davon, wo in einem Eingabedatensatz oder -segment er auftritt.

[0052] In weiteren Ausführungsformen kann die vorgegebene Eigenschaft auf der Reihenfolge basieren, in der die Eingabedatenabschnitte verarbeitet werden. Beispielsweise kann dem dünn besetzten Abschnittindex Information bezüglich jedes n-ten verarbeiteten Eingabedatenabschnitts hinzugefügt werden; oder es wird vielmehr Information bezüglich des Musterdatenabschnitts, der dem n-ten Eingabedatenabschnitt entspricht, hinzugefügt.

[0053] In einer weiteren Ausführungsform wird nicht für alle Eingabedatenabschnitte mit der vorgegebenen Eigenschaft Information zum dünn besetzten Abschnittindex **8** hinzugefügt. Basierend auf einem vorgegebenen Auswahlkriterium, kann dem dünn besetzten Abschnittindex **8** Information bezüglich nur einiger der Eingabedatenabschnitte mit einer vorgegebenen Eigenschaft hinzugefügt werden.

[0054] Bei dem in **Fig.** 1 dargestellten beispielhaften Eingabedatensegment **1** haben die beiden Eingabedatenabschnitte B und D die vorgegebene Eigenschaft, was durch einen Kreis gekennzeichnet wird.

[0055] Anschließend an die Verarbeitung der Eingabedatenabschnitte **2** des Eingabedatensegments **1** wurde dem Abschnittspeicher **4** eine Anzahl von Musterdatenabschnitten hinzugefügt (siehe **Fig.** 3). Wie oben bereits bemerkt wurde, kann es in einer Ausführungsform gegebenenfalls nur ein Auftreten jedes eindeutigen Musterdatenabschnitts **6** im Abschnittspeicher **4** geben. Im Fall des Eingabedatensatzes **1** sind dies die Abschnitte A bis H.

[0056] In den Figuren ist es mit einem Kreis gekennzeichnet, wenn ein Abschnitt eine vorgegebene Eigenschaft hat. Dementsprechend haben in **Fig.** 1 zwei der Eingabedatenabschnitte **2** (B und D) eine vorgegebene Eigenschaft. Somit gilt, dass in einer

Ausführungsform Information dem dünn besetzten Abschnittindex **8** hinzugefügt wird, die sich auf die im Abschnittspeicher **4** gespeicherten Musterdatenabschnitte **6** bezieht, die den Eingabedatenabschnitten entsprechen, die eine vorgegebene Eigenschaft haben. Dementsprechend gilt, wie in **Fig.** 3 dargestellt, dass für die Musterdatenabschnitte B und D im dünn besetzten Abschnittindex **8** Einträge vorgenommen werden – in **Fig.** 3 sind die Einträge dabei mit Kleinbuchstaben bezeichnet. Wie oben bemerkt wurde, bedeutet der Kreis um die Bezugszeichen, dass der Musterdatenabschnitt eine vorgegebene Eigenschaft hat.

[0057] In einer weiteren Ausführungsform kann, basierend auf einem vorgegebenen Auswahlkriterium, auch Information, die nur einen der Musterdatenabschnitte B und D betrifft dem dünn besetzten Abschnittindex **8** hinzugefügt werden.

[0058] Im Zusammenhang mit jedem Eintrag im dünn besetzten Abschnittindex **8** für einen bestimmten Musterdatenabschnitt kann gegebenenfalls eine Liste aller Verzeichnisse gespeichert werden, die auf diesen Musterdatenabschnitt verweisen. Da in diesem Beispiel der Verzeichnisspeicher, der Abschnittindex und der Abschnittspeicher erstmalig befüllt werden, hat jeder der Einträge b und d im dünn besetzten Abschnittindex **8** einen einzigen Verweis auf das Verzeichnis **7** im Verzeichnisspeicher **5**. Mit anderen Worten gibt es im Zusammenhang mit dem Eintrag für den Musterdatenabschnitt b im dünn besetzten Abschnittindex **8** einen Registereintrag, dass das Verzeichnis **7** im Verzeichnisspeicher **5** einen Verweis auf diesen Musterdatenabschnitt b enthält. Dasselbe gilt für die Information im dünn besetzten Abschnittindex **8**, die den Musterdatenabschnitt d betrifft.

[0059] In einer Ausführungsform umfasst die im dünn besetzten Abschnittindex **8** enthaltene Information über einen gegebenen Musterdatenabschnitt **6** die Abschnittkennung dieses Musterdatenabschnitts **6**.

[0060] In einer Ausführungsform kann ein Teil der Abschnittkennung implizit im dünn besetzten Abschnittindex **8** gespeichert werden. Dies bedeutet, dass die Position des restlichen Teils der Abschnittkennung implizit den ersten Teil angeben kann.

[0061] Beispielsweise ist es bei Hashtabellen (der dünn besetzte Abschnittindex **8** kann als Hashtabelle implementiert werden) üblich, dass die ersten paar Bit eines Schlüssels angeben, in welchem Slot der Hashtabelle die Information über diesen Schlüssel gespeichert ist; da jeder Eintrag in diesem Slot einen Schlüssel mit denselben ersten paar Bit hat, ist es nicht notwendig, diese Bits explizit zu speichern.

[0062] In einer Ausführungsform kann im dünn besetzten Abschnittindex **8** nur eine partielle Abschnittkennung gespeichert werden, um die Speicheranforderungen zu verringern. Dadurch können zwei verschiedene Musterdatenabschnitte dieselbe partielle Abschnittkennung haben. Ein Nachteil der Speicherung nur partieller Abschnittkennungen ist, dass die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, basierend auf der irreführenden (partiellen) Information im dünn besetzten Abschnittindex **8**, gegebenenfalls Verzeichnisse schlecht auswählt, was zu verschlechterten Entduplizierungen führt (beispielsweise werden duplizierte Kopien eindeutiger Musterdatenabschnitte **6** im Abschnittspeicher **4** vorhanden sein). Die Vorrichtung wird unter Bezug auf die partiellen Abschnittkennungen im Abschnittindex **8** daher gegebenenfalls annehmen, dass ein Musterdatenabschnitt einem gerade verarbeiteten Eingabedatenabschnitt **2** entspricht, obwohl diese tatsächlich verschieden sein können.

[0063] Ausführungsformen der vorliegenden Erfindung können einen Verifikationsschritt enthalten, der später beschrieben wird. Ein solcher Verifikationsschritt kann Musterdatenabschnitte aussortieren, die einem Eingabedatenabschnitt **2** nicht entsprechen, obwohl ihre jeweiligen partiellen Abschnittkennungen anzeigen, dass sie einander entsprechen. Ein Vorteil der Speicherung nur partieller Abschnittkennungen ist, dass die Größe des dünn besetzten Abschnittindex **8** weiter verringert wird. Dieser Vorteil besteht zusätzlich zur verringerten Größe des dünn besetzten Abschnittindex **8** aufgrund der Tatsache, dass Information nur über einige Musterdatenabschnitte **6** gespeichert wird.

[0064] In einer Ausführungsform umfasst die für einen Musterdatenabschnitt **6** im dünn besetzten Abschnittindex **8** enthaltene Information einen Verweis auf jedes Verzeichnis **7**, das sich im Verzeichnisspeicher **5** befindet und das einen Verweis auf diesen Musterdatenabschnitt **6** enthält. Dies bedeutet, dass für jeden Musterdatenabschnitt **6**, der einen Eintrag im dünn besetzten Abschnittindex **8** hat, eine Liste aller Verzeichnisse **7** im Verzeichnisspeicher **5** gespeichert ist, die wenigstens einen Verweis auf diesen Musterdatenabschnitt **6** enthalten. In einer weiteren Ausführungsform kann auch lediglich eine partielle Liste der Verzeichnisse **7** im Verzeichnisspeicher **5** gespeichert sein, die wenigstens einen Verweis auf diesen Musterdatenabschnitt enthalten. Somit gilt, dass obwohl es viele im Verzeichnisspeicher gespeicherte Verzeichnisse geben kann, die einen Verweis auf einen Musterdatenabschnitt **6** enthalten, der einen Eintrag im dünn besetzten Abschnittindex **8** hat, es sein kann, dass der dünn besetzte Abschnittindex **8** lediglich Angaben über eine begrenzte Anzahl dieser Verzeichnisse enthält.

[0065] Im Betrieb kann der Verzeichnisspeicher **5** viele Verzeichnisse **7** enthalten, von denen jedes ein zuvor verarbeitetes Datensegment **1** repräsentiert. In einer Ausführungsform umfasst der Verzeichnisspeicher **5** Information bezüglich jedem darin enthaltenen Verzeichnis **7**. Die Information kann die Eigenschaften umfassen, die jedem Verzeichnis **7** zugeordnet werden; wie etwa seine Größe, die Zahl der Verweise, die es umfasst, oder der Name und andere Angaben zu dem Datensatz, den es repräsentiert. Die Information für ein bestimmtes Verzeichnis kann eine Abschnittkennung für wenigstens einen der Musterdatenabschnitte **6** umfassen, auf den das Verzeichnis **7** verweist.

ENTDUPLIZIERUNG: EINHAKEN

[0066] In einer Ausführungsform verwendet die Datenverarbeitungsvorrichtung **3** den dünn besetzten Abschnittindex **8** und ein identifiziertes Verzeichnis **7** für die Aufgabe zu identifizieren, welche Eingabedatenabschnitte **2** eines gerade in Verarbeitung befindlichen Eingabedatensegments bereits entsprechende Musterdatenabschnitte **6** im Abschnittspeicher **4** haben. Hierdurch kann in einer extremen Ausführungsform gegebenenfalls lediglich eine Kopie jedes eindeutigen Musterdatenabschnitts **6** gespeichert werden, unabhängig davon, wie oft Eingabedatenabschnitte, die diesem Musterdatenabschnitt **6** entsprechen, in den verarbeiteten Eingabedatensegmenten auftreten. Der Vorgang der Beseitigung oder wenigstens Verringerung der mehrfachen Speicherung von Daten wird Entduplizierung genannt (manchmal auch als Verdichtung bezeichnet).

[0067] Das Eingabedatensegment **1** in **Fig.** 1 kann lediglich ein Segment eines größeren Datensatzes umfassen. Wie oben beschrieben, kann ein Datensatz viele Datensegmente umfassen. Dem Fachmann sind zur Segmentierung von Eingabedaten viele Segmentierungsalgorithmen bekannt. In einigen Ausführungsformen können die Grenzen – das heißt die Ausdehnung der Segmente – willkürlich gewählt werden, wobei der Inhalt der Segmente, so wie sie aufgeteilt wurden, wenig oder nicht berücksichtigt wird.

[0068] Fig. 4 stellt einen Teil eines Datensatzes dar, der nachfolgend von der Datenverarbeitungsvorrichtung der vorliegenden Erfindung verarbeitet wird. **Fig.** 4 zeigt einen Teil eines Datensatzes, der in zwei Segmente **11** und **13** segmentiert wurde. Das Eingabedatensegment **11** umfasst die Eingabedatenabschnitte MJKILABCD und das Datensegment **13** umfasst die Eingabedatenabschnitte EFGHAMJKP.

[0069] Bei der Verarbeitung der Eingabedatensegmente **11** und **13** kann die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, jedes der Eingabesegmente hintereinander der Reihe nach verarbeiten. In einer Ausführungsform kann das in **Fig.** 1 dargestellte Eingabedatensegment **1** unmittelbar vor dem Eingabedatensegment **11** aus **Fig.** 4 verarbeitet worden sein. Mit anderen Worten können die Eingabedatensätze **1**, **11** und **13** zusammen aufeinanderfolgende Teile eines größeren Datensatzes bilden.

[0070] Ohne die Verwendung der Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, wird der Inhalt des Eingabedatensegments **11** gegebenenfalls in seiner Gesamtheit gespeichert. Daher würde, obwohl es für den Leser ersichtlich ist, dass sowohl das Eingabedatensegment **1** (jetzt als Verzeichnis **7** im Verzeichnisspeicher **5** gespeichert) und das Eingabedatensegment **11** die gemeinsamen Eingabedatenabschnitte A, B, C und D umfassen, jedes Auftreten der duplizierten Eingabedatenabschnitte im Abschnittspeicher **4** als ein Musterdatenabschnitt gespeichert. Dies kann eine ineffiziente Nutzung des Abschnittspeichers **4** darstellen. Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, kann die Duplizierung von Daten verringern.

[0071] Für die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, gilt, dass mit dem Zuführen des Eingabedatensegments **11** zur Datenverarbeitungsvorrichtung **3**, das Eingabedatensegment **11** in Eingabedatenabschnitte **12** verarbeitet wird. Die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, ist dafür eingerichtet, den dünn besetzten Abschnittindex **8** zu verwenden, um wenigstens ein Verzeichnis **7** im Verzeichnisspeicher **5** zu identifizieren, das wenigstens einen Verweis auf einen Musterdatenabschnitt enthält, der wenigstens einem der Eingabedatenabschnitte **12** des Eingabedatensegments **11** entspricht und über den Information im dünn besetzten Abschnittindex **8** enthalten ist.

[0072] Wie in **Fig.** 3 dargestellt, enthält der Abschnittspeicher **4** (vor der Verarbeitung des Eingabedatensegments **11**) die Musterdatenabschnitte A, B, C, D, E, F, G und H, von denen jeder Eingabedatensegmente **2** repräsentiert, die im Eingabedatensegment **1**, das in **Fig.** 1 dargestellt ist, aufgetreten sind. Zusätzlich umfasst der Verzeichnisspeicher **5** ein Verzeichnis **7**, das das Eingabedatensegment **1** repräsentiert und das Verweise auf jeden der Musterdatenabschnitte **6** umfasst, die im Abschnittspeicher **4** gespeichert sind. Das Eingabedatensegment **1** aus **Fig.** 1 kann unter Verwendung des Verzeichnisses **7** im Verzeichnisspeicher **5** und der Musterdatenabschnitte **6** im Abschnittspeicher **4** wieder aufgebaut werden.

[0073] Zusätzlich gilt, wie oben beschrieben, dass der dünn besetzte Abschnittindex **8** dafür eingerichtet ist, Information über lediglich einige Musterdatenab-

schnitte **6** zu enthalten. In einer Ausführungsform ist der dünn besetzte Abschnittindex dafür eingerichtet nur über diejenigen Musterdatenabschnitte Information zu enthalten, die eine vorgegebene Eigenschaft haben.

[0074] Weiterhin mit Bezug auf **Fig.** 3 ist zu bemerken, dass nur die Musterdatenabschnitte B und D einen Eintrag im dünn besetzten Abschnittindex **8** haben. Keiner der Musterdatenabschnitte A, C, E, F, G oder H hat einen Verweis im dünn besetzten Abschnittindex **8**.

[0075] In einer Ausführungsform kann die Information im dünn besetzten Abschnittindex **8** die Abschnittkennung, oder die partielle Abschnittkennung, der Musterdatenabschnitte B und D umfassen; und außerdem wenigstens eine partielle Liste von Verzeichnissen **7** im Verzeichnisspeicher **5**, die einen Verweis auf die Musterdatenabschnitte B und D enthalten – in einer weiteren Ausführungsform kann auch eine vollständige Verzeichnisliste vorhanden sein. Bei dem in **Fig.** 3 dargestellten Beispiel ist momentan nur ein Verzeichnis **7** im Verzeichnisspeicher **5** enthalten. Somit gilt, dass im dünn besetzten Abschnittindex **8** im Zusammenhang mit jedem Eintrag für die Musterdatenabschnitte B und D ein Verweis auf nur das einzige im Verzeichnisspeicher **5** gespeicherte Verzeichnis **7** gespeichert wird.

[0076] Die Datenverarbeitungsvorrichtung **3** ist dafür eingerichtet, nach dem Empfang des Eingabedatensegments **11** aus **Fig.** 4, das Eingabedatensegment **11** in Eingabedatenabschnitte **12** zu verarbeiten. Für jeden Eingabedatenabschnitt **12** kann eine Abschnittkennung erzeugt werden. In einer Ausführungsform kann jede der erzeugten Abschnittkennungen mit allen Einträgen im dünn besetzten Abschnittindex **8** verglichen werden, um einen Musterdatenabschnitt zu finden, der einem Eingabedatenabschnitt entspricht. Es wird erkannt, dass das Eingabedatensegment **11** die Eingabedatenabschnitte B und D umfasst, die den Musterdatenabschnitten B und D entsprechen. Bei diese Ausführungsform wird somit, nach dem Vergleich jedes Eingabedatenabschnitts im Eingabedatensegment **11** mit den Einträgen im dünn besetzten Index, festgestellt, dass im dünn besetzten Abschnittindex **8** zwei Einträge vorhanden sind, die auf Musterdatenabschnitte verweisen, die den Eingabedatenabschnitten B und D entsprechen.

[0077] In einer weiteren Ausführungsform werden nur diejenigen Eingabedatenabschnitte des Eingabedatensegments **11** mit den Einträgen im dünn besetzten Abschnittindex **8** verglichen, die eine vorgegebene Eigenschaft haben. Das Eingabedatensegment **11** umfasst die beiden Eingabedatenabschnitte B und D, die zufällig beide die vorgegebene Eigenschaft haben, wodurch in diesem Beispiel für beide Eingabedatenabschnitte B und D positive Übereinstimmungen im dünn besetzten Abschnittindex **8** gefunden werden. Diese Ausführungsform kann die Geschwindigkeit erhöhen, mit der die Verarbeitung eines Datensegments ausgeführt wird. Als eine Folge davon, dass der dünn besetzte Abschnittindex nur Einträge für Musterdatenabschnitte enthält, die die vorgegebene Eigenschaft haben, kann es sein, dass es nur einen geringen oder keinen Vorteil bringt, die Eingabedatenabschnitte, die die vorgegebene Eigenschaft nicht haben, mit den Einträgen im dünn besetzten Abschnittindex zu vergleichen. Dies ist so, da keine Übereinstimmung gefunden wird. Daher kann die Einrichtung eines dünn besetzten Abschnittindex, der eine Ausführungsform der vorliegenden Erfindung ist, die Zeit, die es braucht, um ein Eingabedatensegment zu verarbeiten erheblich verringern und, was wesentlich ist, auch den benötigten RAM-Speicher verringern.

[0078] Zusätzlich ist aus **Fig.** 4 ersichtlich, dass der Eingabedatenabschnitt M ebenfalls eine vorgegebene Eigenschaft hat. Daher kann der Eingabedatenabschnitt M ebenfalls mit den Einträgen im dünn besetzten Abschnittindex **8** verglichen werden. Da jedoch im dünn besetzten Abschnittindex kein Eintrag vorhanden ist, der sich auf einen Musterdatenabschnitt M bezieht, wird kein solches Ergebnis zurückgegeben. Später kann ein Eintrag für den Musterdatenabschnitt M zum dünn besetzten Abschnittindex hinzugefügt werden, wie nachfolgend erläutert wird.

[0079] Im vorliegenden Beispiel ist momentan nur ein Verzeichnis **7** im Verzeichnisspeicher **5** gespeichert. In einigen Ausführungsformen der vorliegenden Erfindung kann der Verzeichnisspeicher **5** mehrere Verzeichnisse **7** umfassen. Es ist daher zu erkennen, dass der dünn besetzte Abschnittindex **8** mehrere Einträge umfassen kann, wobei jeder Eintrag auf mehrere verschiedene Verzeichnisse im Verzeichnisspeicher verweist, die auf den Musterdatenabschnitt verweisen, der dem Eintrag entspricht.

[0080] Mit Verweis auf das in **Fig.** 4 dargestellte Beispiel gilt, dass nach der Verarbeitung des Eingabedatensegments **11** die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, das im Verzeichnisspeicher **5** gespeicherte Verzeichnis **7** für nachfolgende Operationen identifiziert („zurückgibt"). Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, ist dafür eingerichtet, das oder die zurückgegebenen Verzeichnisse zu analysieren, um festzustellen welche Eingabedatenabschnitte des Eingabedatensegments **11** bereits im Abschnittspeicher **4** vorhanden sind. In einigen Ausführungsformen wird gegebenenfalls nur eine Untermenge der zurückgegebenen Verzeichnisse derart analysiert. In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, das zurückgegebene wenigs-

tens eine Verzeichnis zu überprüfen und wenigstens einen Verweis auf einen Musterdatenabschnitt **6** zu identifizieren, der wenigstens einem weiteren Eingabedatenabschnitt des Eingabedatensegments **11** entspricht. Beispielsweise könnten Verweise auf Musterdatenabschnitte **6** identifiziert werden, die weiteren Eingabedatenabschnitten **12** des Eingabedatensegments **11** entsprechen.

[0081] In einer Ausführungsform wird jeder Eingabedatenabschnitt **12** des Eingabedatensegments **11** mit jedem Musterdatenabschnitt **6** verglichen, auf den das zurückgegebene Verzeichnis **7** verweist. Somit wird jeder der Eingabedatenabschnitte des Eingabedatensegments **11** „MJKILABCD" mit den Musterdatenabschnitten „ABCADEFGH" verglichen, auf die das zurückgegebene Verzeichnis **7** verweist. Selbstverständlich müssen zwischen den Eingabedatenabschnitten **12** (A und D), die dazu geführt haben, dass das Verzeichnis zurückgegeben wird, keine Vergleiche ausgeführt werden, da bereits bekannt ist, dass sie übereinstimmen.

[0082] In einer Ausführungsform kann die vollständige Abschnittkennung jedes Eingabedatenabschnitts **12** mit den vollständigen Abschnittkennungen jedes der Musterdatenabschnitte **6**, auf die das identifizierte Verzeichnis **7** verweist, verglichen werden. In einer Ausführungsform, die oben beschrieben wurde, kann das Verzeichnis die Abschnittkennung für jeden Musterdatenabschnitt **6** enthalten, auf den das Verzeichnis **7** verweist. Dementsprechend kann der Vergleichsschritt ausgeführt werden, indem nur die Information, die in einem zurückgegebenen Verzeichnis enthalten ist, und die für das Eingabedatensegment **11** erzeugten Abschnittkennungen verwendet werden. Der Vorteil hiervon ist, dass gegebenenfalls keine Notwendigkeit besteht, für weitere Informationen auf den dünn besetzten Abschnittindex **8** oder den Abschnittspeicher **4** zuzugreifen.

[0083] Weiterhin mit Bezug auf **Fig.** 4 gilt, dass durch den Vergleich jedes der Eingabedatenabschnitte **12** des Eingabedatensegments **11** mit dem Musterdatenabschnitt **6**, auf den das zurückgegebene Verzeichnis verweist, festgestellt werden kann, dass die Eingabedatenabschnitte A und C den Musterdatenabschnitten A und C entsprechen, die bereits im Abschnittspeicher **4** gespeichert sind (da das zurückgegebene Verzeichnisse einen Verweis auf sie enthält). In einer Ausführungsform wird festgestellt, dass die Abschnittkennungen der Eingabedatenabschnitte A und C identisch zu den im Verzeichnis enthaltenen Abschnittkennungen sind, wobei das Verzeichnis auf die entsprechenden Musterdatenabschnitte A und C im Abschnittspeicher verweist.

[0084] Auf jeden Fall kann die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, feststellen, dass Musterdatenabschnitte **6**, die den Eingabedatenabschnitten A, B, C und D entsprechen, bereits vorhanden sind. Dementsprechend müssen Musterdatenabschnitte, die den Eingabedatenabschnitten A, B, C und D des Eingabedatensegments **11** entsprechen, nicht von neuem im Abschnittspeicher **4** gespeichert werden. Der Ressourcenbedarf für das Speichern des originalen Eingabedatensatzes **1** und des Eingabedatensegments **11** ist gegebenenfalls niedriger als deren akkumulierte Originalgröße.

[0085] Weiterhin mit Bezug auf das in **Fig.** 4 dargestellte Eingabedatensegment **11** ist zu erkennen, dass das Segment außerdem die Eingabedatenabschnitte M, J, K, I und L umfasst. Es ist zu erkennen, dass für keinen dieser Eingabedatenabschnitte **12** festzustellen ist, dass er irgendeinem Musterdatenabschnitt **6** entspricht, auf den das zurückgegebene Verzeichnis **7** verweist – das heißt der Vergleichsschritt wird keine Übereinstimmungen ergeben.

[0086] Die Eingabedatenabschnitte M, J, K, I und L können deshalb dem Abschnittspeicher als Musterdatenabschnitte hinzugefügt werden.

[0087] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, ist außerdem dafür eingerichtet, für das Eingabedatensegment **11** ein Verzeichnis zusammenzustellen. Wie obenstehend mit Bezug auf das in **Fig.** 4 dargestellte Beispiel ausgeführt, wurde festgestellt, dass der Abschnittspeicher **4** bereits Musterdatenabschnitte A, B, C und D enthält, die den Eingabedatenabschnitten A, B, C und D des Eingabedatensegments **11** entsprechen. Es wird daher ein Verzeichnis für das Eingabedatensegment **11** zusammengestellt, das Verweise auf diese Musterdatenabschnitte **6** enthält. Der Vorteil hiervon ist, dass ein Teil des Verzeichnisses für das Eingabedatensegment **11** bereits zusammengestellt wurde, ohne dass weitere Musterdatenabschnitte dem Abschnittspeicher hinzugefügt wurden. Da die Eingabedatenabschnitte M, J, K, I und L dem Abschnittspeicher hinzugefügt werden, gilt zusätzlich, dass das Verzeichnis für das Eingabedatensegment **11** mit Verweisen auf diese Musterdatenabschnitte zusammengestellt wird. Das neue Verzeichnis kann dann dem Abschnittspeicher **5** hinzugefügt werden.

[0088] Zusätzlich kann dem dünn besetzten Abschnittindex weitere Information hinzugefügt werden. Beispielsweise ist zu erkennen, dass die beiden jetzt im Abschnittspeicher gespeicherten Verzeichnisse auf die Musterdatenabschnitte B und D verweisen. Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, ist dafür eingerichtet, einen Verweis im Zusammenhang mit jedem relevanten Registereintrag im dünn besetzten Abschnittindex auf diese Verzeichnisse hinzuzufügen. Für den Eintrag b (der zum Musterdatenabschnitt B gehört) im dünn besetzten Abschnittindex **8**

wird es somit einen Verweis auf beide Verzeichnisse im Verzeichnisspeicher geben. Dasselbe trifft auf den Eintrag d im dünn besetzten Abschnittindex **8** zu (der zum Musterdatenabschnitt D gehört).

[0089] Außerdem ist zu erkennen, dass der Eingabedatenabschnitt M des Eingabedatensegments **11** die vorgegebene Eigenschaft hat – angezeigt durch einen Kreis. Im dünn besetzten Abschnittindex kann ein neuer Eintrag erstellt werden, der sich auf den Musterdatenabschnitt M bezieht. Der Eintrag kann einen Verweis auf das neu hinzugefügte Verzeichnis im Verzeichnisspeicher **5** enthalten; jedoch nicht auf das andere Verzeichnis, da dieses keinen Verweis auf einen Musterdatenabschnitt enthält, der dem Eingabedatenabschnitt M entspricht.

[0090] Bei der Verarbeitung zukünftiger Eingabedatensegmente, die einen Eingabedatenabschnitt umfassen, der dem Musterdatenabschnitt M entspricht, kann das neu hinzugefügte Verzeichnis identifiziert und in einem Vergleichsschritt mit dem neuen Eingabedatensegment verwendet werden.

[0091] Der Vorteil des Führens eines dünn besetzten Abschnittindex ist, dass weniger Speicherplatz benötigt wird als im Fall, dass ein ‚vollständiger' Abschnittindex gespeichert wurde (also einer, der Information über jeden Musterdatenabschnitt enthält). Doch auch wenn ein dünn besetzter Abschnittindex verwendet wird, kann die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, immer noch dazu fähig sein, in vorhergehenden Verzeichnissen Verweise auf Musterdatenabschnitte zu identifizieren, die Eingabedatenabschnitten eines Eingabedatensegments entsprechen, das gerade verarbeitet wird.

[0092] Mit Bezug auf die **Fig.** 3 und **Fig.** 4 gilt, dass obwohl der dünn besetzte Abschnittindex Information nur über die Musterdatenabschnitte B und D enthält, diese Einträge dennoch dazu führen, dass das erste Verzeichnis identifiziert wird. Passend wurde mit diesem zurückgegebenen Verzeichnis festgestellt, dass das Eingabedatensegment **11** ebenfalls die Eingabedatenabschnitte A und C umfasst, die bereits als Musterdatenabschnitte im Abschnittspeicher vorhanden sind.

[0093] Das Vorgehen der Erfindung auf diese Weise kann als „Einhaken" (engl. Hooking) bezeichnet und konzeptuell so verstanden werden, dass an jedem Verweis auf einen Musterdatenabschnitt über den der dünn besetzte Abschnittindex Information enthält, Haken zu Verzeichnissen befestigt werden. Wenn ein Eingabedatensegment verarbeitet wird, werden somit Verzeichnisse, die auf einen Musterdatenabschnitt verweisen, der einem Eingabedatenabschnitt des Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist, zur Analyse mit „herangezogen". Je mehr „Haken" des gegebenen Verzeichnisses es gibt, die Eingabedatenabschnitten des Eingabedatensegments entsprechen, desto wahrscheinlicher ist es, dass es „herangezogen" wird und dadurch, dass es mehr „Haken" hat, ist es wahrscheinlich, dass das Verzeichnis nützlicher für eine Entduplizierung ist.

[0094] Ein Vorteil der Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, ist es, dass keine vollständige Durchsuchung des Abschnittspeichers **4** für jeden Eingabedatenabschnitt **2** notwendig ist, um festzustellen, ob er bereits als Musterdatenabschnitt **6** gespeichert wurde. Stattdessen kann die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, die Verzeichnisse **7** verwenden, die für zuvor verarbeitete und gespeicherte Datensegmente erzeugt wurden. Die Vorteile der Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, werden überdies deutlich, wenn die zu verarbeitenden Eingabedatensegmente, weitgehend, ähnlich zu zuvor verarbeiteten Datensegmenten sind. Beispielsweise kann zwischen zwei vollständigen Datensicherungsvorgängen nur ein kleiner Anteil der jeweiligen Datensegmente verscheiden sein. Muss systematisch über alle im Abschnittspeicher **4** gespeicherten Musterdatenabschnitte **6** gesucht werden, um Musterdatenabschnitte **6** zu finden, die jedem der Eingabedatenabschnitte eines Eingabedatensegments entsprechen, kann dies ineffizient und zeitaufwendig sein.

[0095] Die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, kann gegebenenfalls den Umstand nutzen, dass jedes verarbeitete Eingabedatensegment **1** ähnlich sein kann. Daher können vorhergehende ähnliche Verzeichnisse dazu verwendet werden, wenigstens einen Teil eines neuen Verzeichnisses für das letzte Eingabedatensegment zusammenzustellen.

[0096] In einer Ausführungsform gilt, dass nachdem das wenigstens eine Verzeichnis durch „Einhaken" identifiziert wurde, die Datenverarbeitungsvorrichtung **3** dafür eingerichtet ist, in diesen Verzeichnissen nach allen weiteren Verweisen auf Musterdatenabschnitte **6** im Abschnittspeicher **4** zu suchen, die weiteren Eingabedatenabschnitten eines Eingabedatensegments entsprechen, das gerade verarbeitet wird. In einer Ausführungsform wird die Suche ausgeführt, indem jeder Eingabedatenabschnitt aus einem Eingabedatensegment der Reihe nach ausgewählt wird – eventuell ausgenommen der Eingabedatenabschnitt, der zur Identifizierung des Verzeichnisses geführt hat – und ihn mit jedem Verweis in dem einen oder den mehreren identifizierten Verzeichnissen zu vergleichen. Falls ein Verweis auf einen entsprechenden Musterdatenabschnitt **6** gefunden wird, wird der entsprechende Eingabedatenabschnitt in ei-

nem neuen Verzeichnis mit einem Verweis auf den Musterdatenabschnitt **6** repräsentiert. Der Suchvorgang kann fortfahren, bis alle Eingabedatenabschnitte mit allen Verweisen in dem oder den identifizierten einen oder mehreren Verzeichnissen verglichen wurden.

[0097] insbesondere dort der Fall sein, wo die ‚alten' Musterdatenabschnitte **6** am Anfang des Abschnittspeichers **4** gespeichert sind und deshalb wahrscheinlich zuerst durchsucht würden.

[0098] Die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, kann andererseits wenigstens ein Verzeichnis **7** im Verzeichnisspeicher **5** identifizieren, das wenigstens einen Verweis auf einen Musterdatenabschnitt **6** enthält, der wenigstens einem Eingabedatenabschnitt **2** entspricht. Ausführungsformen der vorliegenden Erfindung könne daher den Umstand ausnutzen, dass Eingabedatensegmente, die einen bestimmten Eingabedatenabschnitt enthalten, wobei dieser Eingabedatenabschnitt einem Musterdatenabschnitt **6** entspricht, der bereits im Abschnittspeicher **4** vorhanden ist und der einen Eintrag im dünn besetzten Abschnittindex **8** hat, ebenfalls Eingabedatenabschnitte enthalten können, die weiteren Musterdatenabschnitten **6** entsprechen, die bereits im Abschnittspeicher **4** gespeichert sind.

[0099] In einer Ausführungsform der vorliegenden Erfindung gilt, dass nach dem Erzeugen einer Abschnittkennung für einen Eingabedatenabschnitt **2** und dem Identifizieren einer entsprechenden Abschnittkennung im dünn besetzten Abschnittindex **8**, die zu einem Musterdatenabschnitt **6** gehört und im dünn besetzten Abschnittindex **8** gespeichert ist, die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, dafür eingerichtet ist, einen Verifikationsvorgang auszuführen. Der Verifikationsvorgang umfasst den Vergleich des Eingabedatenabschnitts **2** mit dem identifizierten Musterdatenabschnitt **6**, der im Abschnittspeicher **4** gespeichert ist, um zu bestätigen, ob die zwei Datenabschnitte tatsächlich denselben Inhalt haben. Ohne den Verifikationsvorgang und insbesondere dann, wenn partielle Abschnittkennungen verwendet werden, kann es sein, dass ein Musterdatenabschnitt **6**, der als ‚entsprechend' identifiziert wurde, tatsächlich einen anderen Inhalt als der Eingabedatenabschnitt **2** hat. Die Aufnahme eines Verweises auf den nicht-entsprechenden Musterdatenabschnitt **6** würde in das Verzeichnis einen Fehler einführen und die genaue Wiederherstellung von im Verzeichnis repräsentierten Daten verhindern.

[0100] In einer werteren Ausführungsform kann der Verifikationsvorgang durch den Vergleich der Abschnittkennung eines Eingabedatenabschnitts mit einer Abschnittkennung, die in einem identifizierten Verzeichnis enthalten ist, ausgeführt werden. Ein Vorteil hiervon ist, dass gegebenenfalls überhaupt kein Zugriff auf den Abschnittspeicher erforderlich ist. Der Verifikationsvorgang kann ausgeführt werden, indem lediglich die Information verwendet wird, die im Verzeichnis enthalten ist, sowie die Abschnittkennungen, die für die Eingabedatenabschnitte erzeugt werden. Falls partielle Abschnittkennungen im dünn besetzten Abschnittindex **8** gespeichert werden, kann eine Situation eintreten, in der die Abschnittkennung eines Eingabedatenabschnitts mit der partiellen Abschnittkennung eines Musterdatenabschnitts, die im dünn besetzten Abschnittindex **8** gespeichert ist, übereinstimmt, obwohl die zugehörigen Eingabe-/Musterdatenabschnitte nicht miteinander übereinstimmen. Daher kann es sein, dass die Verzeichnisse, die dafür identifiziert wurden, dass sie Verweise auf einen Musterdatenabschnitt enthalten, der einem Eingabedatenabschnitt entspricht, tatsächlich auf keine Musterdatenabschnitte verweisen können, die irgendwelchen Eingabedatenabschnitten entsprechen. In einer Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, für das oder die identifizierten Verzeichnisse einen Verifikationsvorgang auszuführen. In einer Ausführungsform gilt, dass wenn wenigstens ein Verzeichnis identifiziert wurde, die Abschnittkennung verifiziert wird, die in dem oder den Verzeichnissen des Musterdatenabschnitts gespeichert ist, für den angezeigt wurde, dass er einem Eingabedatenabschnitt entspricht. Nur falls die Abschnittkennung identisch zur Abschnittkennung des Eingabedatenabschnitts ist, kann für nachfolgende Vorgänge gegebenenfalls das Verzeichnis verwendet werden. Diese Ausführungsform kann dieselbe Wirkung haben wie das Ausführen des Verifikationsvorgangs durch Lesen vom Abschnittspeicher **4**, sie erfordert jedoch keinen Zugriff auf den Abschnittspeicher **4**. Es ist zu erkennen, dass das zurückgegebene Verzeichnis viel kleiner sein kann als der Abschnittspeicher **4**. Dadurch kann es das Ausführen eines Vergleichsvorgangs unter Verwendung des identifizierten Verzeichnisses anstatt des Abschnittspeichers **4** gestatten, dass wenigstens ein Teil der Daten für den Vergleich verarbeitet wird, während sie im RAM sind.

[0101] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, kann zur Verdichtung von Eingabedatensätzen für die Speicherung, Verschlüsselung oder Übertragung verwendet werden. Beispielsweise können die Eingabedaten Sätze von Sicherungsdaten von einem ersten Datenträger repräsentieren, zur Speicherung auf einem zweiten Datenträger. Die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, vergleicht, wie oben beschrieben, eine Abschnittkennung eines Eingabedatenabschnitts **2** mit den Abschnittkennungen, die in einem dünn besetzten Abschnittindex **8** gespeichert sind. Der Vergleichsschritt kann einen direkten Zugriff auf

die im dünn besetzten Abschnittindex **8** enthaltenen Daten erfordern. In einer Ausführungsform kann der dünn besetzte Abschnittindex **8** in Speicher mit wahlfreiem Zugriff (RAM) gespeichert sein. RAM-Speicher erlaubt einen schnellen und wahlfreien Zugriff auf die darin enthaltenen Informationen. Es kann jedoch die Anforderung geben, den für eine Datenverarbeitungsvorrichtung erforderlichen RAM-Speicher zu verringern. Durch das Bereitstellen eines dünn besetzten Abschnittindex **8**, der im RAM zu speichern ist, erfordert die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, weniger RAM-Speicher als eine Datenverarbeitungsvorrichtung ohne einen dünn besetzten Index.

[0102] Ohne das Bereitstellen eines Abschnittindex vergleicht die Datenverarbeitungsvorrichtung gegebenenfalls einen Eingabedatenabschnitt **2** mit jedem Musterdatenabschnitt **6**, der im Abschnittspeicher **4** gespeichert ist. Da der Abschnittspeicher **4** unter Umständen sehr groß ist, kann es schwierig oder einfach nicht möglich sein, den gesamten Inhalt des Abschnittspeichers **4** im RAM zu speichern. Der Abschnittspeicher **4** kann in nicht-flüchtigem Speicher, wie etwa einer Festplatte, gespeichert werden. Das Lesen von Daten vom Abschnittspeicher **4** erfordert daher einen Festplatten-Lesevorgang. Dies kann erheblich langsamer sein als der Zugriff auf Daten, die im RAM gespeichert sind. Die Datenverarbeitungsvorrichtung **3**, die eine Ausführungsform der vorliegenden Erfindung ist, umfasst einen dünn besetzten Abschnittindex **8**, der sich im RAM-Speicher befinden kann, was einen schnelleren Zugriff auf die darin enthaltenen Informationen gestattet. Hierdurch können im Abschnittspeicher **4** gespeicherte Musterdatenabschnitte **6**, die einem Eingabedatenabschnitt **2** entsprechen, leichter identifiziert werden, ohne dass ein durchgängiger direkter Zugriff auf den Abschnittspeicher **4** erforderlich ist. Es kann, wie oben beschrieben, einen Verifikationsvorgang geben. Dieser Vorgang kann den Zugriff auf einen Musterdatenabschnitt **6** erfordern, der im Abschnittspeicher **4**, auf der Festplatte, gespeichert ist, jedoch erfordert dies gegebenenfalls nur eine Festplattenpositionierung des Abschnittspeichers **4** und das Abrufen eines einzelnen Musterdatenabschnitts **6**.

[0103] Für Ausführungsformen der vorliegenden Erfindung, die einen dünn besetzten Abschnittindex **8** umfassen gilt, dass für einen ersten Eingabedatenabschnitt es einen ersten Musterdatenabschnitt **6** im Abschnittspeicher geben kann, der dem ersten Eingabedatenabschnitt entspricht; jedoch gibt es keinen Eintrag im dünn besetzten Abschnittindex **8**, der sich auf den ersten Musterdatenabschnitt **6** bezieht. Es kann jedoch einen Eintrag im dünn besetzten Abschnittindex **8** für einen zweiten Musterdatenabschnitt **6** geben, der einem zweiten Eingabedatenabschnitt entspricht. Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der Erfindung ist, identifiziert dann das oder die Verzeichnisse, die auf den zweiten Musterdatenabschnitt **6** verweisen. Es kann eine nachfolgende Durchsuchung dieses oder dieser Verzeichnisse ausgeführt werden. Es kann sein, dass das oder die identifizierten Verzeichnisse keine Verweise auf den ersten Musterdatenabschnitt enthalten. Oder es wurden alle Suchvorgänge in dem oder den Verzeichnissen beendet, bevor ein Verweis auf den ersten Musterdatenabschnitt **6** gefunden wurde, obwohl ein Verzeichnis gegebenenfalls einen Verweis auf den ersten Musterdatenabschnitt **6** enthält.

[0104] Es ist möglich, dass die gespeicherten Verzeichnisse, die auf den ersten entsprechenden Musterdatenabschnitt **6** verweisen, nicht auf den zweiten Musterdatenabschnitt **6** verweisen. Wobei in diesem Fall die Datenverarbeitungsvorrichtung, die eine Ausführungsform der Erfindung ist, den ersten Musterdatenabschnitt **6** nicht identifizieren würde, wenn sie die Verzeichnisse analysiert, die den zweiten Musterdatenabschnitt **6** enthalten.

[0105] Daher wird die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, gegebenenfalls den ersten Eingabedatenabschnitt als neuen Musterdatenabschnitt **6** im Abschnittspeicher speichern, obwohl bereits ein Musterdatenabschnitt **6** im Abschnittspeicher **4** vorhanden ist, der dem Eingabedatenabschnitt entspricht.

[0106] Trotzdem können die Vorteile des niedrigeren Bedarfs an RAM-Speicher und dem verringerten Zeitbedarf zur Durchsuchung des dünn besetzten Abschnittindex **8** die Nachteile der zweimaligen Speicherung einiger Eingabedatenabschnitte **2** als Musterdatenabschnitte **6** ausgleichen.

[0107] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, kann dafür eingerichtet sein, Musterdatenabschnitte **6** im Abschnittspeicher **4** zu identifizieren, die wenigstens einigen Eingabedatenabschnitten **2** entsprechen, wobei sie nur einen dünn besetzten Index umfasst. In einem extremen und vielleicht idealen Beispiel könnte es keine doppelten Einträge im Abschnittspeicher **4** geben. Die Datenverarbeitungsvorrichtung **3** mit einem dünn besetzten Abschnittindex **8** kann ebenso oder fast ebenso effizient bei der Verdichtung von Eingabedaten sein wie ein Datenprozessor **3** mit einem vollständigen Abschnittindex **8**. Unter Effizienz wird hier verstanden, dass die im Abschnittspeicher **4** gespeicherten Musterdatenabschnitte **6** nicht dupliziert sind oder wenigstens bis zu einem vorgegebenen Grad nicht dupliziert sind. Ein gewisse Duplizierung von Musterdatenabschnitten kann gestattet sein. Außerdem kann eine gewisse Falschidentifizierung von Verzeichnissen **7** gestattet sein, die einen Verweis auf einen Musterdatenabschnitt **6** umfassen, der einem Eingabedatenabschnitt entspricht. Ausführungsformen der vorliegen-

den Erfindung können entsprechend einer Vorteilsabwägung konfiguriert sein – der abzusehende Nachteil davon, einige Duplizierungen von Daten oder Falschidentifizierungen von Verzeichnissen **7** zu gestatten, wird von der zugehörigen Abnahme der erforderlichen Größe des dünn besetzten Abschnittindex **8** oder der zugehörigen Zunahme in der Effizienz der Vorrichtung als Ganzes ausgeglichen oder übertroffen.

[0108] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, wird nun mit Bezug auf **Fig.** 4 beschrieben.

[0109] Wie oben angesprochen, zeigt **Fig.** 4 einen Teil eines Datensatzes der die Eingabedatensegmente **11** und **13** umfasst. So wie oben beschrieben, wurde das Eingabedatensegment **11** verarbeitet und es wurde dem Abschnittspeicher **4** ein entsprechendes Verzeichnis hinzugefügt. Als nächstes wird die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, das Eingabedatensegment **13** verarbeiten. Wie in **Fig.** 4 zu sehen ist, umfasst das Eingabedatensegment **13** die Eingabedatenabschnitte E, F, G, H, A, M, J, K, P.

[0110] Wie zuvor, wird dabei das Eingabedatensegment **13** in Eingabedatenabschnitte **12** verarbeitet. Jeder dieser Eingabedatenabschnitte **12** kann dann gegebenenfalls im dünn besetzten Abschnittindex **8** „nachgeschlagen" werden. In einer Ausführungsform werden nur diejenigen Eingabedatenabschnitte **12** des Eingabedatensegments **13** mit einer „vorgegebenen Eigenschaft" im dünn besetzten Abschnittindex **8** nachgeschlagen. In jedem Fall ist es die Absicht, wenigstens einen der Eingabedatenabschnitte **12** des Eingabedatensegments **13** mit den „Haken" zu vergleichen, die im dünn besetzten Abschnittindex **8** gespeichert sind.

[0111] **Fig.** 4 ist zu entnehmen, dass nur die Eingabedatenabschnitte M und P des Eingabedatensegments **13** die vorgegebene Eigenschaft haben. Bei der Verarbeitung des Eingabedatensegments **13** wird die Datenverarbeitungsvorrichtung somit Verzeichnisse identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem Eingabedatenabschnitt des Eingabedatensegments **13** entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist. Wie oben beschrieben, sind nach der Verarbeitung der Eingabedatensegmente **1** und **11** zwei Verzeichnisse **7** im Verzeichnisspeicher **5** gespeichert. Zusätzlich gibt es dann für jeden der Musterdatenabschnitte B, D und M einen Eintrag im dünn besetzten Abschnittindex **8**. Es ist zu erkennen, dass nur das zweite Verzeichnis **7** im Verzeichnisspeicher **5** – also das zu Eingabedatensegment **11** gehörige – einen Verweis auf den Musterdatenabschnitt M hat. Dementsprechend wird die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, bei der Verarbeitung des Eingabedatensegments **13** gegebenenfalls jeden der Eingabedatenabschnitte **12** des Eingabedatensegments **13** „nachschlagen" und jeden dieser Eingabedatenabschnitte mit den Einträgen im dünn besetzten Abschnittindex **8** vergleichen. In einer weiteren Ausführungsform werden gegebenenfalls nur diejenigen Eingabedatenabschnitte **12** des Eingabedatensegments **13** mit der „vorgegebenen Eigenschaft" – also M und P – im dünn besetzten Abschnittindex **8** „nachgeschlagen". In einer solchen Ausführungsform wird festgestellt, dass im dünn besetzten Abschnittindex **8** ein Eintrag vorhanden ist, der sich auf den Musterdatenabschnitt M bezieht, der dem Eingabedatenabschnitt M des Eingabedatensegments **13** entspricht. Als Ergebnis hiervon gibt die Datenverarbeitungsvorrichtung das zweite im Verzeichnisspeicher **5** gespeicherte Verzeichnis **7** (das heißt, das zum Eingabedatensegment **11** gehörende) zurück. Vom Eingabedatenabschnitt ‚P' wird kein Ergebnis zurückgegeben, obwohl gegebenenfalls ein Eintrag für P dem dünn besetzten Index hinzugefügt wird.

[0112] Anschließend kann die Datenverarbeitungsvorrichtung jeden der Eingabedatenabschnitte **12** des Eingabedatensegments **13** mit jedem der Verweise vergleichen, die im zurückgegebenen Verzeichnis (für Verzeichnis **11**) enthalten sind. In einer weiteren Ausführungsform können wie oben beschrieben die Abschnittkennungen der entsprechenden Eingabe- und Musterdatenabschnitte miteinander verglichen werden.

[0113] Als Ergebnis des Vergleichsschritts wird festgestellt, dass das zurückgegebene Verzeichnis, das zum Eingabedatensegment **11** gehört, außerdem Verweise auf die Musterdatenabschnitte J und K enthält, die den Eingabedatenabschnitten J und K des Eingabedatensegments **13** entsprechen. Dementsprechend kann für das Eingabedatensegment **13** ein Verzeichnis zusammengestellt werden, das Verweise auf die Musterdatenabschnitte M, J und K umfasst, die bereits im Abschnittspeicher **4** gespeichert sind.

[0114] Es ist zu erkennen, dass das zurückgegebene Verzeichnis keinerlei Verweise auf Musterdatenabschnitte enthält, die den Eingabedatenabschnitten E, F, G, H, A und P des Eingabedatensegments **13** entsprechen.

[0115] Somit gilt, dass ohne die weiteren Eigenschaften der Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, (die weiter unten beschrieben werden) jeder der Eingabedatenabschnitte E, F, G, H, A und P gegebenenfalls dem Abschnittspeicher als Musterdatenabschnitt hinzugefügt wird. Dies ist so ungeachtet der Tatsache, dass (wie für den Leser zu erkennen ist) die Mus-

terdatenabschnitte E, F, G, H, und A bereits im Abschnittspeicher vorhanden sind. Somit kann ohne die weiteren Eigenschaften der vorliegenden Erfindung, so wie untenstehend beschrieben, eine unnötige Duplizierung der Daten im Abschnittspeicher auftreten.

[0116] Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, – ist dafür eingerichtet, Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des aktuell verarbeiteten Segments entsprechen, indem sie wenigstens ein Verzeichnis verwendet, das identifiziert wurde als wenigstens ein weiteres Segment vorhergehender Eingabedaten verarbeitet wurde.

[0117] Mit Bezug auf das in **Fig.** 4 dargestellte Beispiel gilt somit, dass die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, dafür eingerichtet ist, die Eingabedatenabschnitte **12** des Eingabedatensegments **13** mit wenigstens einigen der Musterdatenabschnitte zu vergleichen, auf die die Verzeichnisse verweisen, die bei der Verarbeitung des vorhergehenden Eingabedatensegments **11** zurückgegeben wurden.

[0118] Wie oben beschrieben, wurde bei der Verarbeitung des Eingabedatensegments **11** das erste dem Verzeichnisspeicher hinzugefügte Verzeichnis **7** ebenfalls zurückgegeben.

[0119] Daher werden bei der Verwendung einer Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, alle Eingabedatenabschnitte **12** des Eingabedatensegments **13** gegebenenfalls mit allen Musterdatenabschnitten **4** verglichen, auf die jedes der Verzeichnisse im Verzeichnisspeicher verweist.

[0120] Bei der Ausführung des Vergleichsschritts wird daher festgestellt, dass das andere Verzeichnis im Verzeichnisspeicher Verweise auf die Musterdatenabschnitte E, F, G, H und A umfasst, die den ersten fünf Eingabedatenabschnitten **12** des Eingabedatensegments **13** entsprechen. Der einzige Eingabedatenabschnitt **12** im Eingabedatensegment **13**, für den kein entsprechender Datenabschnitt gefunden wird, ist der Eingabedatenabschnitt P. Daher wird in einer Ausführungsform ein Musterdatenabschnitt P, der dem Eingabedatenabschnitt P entspricht, dem Abschnittspeicher hinzugefügt.

[0121] Es ist zu erkennen, dass durch die Verwendung wenigstens eines Verzeichnisses, das bei der Verarbeitung wenigstens eines weiteren Eingabesegments von Eingabedaten identifiziert wurde, die Duplizierung von Musterdatenabschnitten verringert werden kann. Bei dem obenstehend mit Bezug auf **Fig.** 4 beschriebenen Beispiel, enthält der Abschnittspeicher jetzt nur eine einzelne Instanz jedes Musterdatenabschnitts. Ohne die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, wären gegebenenfalls Duplikate von wenigstens den Musterdatenabschnitten E, F, G, H und A im Abschnittspeicher vorhanden. Da der Eingabedatenabschnitt P (und somit sein entsprechender Musterdatenabschnitt) die vorgegebene Eigenschaft hat, kann somit in der oben beschriebenen Weise die zum Musterdatenabschnitt P gehörende Information dem dünn besetzten Index hinzugefügt werden.

[0122] Wird die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, verwendet, ist im dargestellten Beispiel zu sehen, dass für acht der neun Eingabedatenabschnitte **12** des Eingabedatensegments **13** festgestellt wird, dass sie Musterdatenabschnitten entsprechen, die bereits im Abschnittspeicher vorhanden sind.

[0123] Bei dem Vergleich der Eingabedatenabschnitte der Eingabedatensegmente **1**, **11** und **13**, ist, so wie in **Fig.** 5 dargestellt, für den Leser zu erkennen, dass das Muster der Abschnitte im Eingabedatensegment **1** fast identisch mit der Abfolge der Eingabedatenabschnitte **12** ist, die über die Grenze zwischen den Eingabedatensegmenten **11** und **13** hinweggeht. Mit anderen Worten gilt, dass wenn die Grenzen der Eingabedatensegmente **11** und **13** (mit Bezug auf **Fig.** 4) um fünf Plätze nach rechts verschoben würden, das sich ergebende Eingabedatensegment **11** fast vollständig dem Verzeichnis entsprechen würde, das für das Eingabedatensegment **1** erzeugt wurde. In der Praxis gibt jedoch die frei festgelegte Art, wohin die Segmentgrenzen gelegt werden, vor, dass Musterabfolgen von Eingabedatenabschnitten durch die Grenze eines Eingabedatensegments aufgeteilt werden. Die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, versucht einige der Nachteile dieser Eigenschaft zu beseitigen.

[0124] Mit Bezug auf die Figuren ist zu erkennen, dass nur die Eingabedatenabschnitte B und D des Eingabedatensegments **1** die vorgegebene Eigenschaft haben. Damit nachfolgend ein Verzeichnis für das Eingabedatensegment **1** abgerufen werden kann, muss daher das aktuell verarbeitete Eingabedatensegment wenigstens einen der Eingabedatenabschnitte B und D umfassen, so dass das dem Eingabedatensegment **1** entsprechende Verzeichnis „eingehakt" werden kann.

[0125] Obwohl das Eingabedatensegment **13** die Eingabedatenabschnitte E, F, G, H umfasst, hat keiner dieser Eingabedatenabschnitte die vorgegebene Eigenschaft, so dass das Auftreten dieser Abfolge von Eingabedatenabschnitten (E, F, G, H) in einem Eingabedatensegment nicht notwendigerweise das Verzeichnis „einhaken" wird, das dem Eingabedatensatz **1** entspricht.

[0126] Das zugrundeliegende Prinzip der Erfindung ist, dass in einem Strom von Eingabedatensegmenten ein Muster vorhanden sein kann, wobei dieses Muster zufällig durch eine Segmentgrenze aufgeteilt ist. Durch die Verwendung der Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, und insbesondere durch die Verwendung wenigstens eines Verzeichnisses, das bei der Verarbeitung eines weiteren Eingabedatensegments identifiziert wurde, kann gegebenenfalls festgestellt werden, dass der letzte Teil des wenigstens einen Verzeichnisses, das bei der Verarbeitung eines weiteren Eingabedatensegments identifiziert wurde, dem Anfangsteil des aktuell verarbeiteten Eingabedatensegments entspricht.

PRIORISIERUNG VON VERZEICHNISSEN

[0127] Wie zuvor beschrieben, ist die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, dafür eingerichtet, Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Eingabedatenabschnitte entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist. Dabei verwendet die Datenverarbeitungsvorrichtung „Haken" im dünn besetzten Abschnittindex, um eine Liste von Verzeichnissen zurückzugeben.

[0128] Außerdem ist die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, dafür eingerichtet, wenigstens ein Verzeichnis zu verwenden, das bei der Verarbeitung wenigstens eines weiteren Segments von Eingabedaten identifiziert wurde.

[0129] Alle derart identifizierten Verzeichnisse werden als „potentielle" Verzeichnisse bezeichnet.

[0130] Alle potentiellen Verzeichnisse können für nachfolgende Vorgänge priorisiert werden. Die potentiellen Verzeichnisse können entsprechend dem Wert einer Bewertung priorisiert werden, der jedem der potentiellen Verzeichnisse zugeordnet wurde.

[0131] Wie oben beschrieben gilt, dass bei der Verarbeitung eines Eingabedatensegments alle Eingabedatenabschnitte mit einer vorgegebene Eigenschaft im dünn besetzten Abschnittindex „nachgeschlagen" (look up) werden – diese werden als ‚Look-Up' Abschnitte bezeichnet. Im Zusammenhang mit jedem der Einträge für einen Musterdatenabschnitt im dünn besetzten Abschnittindex gibt es eine Liste mit wenigstens einem Verzeichnis, das einen Verweis auf diesen Musterdatenabschnitt enthält. In einigen Fällen kann ein Verzeichnis im Verzeichnisspeicher Verweise auf mehr als einen Musterdatenabschnitt haben, der einen Eintrag im dünn besetzten Abschnittindex hat und der einem Look-Up-Abschnitt entspricht. Beispielsweise wird bei der Verarbeitung eines Eingabedatensegments mit den Eingabedatenabschnitten B und D das Verzeichnis identifiziert, das dem Eingabedatensegment **1** entspricht, und es enthält zwei Verweise auf Musterdatenabschnitte, die einen Eintrag im dünn besetzten Abschnittindex haben und die diesen Look-Up-Abschnitten B und D entsprechen. Hierfür kann gesagt werden, dass das Verzeichnis zwei „Treffer" hat. Je mehr „Treffer" ein Verzeichnis hat, desto mehr Verweise enthält es auf Musterdatenabschnitte, die Look-Up-Datenabschnitten eines aktuell verarbeiteten Eingabedatensegments entsprechen. Es kann daher angenommen werden, das je höher die Zahl von „Treffern" für ein Verzeichnis ist, desto wahrscheinlicher es wenigstens teilweise mit dem aktuell verarbeiteten Eingabedatensegment übereinstimmt.

Verfahren A

[0132] In einer Ausführungsform wird jedem zurückgegebenen Verzeichnis eine Bewertung zugeordnet, die auf der Zahl der „Treffer" basiert, die es hat. Die Bewertung jedes Verzeichnisses ist daher gleich der Zahl von Musterdatenabschnitten, auf die das Verzeichnis verweist, die einen Eintrag im dünn besetzten Abschnittindex haben und die Look-Up-Abschnitten des aktuell verarbeiteten Eingabedatensegments entsprechen. Die Verzeichnisse mit den meisten „Treffern" können gegebenenfalls am höchsten priorisiert werden, da es wahrscheinlich ist, dass diese Verzeichnisse ähnlich dem aktuell verarbeiteten Eingabedatensegment sind.

[0133] Das Verzeichnis mit der höchsten Bewertung kann für den nachfolgenden Vorgang ausgewählt werden. Der nachfolgende Vorgang kann der Vergleich der Musterdatenabschnitte, auf die das Verzeichnis verweist, mit allen Eingabedatenabschnitten des aktuell verarbeiteten Eingabedatensegments sein.

[0134] Nachdem alle Eingabedatenabschnitte mit den Musterdatenabschnitten, auf die das Verzeichnis verweist, verglichen wurden, werden alle etwaig vorkommenden und entsprechenden Eingabedatenabschnitte in einem neu zusammengestellten Verzeichnis für dieses Eingabedatensegment durch einen Verweis auf den entsprechenden Musterdatenabschnitt repräsentiert.

[0135] Wenn ein Verzeichnis aus der Liste der potentiellen Verzeichnisse ausgewählt wird, kann gesagt werden, dass das ausgewählte Verzeichnis ein „Champion" ist.

[0136] Nachdem das aktuelle Championverzeichnis verarbeitet wurde, kann ein weiterer Champion aus den potentiellen Verzeichnissen gewählt werden. Das nächste ausgewählte Championverzeichnis

kann das Verzeichnis mit der zweithöchsten Bewertung sein (wobei der vorhergehende Champion die höchste Bewertung hatte).

[0137] Es ist zu erkennen, dass es in der Liste der potentiellen Verzeichnisse viele Verzeichnisse geben kann und nur einige davon gegebenenfalls als Champions ausgewählt werden. Wie oben beschrieben, kann die Verarbeitung eines Eingabedatensegments beendet werden, nachdem eine vorgegebene Bedingung erreicht wurde. Diese Bedingung kann in einer Ausführungsform sein, dass ein Eingabedatensegment mit einer vorgegebenen Zahl von zurückgegebenen Verzeichnissen verglichen wurde. Oder sie kann in einer bevorzugten Situation sein, dass zu allen Eingabedatenabschnitten des aktuell verarbeiteten Eingabedatensegments entsprechende Musterdatenabschnitte gefunden wurden. Es gäbe somit keinen Grund für eine weitere Verarbeitung.

[0138] In jedem Fall kann es weit weniger Champions als insgesamt potentielle Verzeichnisse gegen.

[0139] Wenn eine Championverzeichnis verarbeitet wurde, kann es von der Liste der potentiellen Verzeichnisse entfernt werden. Alternativ kann es in der Liste der potentiellen Verzeichnisse verbleiben, es werden jedoch Vorkehrungen getroffen, um sicherzustellen, dass es bei der Verarbeitung des aktuellen Eingabedatensegments nicht erneut als Champion ausgewählt wird

Verfahren B

[0140] In einer alternativen Ausführungsform kann die Bewertung der potentiellen Verzeichnisse neu berechnet werden, nachdem ein Champion ausgewählt wurde. In einer Ausführungsform wird die Bewertung eines Verzeichnisses erneut berechnet, so dass sie gleich der Zahl von Verweisen ist, die jedes Verzeichnis auf Look-Up-Abschnitte enthält, auf die NICHT durch die zuvor ausgewählten Champions verwiesen wurde. Falls ein Champion Verweise auf Musterdatenabschnitte umfasst, die bestimmten Look-Up-Abschnitten entsprechen, dann haben somit alle verbleibenden potentiellen Verzeichnisse, die nur Verweise auf dieselben Musterdatenabschnitte umfassen, eine Bewertung von Null.

[0141] Mit anderen Worten gilt, dass wenn ein Verzeichnis abgerufen und als Champion ausgewählt wurde, das einen bestimmten ‚Haken' verwendet hat, allen Verzeichnissen, die ebenfalls unter Verwendung dieses Hakens abgerufen wurden, wegen dieses Hakens in nachfolgenden Verarbeitungen von Verzeichnissen keine Priorität gegeben wird.

Verfahren C

[0142] In einer Ausführungsform der vorliegenden Erfindung werden kürzlich abgerufene oder erzeugte Verzeichnisse in einem Verzeichniscache gehalten. Der Cache nimmt gegebenenfalls eine vorgegebene Zahl von Verzeichnissen auf. Vorzugsweise wird der Cache im RAM gehalten.

[0143] Bei der Verarbeitung jedes Eingabedatensegments werden dessen Champions (oder wenigstens einige) von der Festplatte abgerufen und im Cache gespeichert. Neu erzeugte Verzeichnisse können ebenfalls dem Cache hinzugefügt werden. Wenn der Cache eine vorgegebene Zahl von Verzeichnissen hält, erfordert das Hinzufügen von Verzeichnissen zum Cache das Entfernen vorhandener Verzeichnisse aus dem Cache. Es können alle bekannten Verwaltungsverfahren zur Verwaltung eines Caches verwendet werden, um festzustellen, welches Verzeichnis zu entfernen ist (beispielsweise das am längsten nicht mehr verwendete, dasjenige, auf welches am wenigsten häufig zugegriffen wurde etc.). Eine bestimmte Ausführungsform der vorliegenden Erfindung verwendet das Verfahren mit dem Verzeichnis, das am längsten nicht mehr verwendet wurde.

[0144] In einer weiteren Ausführungsform wird das als nächstes aus dem Cache zu entfernende Verzeichnis (teilweise) dadurch ausgesucht, dass versucht wird zu schätzen, wie nützlich jedes Verzeichnis für die nächsten paar zu verarbeitenden Eingabedatensegmente sein wird, um dann die Verzeichnisse zu entfernen, die am wenigsten nützlich erscheinen. Beispielsweise könnte ein Verzeichnis als nützlicher eingestuft werden, wenn es die letzten 10% der Abschnitte des aktuellen Eingabedatensegments abdeckt.

[0145] In einer Ausführungsform können die Bewertungen potentieller Verzeichnisse, die bereits im Verzeichniscache vorhanden sind, gewichtet werden, so dass sie in der Liste potentieller Verzeichnisse an höherer Stelle erscheinen, als dies anderweitig gleichwertige Verzeichnisse tun, die nicht im Verzeichniscache sind. Mit ‚höher' ist hier gemeint, dass das potentielle Verzeichnis eine relativ höhere effektive Bewertung hat. Die vor kürzerem hinzugefügten Verzeichnisse werden daher mit höherer Wahrscheinlichkeit als Champions ausgewählt. Verzeichnissen, die sich bereits im Cache befinden, kann ein höheres Gewicht gegeben werden, da der Vergleich mit ihnen billiger ist (es ist kein Festplattenzugriff notwendig), wodurch es vorteilhaft ist, sie zuerst zu verwenden, selbst dann, wenn dies anderweitig etwas ungünstiger wäre, in der Erwartung, dass hierdurch teure Festplattenzugriffe vermieden werden, die dadurch entstehen, dass Verzeichnisse von der Festplatte abgerufen und dem Cache zugeführt werden.

[0146] In einer Ausführungsform kann diese extra Gewichtung gegebenenfalls einem Verzeichnis nicht zugeteilt werden, wenn anderenfalls seine Bewertung bei der Verarbeitung des aktuellen Eingabedatensegments Null gewesen wäre. Dies dient dazu, es zu vermeiden, dass ein Verzeichnis mit einem Eingabedatensegment verglichen wird, das keine Verweise auf Musterdatenabschnitte hat, die Look-Up-Abschnitten im Eingabedatensegment entsprechen.

Verfahren D

[0147] In einer Ausführungsform kann zur Bewertung der potentiellen Verzeichnisse ein zusätzlicher Bonus hinzugefügt werden, der darauf basiert, wann sie zuletzt als ein „Champion" ausgewählt wurden. Die Bewertung kann wenigstens teilweise darauf basieren, vor wie langer Zeit ein gegebenes Verzeichnis bei der Verarbeitung vorhergehender Eingabedatensegmente als ein Champion ausgewählt wurde. Falls beispielsweise ein potentielles Verzeichnis bei der Verarbeitung des unmittelbar vorangehenden Eingabedatensegments als Championverzeichnis ausgewählt wurde, ist es wahrscheinlich, dass es bei der Verarbeitung des aktuellen Eingabedatensegments nützlich sein wird. Entsprechend kann seine Bewertung angepasst werden, damit seine Wahrscheinlichkeit steigt, bei der Verarbeitung des aktuellen Eingabedatensegments als Champion ausgewählt zu werden. Außerdem gilt, dass wenn ein gegebenes potentielles Verzeichnis zuletzt bei der Verarbeitung eines viel früheren Eingabedatensegments als Champion ausgewählt wurde, es unwahrscheinlicher ist, dass es bei der Verarbeitung des aktuellen Eingabedatensegments nützlich ist. Seine Bewertung kann entsprechend angepasst werden. In einer Ausführungsform kann die Bewertung eines potentiellen Verzeichnisses darauf basierend „abklingen", wie lange es her ist, dass es zuletzt als Champion verwendet wurde.

[0148] In noch einer weiteren Ausführungsform kann die Bewertung eines potentiellen Verzeichnisses darauf basierend eingestellt werden, wie nützlich dieses war, als es bei der Verarbeitung eines gegebenen Eingabedatensegments als Champion ausgewählt wurde. Beispielsweise könnte ein Championverzeichnis ausgewählt werden, jedoch könnte es keine Verweise auf Musterdatenabschnitte enthalten, die weiteren Eingabedatenabschnitten des aktuell verarbeiteten Eingabedatensegments entsprechen. Mit anderen Worten wurde es zwar als Champion ausgewählt, konnte jedoch bei der Entduplizierung der im Eingabedatensegment enthaltenen Daten nicht helfen.

[0149] Falls jedoch ein potentielles Verzeichnis als es als Championverzeichnis ausgewählt wurde, besonders nützlich bei der Verarbeitung dieses Eingabedatensegments war, kann diesem potentiellen Verzeichnis bei der Verarbeitung des aktuellen Eingabedatensegments eine höhere Bewertung zugeordnet werden.

Verfahren E

[0150] Wie oben beschrieben, ist die Datenverarbeitungsvorrichtung, die eine Ausführungsform der vorliegenden Erfindung ist, bei der Verarbeitung eines Segments von Eingabedaten dafür eingerichtet, Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Eingabedatenabschnitte des Segments von Eingabedaten entspricht; und über den Information im dünn besetzten Abschnittindex enthalten ist.

[0151] In dieser Ausführungsform werden daher nur die Eingabedatenabschnitte des aktuell verarbeiteten Eingabedatensegments mit der vorgegebene Eigenschaft verwendet, um Verzeichnisse aufzufinden. In den oben beschriebenen Ausführungsformen werden Verzeichnisse, die bei der Verarbeitung der vorherigen Eingabedatensegmente identifiziert wurden, der Liste potentieller Verzeichnisse hinzugefügt.

[0152] In einer weiteren Ausführungsform ist die Datenverarbeitungsvorrichtung dafür eingerichtet, Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf Musterdatenabschnitte haben, die einem aus einer vorgegebenen Auswahl von Eingabedatenabschnitten entsprechen und über die Information im dünn besetzten Abschnittindex enthalten ist.

[0153] In dieser Ausführungsform kann die vorgegebene Auswahl von Eingabedatenabschnitten alle Eingabedatenabschnitte des aktuell verarbeiteten Eingabedatensegments umfassen. Zusätzlich kann die Auswahl von Eingabedatenabschnitten wenigstens einen Eingabedatenabschnitt umfassen, der in einem weiteren Eingabedatensegment enthalten ist.

[0154] Mit Bezug auf **Fig.** 4 gilt, dass bei der Verarbeitung des Eingabedatensegments **13**, zusätzlich zum Identifizieren von Verzeichnissen, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Look-Up-Abschnitte (M und P) entspricht, die Datenverarbeitungsvorrichtung versuchen kann, Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der wenigstens einem der Eingabedatenabschnitte des Eingabedatensegments **11** entspricht, also dem Eingabedatensegment, das dem aktuellen Datensegment unmittelbar vorausgeht.

[0155] In einer Ausführungsform kann das „weitere" Eingabedatensegment das unmittelbar vorausgehende Eingabedatensegment sein.

[0156] In einer Ausführungsform können die aus dem vorherigen Eingabedatensegment verwendeten

Eingabedatenabschnitte die letzten N Eingabedatenabschnitte des vorherigen Eingabedatensegments sein oder sie können zufällig ausgewählt werden.

[0157] In einer Ausführungsform können die Eingabedatenabschnitte verwendet werden, die in den letzten 50% des vorherigen Eingabedatensegments enthalten waren. in einer weiteren Ausführungsform können die letzten 25% der Eingabedatenabschnitte der ausgewählten Eingabedatensegmente verwendet werden.

[0158] Die aus anderen Eingabedatensegmenten verwendeten Eingabedatenabschnitte werden nur zum Identifizieren von Verzeichnissen verwendet und diese Eingabedatensegmente werden nicht mit den zurückgegebenen Verzeichnissen verglichen. Dies ist so, da diese Eingabedatenabschnitte gegebenenfalls im aktuell verarbeiteten Eingabedatensegment nicht vorhanden sind. Es ist auch zu erkennen, dass nur die Eingabedatenabschnitte aus den anderen Eingabedatensegmenten, die die vorgegebene Eigenschaft haben, von Nutzen sind.

[0159] Mit Bezug auf **Fig.** 5 sei für dieses Beispiel angenommen, dass neben dem identifizieren von Verzeichnissen, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Eingabedatenabschnitte **12** des Eingabedatensegments **13** entspricht, die Datenverarbeitungsvorrichtung auch versucht Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, die den letzten vier Eingabedatenabschnitten des Eingabedatensegments **11** entsprechen und über die Information im dünn besetzten Abschnittindex enthalten ist.

[0160] In diesem Beispiel ist zu sehen, dass die Eingabedatenabschnitte B und D des Eingabedatensegments **11** die vorgegebene Eigenschaft haben. Dadurch werden beide bereits im Verzeichnisspeicher gespeicherten Verzeichnisse als potentielle Verzeichnisse zurückgegeben. Bei der nachfolgenden Verarbeitung der potentiellen Verzeichnisse ist es wahrscheinlich, dass das dem Eingabedatensegment **1** entsprechende Verzeichnis als Champion ausgewählt wird, in Abhängigkeit von dessen Position in der Liste der potentiellen Verzeichnisse. Wenn dies der Fall ist, wird bei der Verarbeitung dieses zurückgegebenen Verzeichnisses festgestellt, dass das Verzeichnis Verweise auf die Musterdatenabschnitte umfasst, die den Musterdatenabschnitten E, F, G, H und A des Eingabedatensegments **13** entsprechen.

[0161] Als ein Ergebnis einer solchen Verarbeitung wird festgestellt, dass es bereits im Abschnittspeicher vorhandene Musterdatenabschnitte gibt, die acht der neun Eingabedatenabschnitte **12** des Eingabedatensegments **13** entsprechen.

[0162] Obwohl im oben beschriebenen Beispiel vier der Eingabedatenabschnitte **12** des Eingabedatensegments **11** verwendet wurden, können auch weniger Eingabedatenabschnitte verwendet werden, wobei die Vorteile der vorliegenden Erfindung immer noch vorhanden sind.

[0163] Beispielsweise kann in einer Ausführungsform die Datenverarbeitungsvorrichtung versuchen Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der dem letzten Eingabedatenabschnitt des vorherigen Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex vorhanden ist.

[0164] Bei dem in **Fig.** 4 dargestellten Beispiel würde dies bedeuten, dass die Datenverarbeitungsvorrichtung versucht Verzeichnisse zu identifizieren, die einen Verweis auf den Musterdatenabschnitt D haben. Da der Musterdatenabschnitt D den vorgegebenen Verweis hat, wird dies die anderen Verzeichnisse im Abschnittspeicher identifizieren.

[0165] Wenn Verzeichnisse auf der Festplatte gespeichert werden, können sie in fortlaufende Listen gruppiert werden, die als „Gruppe” (engl. gangs) bezeichnet werden. Beispielsweise könnten die Verzeichnisse der Segmente **1**, **2** und **3** in Gruppe **1** gespeichert werden; und die Verzeichnisse der Segmente **4**, **5** und **6** könnten in Gruppe **2** sein, und so weiter. In einer Ausführungsform fällt die Größe einer Gruppe mit der maximalen Ausdehnung von Daten zusammen, die ein Schreib/Lesekopf in einem einzelnen Vorgang lesen kann. Das Lesen aller Verzeichnisse einer ganzen Gruppe wird daher die gleiche oder eine ähnliche Zeitspanne beanspruchen, wie das Lesen eines einzigen Verzeichnisses in dieser Gruppe. Es kann daher vorteilhaft sein, die gesamte Gruppe, die ein interessierendes Verzeichnis enthält, in den RAM-Speicher einzulesen. Nach diesem Vorgang können einige oder alle dieser Verzeichnisse in den Verzeichniscache eingefügt werden.

[0166] Soweit sie in dieser Beschreibung und diesen Ansprüchen verwendet werden, bedeuten die Ausdrücke „umfassen” und „umfassend” und Variationen hiervon, dass die angegebenen Eigenschaften, Phasen oder Bestandteile eingeschlossen sind. Die Ausdrücke sind nicht so auszulegen, dass das Vorhandensein anderer Eigenschaften, Phasen oder Bestandteile ausgeschlossen ist.

[0167] Die in der vorangehenden Beschreibung, den nachfolgenden Ansprüchen oder der beigefügten Zeichnung offenbarten Eigenschaften, ausgedrückt in ihren speziellen Formen oder durch Mittel, die die offenbarte Funktion ausführen, oder ein Verfahren oder Prozess zum Erreichen des offenbarten Ergebnisses, können, auf geeignete Weise, getrennt

oder in irgendeiner Kombination dieser Eigenschaften, verwendet werden, um die Erfindung in ihren verschiedenen Formen zu realisieren.

[0168] Eingabedatenabschnitte, die eine vorgegebene Eigenschaft haben, Zusammenstellen einer Liste potentieller Verzeichnisse aus einem Verzeichnisspeicher, wobei die Liste Folgendes umfasst: wenigstens ein Verzeichnis, das einen Verweis auf einen Musterdatenabschnitt hat, der wenigstens einem der ausgewählten Eingabedatenabschnitte entspricht; und wenigstens ein Verzeichnis, das bei der Verarbeitung wenigstens eines weiteren Segments von Eingabedaten identifiziert wurde; und Priorisieren und Verarbeiten der potentiellen Verzeichnisse, um Musterdatenabschnitte zu identifizieren, weiteren Eingabedatenabschnitten des gerade verarbeiteten Segments entsprechen.

[0169] Eine weitere Ausführungsform der vorliegenden Erfindung stellt eine Datenverarbeitungsvorrichtung bereit, die Folgendes umfasst: einen Abschnittspeicher, der Musterdatenabschnitte enthält, einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Daten repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst, einen dünn besetzten Abschnittindex, der Information über nur einige Musterdatenabschnitte enthält, wobei die Verarbeitungsvorrichtung für Folgendes eingerichtet ist: für ein erstes Eingabedatensegment Verzeichnisse zu identifizieren, die wenigstens einen Verweis auf einen der Musterdatenabschnitte haben, der einem der Eingabedatenabschnitte des ersten Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist; das Verwenden wenigstens eines der identifizierten Verzeichnisse bei der Verarbeitung eines zweiten Eingabedatensegments, um Musterdatenabschnitte zu identifizieren, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen.

[0170] Mit Bezug auf **Fig.** 6 umfasst die vorliegende Erfindung ein Verfahren zur Verarbeitung von Daten, das Folgendes verwendet:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Daten repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst; und
einen dünn besetzten Abschnittindex, der Information über nur einige Musterdatenabschnitte enthält,
wobei das Verfahren Folgendes umfasst:
Verarbeiten **14** von Eingabedaten in mehrere Eingabedatensegmente, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
Identifizieren **15** eines ersten Satzes von Verzeichnissen, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte eines ersten Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist; und
Verwenden **16** des identifizierten ersten Satzes von Verzeichnissen und wenigstens eines Verzeichnisses, das beim Verarbeiten vorhergehender Eingabedaten identifiziert wurde, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen.

[0171] Soweit sie in dieser Beschreibung und diesen Ansprüchen verwendet werden, bedeuten die Ausdrücke „umfassen” und „umfassend” und Variationen hiervon, dass die angegebenen Eigenschaften, Phasen oder Bestandteile eingeschlossen sind. Die Ausdrücke sind nicht so auszulegen, dass das Vorhandensein anderer Eigenschaften, Phasen oder Bestandteile ausgeschlossen ist.

[0172] Die in der vorangehenden Beschreibung, den nachfolgenden Ansprüchen oder der beigefügten Zeichnung offenbarten Eigenschaften, ausgedrückt in ihren speziellen Formen oder durch Mittel, die die offenbarte Funktion ausführen, oder ein Verfahren oder Prozess zum Erreichen des offenbarten Ergebnisses, können, auf geeignete Weise, getrennt oder in irgendeiner Kombination dieser Eigenschaften, verwendet werden, um die Erfindung in ihren verschiedenen Formen zu realisieren. Fabschn

Patentansprüche

1. Datenverarbeitungsvorrichtung zum entduplizierten Speichern . von Eingabedaten, Folgendes umfassend:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Eingabedatensegmente repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst,
einen dünn besetzten Abschnittindex, der Einträge zu nur einigen im Abschnittspeicher gespeicherten Musterdatenabschnitten und Verweise auf Verzeichnisse im Verzeichnisspeicher enthält, die jeweils wenigstens einen Verweis auf die im dünn besetzten Abschnittindex eingetragenen und im Abschnittspeicher gespeicherten Musterdatenabschnitte enthalten,
wobei die Verarbeitungsvorrichtung dafür eingerichtet ist,
Eingabedaten in mehrere Eingabedatensegmente zu verarbeiten, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
ein erstes Eingabedatensegment zu verarbeiten, umfassend:

– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen,
– Identifizieren eines ersten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der im dünn besetzten Abschnittindex eingetragenen Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte des ersten Eingabedatensegments entspricht;
– Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen, und dies unter Verwendung des identifizierten ersten Satzes von Verzeichnissen,
ein zweites Eingabedatensegment zu verarbeiten, umfassend:
– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen,
– Identifizieren eines zweiten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des zweiten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte des zweiten Eingabedatensegments entspricht,
– Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen, und dies unter Verwendung des identifizierten zweiten Satzes von Verzeichnissen und wenigstens eines ausgewählten priorisierten Verzeichnisses des ersten Satzes von Verzeichnissen, wobei die Priorisierungskriterien wenigstens teilweise darauf basieren,
• wann jedes Verzeichnis dem Verzeichnisspeicher hinzugefügt wurde oder
• ob jedes einzelne Verzeichnis zur Zeit in einem Verzeichniscache gehalten wird
wobei die Datenverarbeitungsvorrichtung dafür eingerichtet, jedes ausgewählte priorisierte Verzeichnis des ersten Satzes von Verzeichnissen in absteigender Reihenfolge seiner Priorisierung zu verarbeiten, bis eine vorgegebene Bedingung zutrifft.

2. Datenverarbeitungsvorrichtung nach Anspruch 1, wobei der dünn besetzte Abschnittindex Information über Musterdatenabschnitte enthält, die eine vorgegebene Eigenschaft haben.

3. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, die dafür eingerichtet ist, die Eingabedatenabschnitte des zweiten Eingabedatensegments mit den Musterdatenabschnitten zu vergleichen, auf die wenigstens eines aus dem identifizierten zweiten Satz von Verzeichnissen oder das wenigstens eine Verzeichnis des ersten Satzes von Verzeichnissen verweisen, um Musterdatenabschnitte zu identifizieren, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen.

4. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, die dafür eingerichtet ist, wenigstens eines aus dem zweiten Satz identifizierter Verzeichnisse im Verzeichniscache zu speichern.

5. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, die dafür eingerichtet ist, wenigstens eines der priorisierten Verzeichnisse auszuwählen, um Musterdatenabschnitte zu identifizieren, die den weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen.

6. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, wobei das erste Eingabedatensegment dem zweiten Eingabedatensegment in den Eingabedaten unmittelbar vorausgeht.

7. Verfahren zum entduplizierten Speichern von Eingabedaten, Folgendes verwendend:
einen Abschnittspeicher, der Musterdatenabschnitte enthält,
einen Verzeichnisspeicher, der mehrere Verzeichnisse enthält, von denen jedes wenigstens einen Teil zuvor verarbeiteter Eingabedatensegmente repräsentiert und wenigstens einen Verweis auf wenigstens einen der Musterdatenabschnitte umfasst; und
einen dünn besetzten Abschnittindex, der Einträge zu nur einigen im Abschnittspeicher gespeicherten Musterdatenabschnitten und Verweise auf Verzeichnisse im Verzeichnisspeicher enthält, die jeweils wenigstens einen Verweis auf die im dünn besetzten Abschnittindex eingetragenen und im Abschnittspeicher gespeicherten Musterdatenabschnitte enthalten,
wobei das Verfahren Folgendes umfasst:
Verarbeiten von Eingabedaten in mehrere Eingabedatensegmente, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist;
Verarbeiten eines ersten Eingabedatensegments, umfassend:
– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen;
– Identifizieren eines ersten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der im dünn besetzten Abschnittindex eingetragenen Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte eines ersten Eingabedatensegments entspricht und
– Verwenden des identifizierten ersten Satzes von Verzeichnissen, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen,

Verarbeiten eines zweiten Eingabedatensegments, umfassend:
– Feststellen, ob der dünn besetzte Abschnittindex Einträge zu Musterdatenabschnitten enthält, die Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen;
– Identifizieren eines zweiten Satzes von Verzeichnissen, auf die der dünn besetzte Abschnittindex verweist, wobei jedes Verzeichnis des zweiten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte eines zweiten Eingabedatensegments entspricht; und
– Verwenden des identifizierten zweiten Satzes von Verzeichnissen und wenigstens eines ausgewählten priorisierten Verzeichnisses des ersten Satzes von Verzeichnissen, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des zweiten Eingabedatensegments entsprechen, wobei die Priorisierungskriterien wenigstens teilweise darauf basieren,
• wann jedes Verzeichnis dem Verzeichnisspeicher hinzugefügt wurde oder
• ob jedes einzelne Verzeichnis zur Zeit in einem Verzeichniscache gehalten wird,
wobei jedes ausgewählte priorisierte Verzeichnis des ersten Satzes von Verzeichnissen in absteigender Reihenfolge seiner Priorisierung verwendet wird, bis eine vorgegebene Bedingung zutrifft.

8. Verfahren zur Verarbeitung von Daten nach Anspruch 7, Folgendes umfassend:
Vergleichen der Eingabedatenabschnitte des zweiten Eingabedatensegments mit den Musterdatenabschnitten, auf die wenigstens eines aus dem identifizierten zweiten Satz von Verzeichnissen oder wenigstens ein Verzeichnis des ersten Satzes von Verzeichnissen verweisen.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

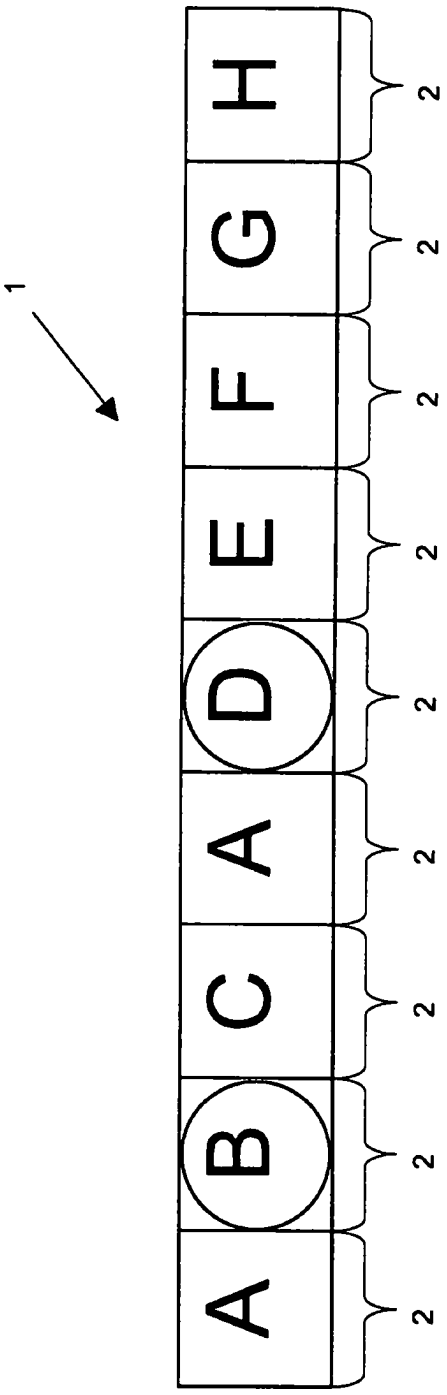


FIG 1

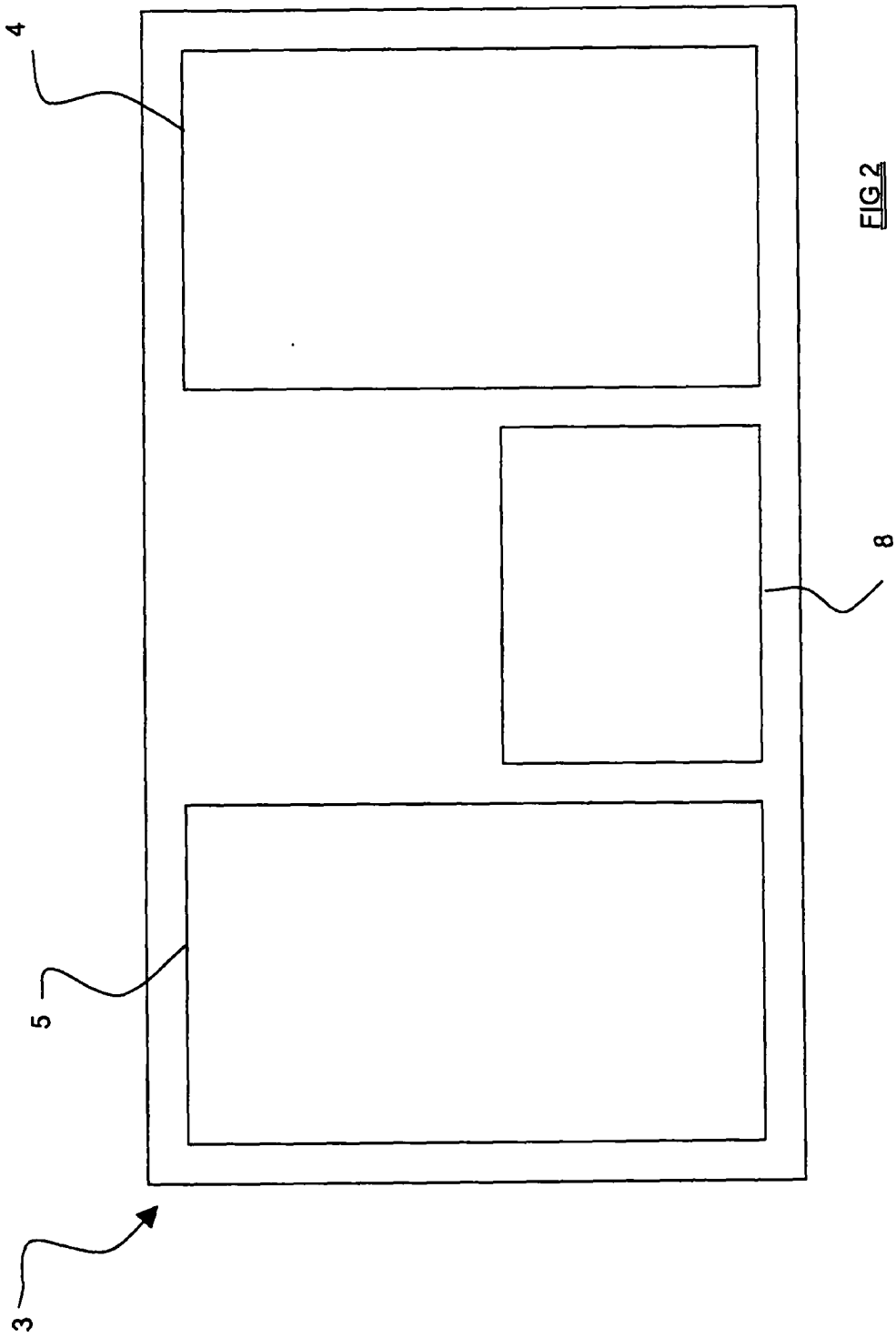


FIG 2

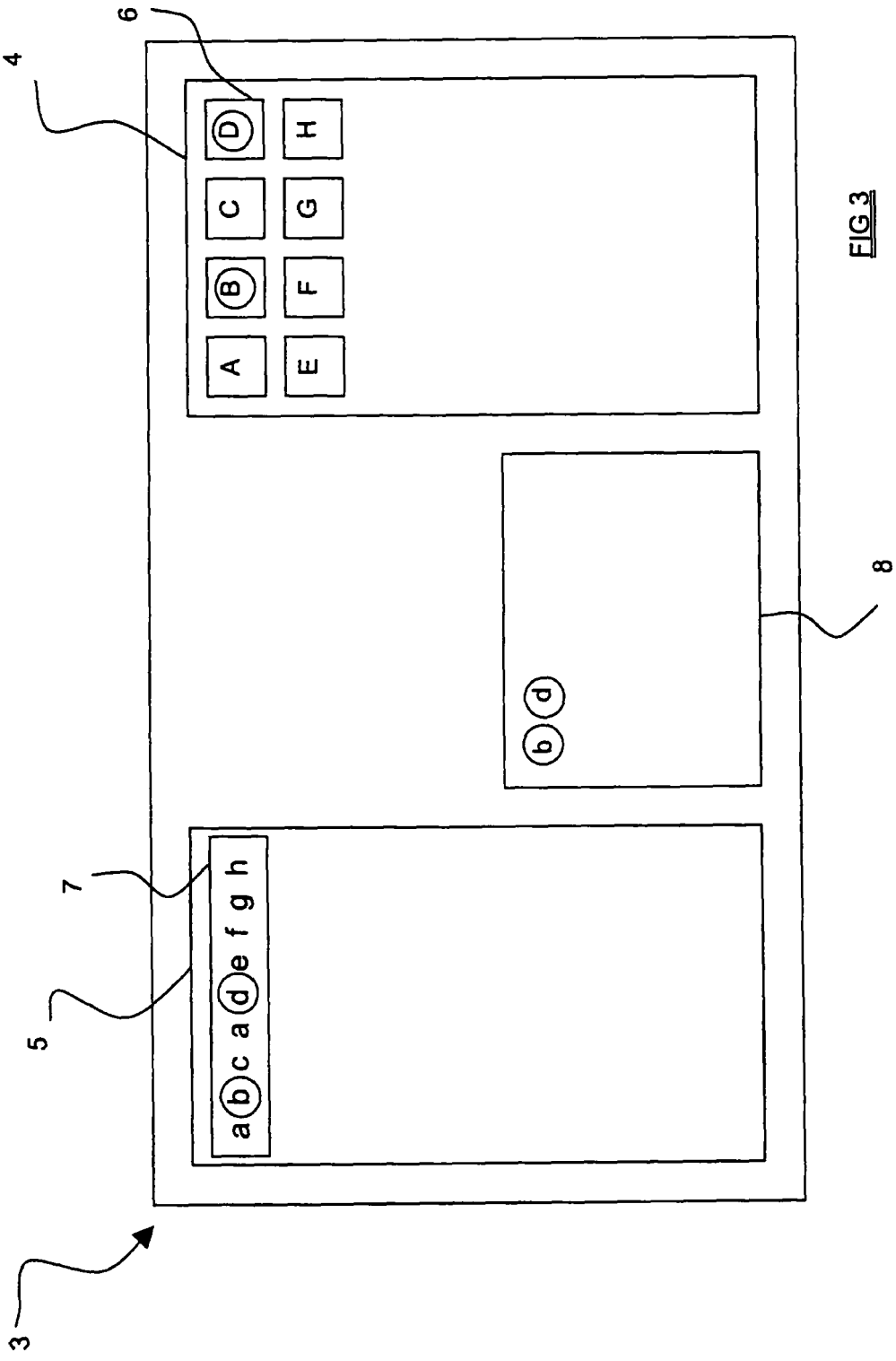


FIG 3

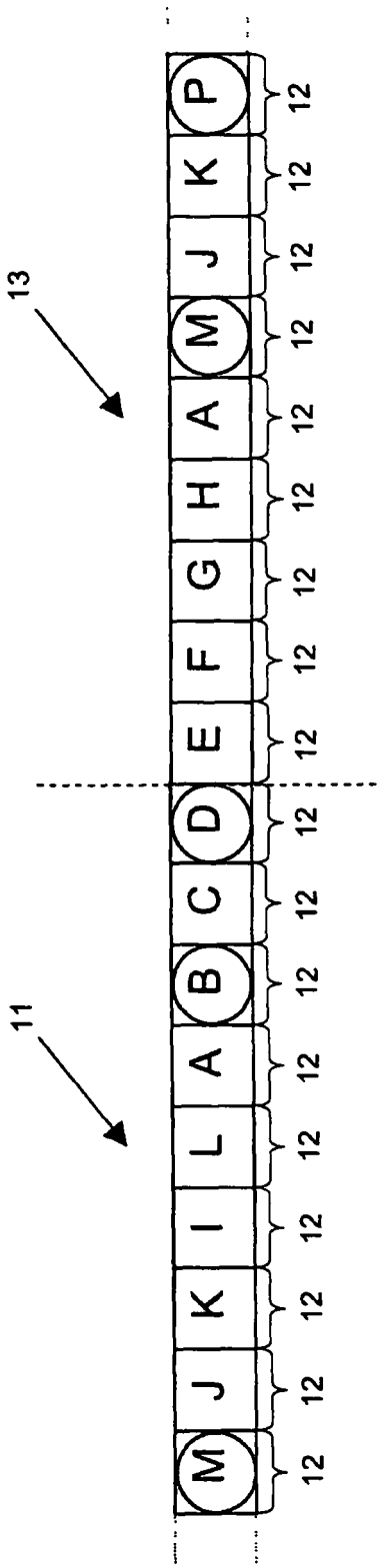


FIG 4

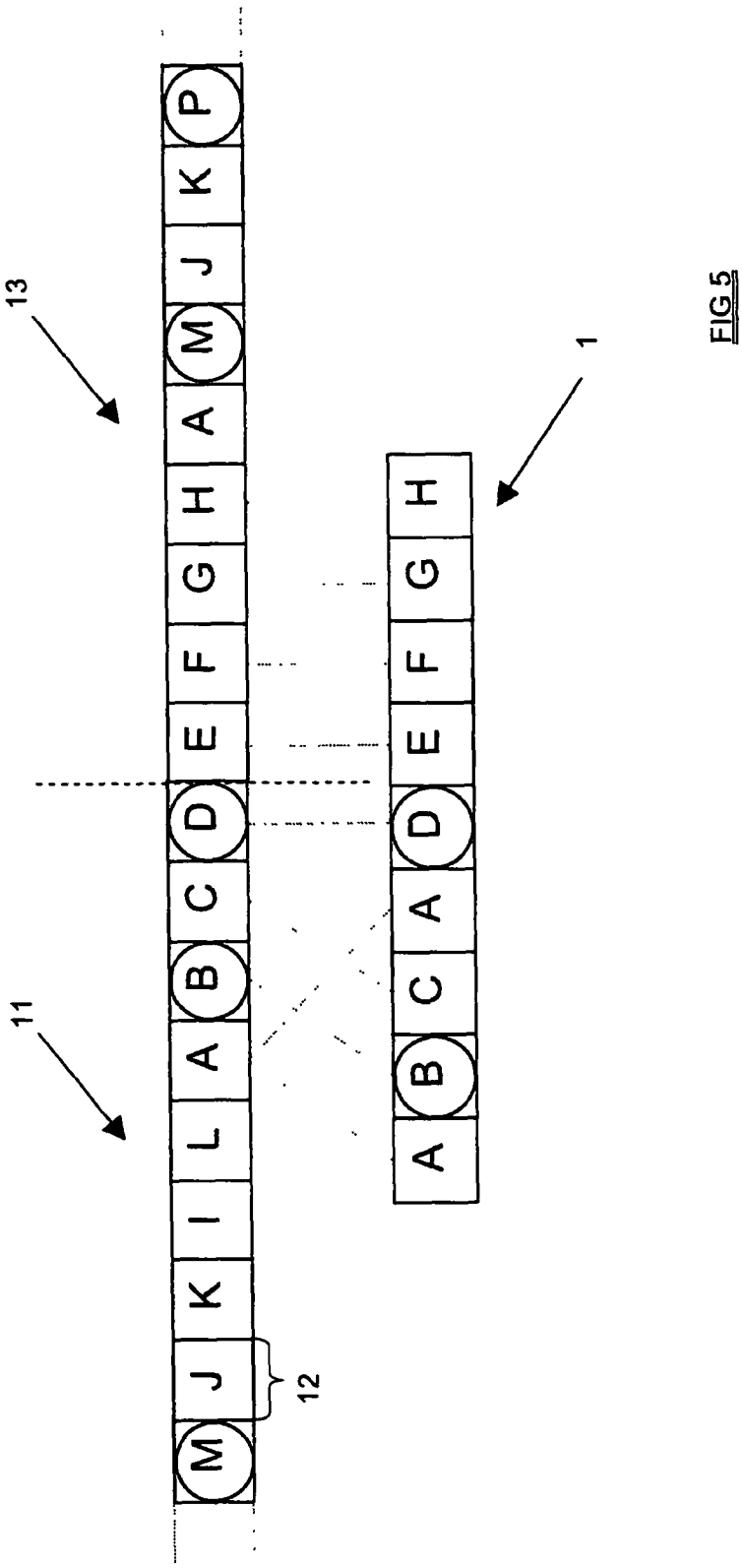


FIG 5

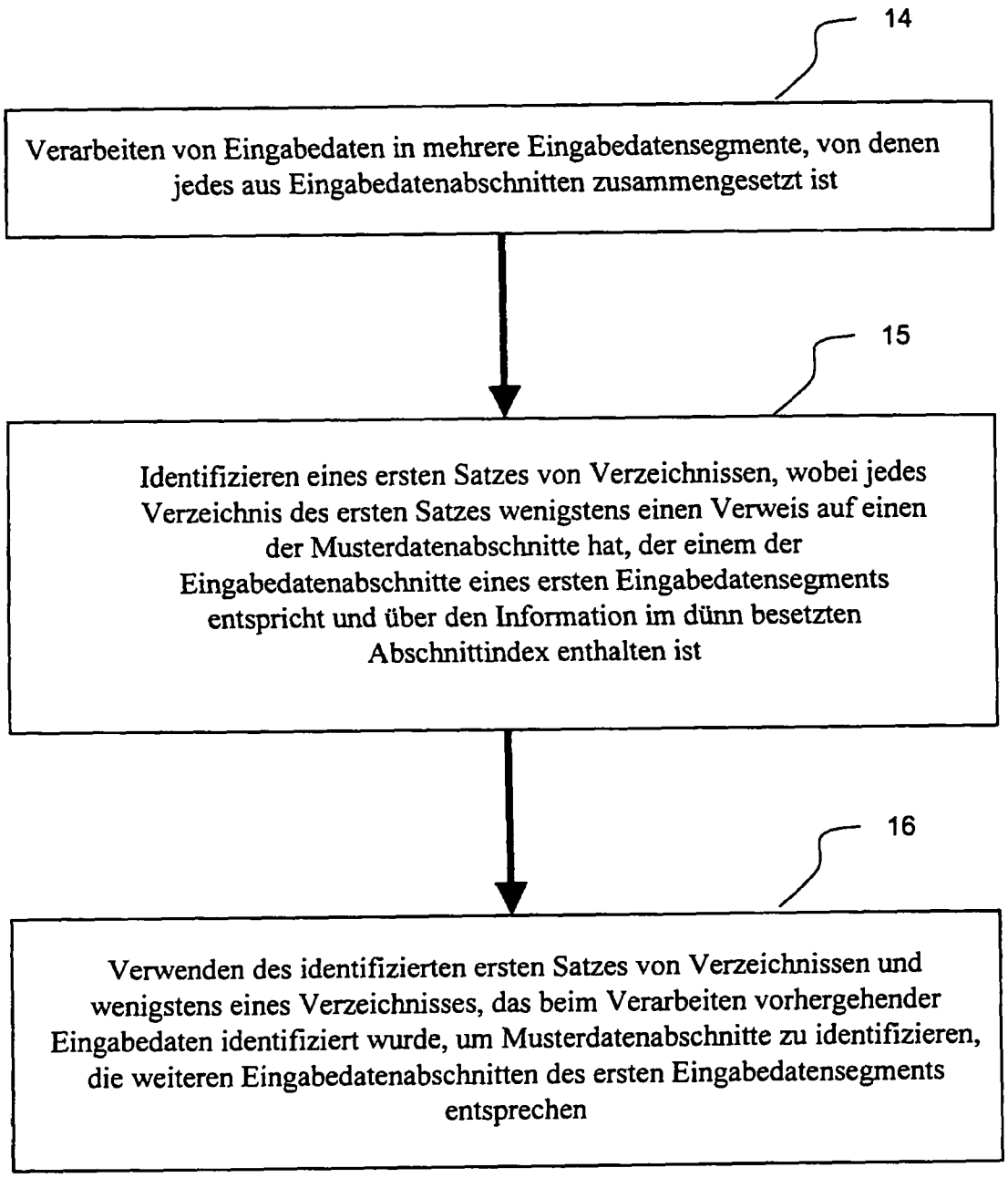

14
Verarbeiten von Eingabedaten in mehrere Eingabedatensegmente, von denen jedes aus Eingabedatenabschnitten zusammengesetzt ist
15
Identifizieren eines ersten Satzes von Verzeichnissen, wobei jedes Verzeichnis des ersten Satzes wenigstens einen Verweis auf einen der Musterdatenabschnitte hat, der einem der Eingabedatenabschnitte eines ersten Eingabedatensegments entspricht und über den Information im dünn besetzten Abschnittindex enthalten ist
16
Verwenden des identifizierten ersten Satzes von Verzeichnissen und wenigstens eines Verzeichnisses, das beim Verarbeiten vorhergehender Eingabedaten identifiziert wurde, um Musterdatenabschnitte zu identifizieren, die weiteren Eingabedatenabschnitten des ersten Eingabedatensegments entsprechen

FIG 6