

DOCUMENT MADE AVAILABLE UNDER THE PATENT COOPERATION TREATY (PCT)

| | |
|--|--|
| International application number: | PCT/AT2018/060192 |
| International filing date: | 17 August 2018 (17.08.2018) |
| Document type: | Certified copy of priority document |
| Document details: | Country/Office: AT |
| | Number: A50750/2017 |
| | Filing date: 07 September 2017 (07.09.2017) |
| Date of receipt at the International Bureau: | 04 September 2018 (04.09.2018) |

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a),(b) or (b-bis)



Kanzleigebür € 25,00
Schriftengebühr € 75,00

Aktenzeichen: A 50750/2017

Das Österreichische Patentamt bestätigt, dass die Firma
LieberLieber Software GmbH, Handelskai 340/5, 1020 Wien, ÖSTERREICH,
am 07.09.2017 eine Patentanmeldung betreffend
„Deutscher Titel: Verfahren zum Vergleich von zwei Graphen“,
überreicht hat und dass die beigeheftete Beschreibung samt Zeichnungen mit der
ursprünglich, zugleich mit dieser Patentanmeldung überreichten Beschreibung samt
Zeichnungen übereinstimmt.

Österreichisches Patentamt
Wien, 20.08.2018

Die Präsidentin:
i.A. FOINSP Gabriela Theil

Verfahren zum Vergleich von zwei Graphen

Die Erfindung betrifft ein Verfahren zum Vergleich von zwei Graphen, insbesondere von solchen Graphen, von denen einer durch manuelle Bearbeitung des anderen hergestellt wurde. Weiters betrifft die Erfindung ein Verfahren zur konfliktssicheren Aktualisierung solcher Graphen sowie zum kollaborativen Arbeiten einer Vielzahl von Personen an einem gemeinsamen Graphen.

Aus dem Stand der Technik ist es bekannt, durch Benutzung unterschiedlicher Softwarepakete Datenstrukturen zu erstellen, die in Form von Graphen abgebildet werden. Dabei wird regelmäßig ein kollaborativer Ansatz gewählt, wobei eine Vielzahl von Benutzern auf denselben Graphen bzw. auf dieselbe Datenmenge zugreifen, die durch denjenigen Graphen repräsentiert wird. Gerade durch die Anforderung einer verteilten Zusammenarbeit einer Vielzahl von Benutzern auf dieselbe Datenstruktur, wobei die einzelnen Benutzer insbesondere weltweit von unterschiedlichen Standorten aus agieren, besteht das Problem, dass der betreffende gemeinsam bearbeitete Datenbestand sehr groß ist und nicht dazu geeignet ist, über das Internet an jeden Benutzer gleichzeitig übertragen zu werden. Darüber hinaus besteht vielfach auch nicht die Anforderung, dass jeder Benutzer an jeder Stelle des einzelnen Graphen Änderungen vorzunehmen braucht.

Die Erfindung stellt sich die Aufgabe, ein Verfahren zum Vergleichen von Graphen bereit zu stellen, mit dem es für den einzelnen Benutzer möglich ist, Änderungen in Graphen bzw. in Teilen des Graphen zu detektieren.

Die Erfindung löst diese Aufgabe bei einem Verfahren der eingangs genannten Art mit dem Kennzeichen des Patentanspruchs 1. Dabei ist vorgesehen, dass

- wobei jeder der Graphen ein azyklischer gerichteter Teilgraph umfassend sämtliche der Knoten der jeweiligen Graphen zugeordnet ist,
- wobei jedem Knoten der beiden Graphen jeweils Kanteninformationen hinsichtlich derjenigen Kanten zugeordnet werden, die im jeweiligen Graphen, aber nicht in dem dem Graphen zugeordneten azyklischen gerichteten Teilgraphen enthalten sind,
- wobei jedem Knoten der beiden Graphen jeweils gegebenenfalls weitere Informationen zugeordnet sind,
- wobei ausgehend von den Blattknoten, insbesondere mittels post-order-traversal, der gerichteten azyklischen Teilgraphen für den jeweiligen Knoten jeweils ein Hash-Wert basierend auf den dem Knoten zugeordneten Kanteninformationen und weiteren

Informationen und die Hash-Werte seiner, insbesondere unmittelbaren, Kindknoten berücksichtigt werden,

- wobei zum Vergleich der beiden Graphen rekursiv ausgehend vom Wurzelknoten die Hash-Werte einander entsprechender Knoten verglichen werden, wobei

- bei Identität der Hash-Werte die Identität aller Kindknoten des jeweiligen Knotens sowie aller dem jeweiligen Knoten zugeordneten Informationen festgestellt wird und

- bei unterschiedlichen Hash-Werten

a) die dem Knoten zugeordneten Informationen sowie gegebenenfalls diejenigen Informationen, die den dem Knoten zugeordneten Kanten zugewiesen sind, verglichen werden und bei Vorliegen von Unterschieden dieser Informationen die beiden Knoten als unterschiedlich markiert werden, und

b) die Kindknoten der jeweiligen Knoten rekursiv nach vorstehend genannten Kriterien auf Unterschiedlichkeit geprüft werden.

Eine solche Vorgehensweise kann insbesondere in einem Interface zur Erstellung von Benutzerapplikationen zur Verfügung gestellt werden.

Ein besonders einfacher Vergleich von Graphen, die sich durch Abwandlung einer im wesentlichen identischen Vorlage ergeben, sieht vor, dass

den Knoten des ersten Graphen bei der Erstellung jeweils Schlüssel zugewiesen wurden der zweite Graph durch Abänderung, Hinzufügung und/oder Löschung einzelner Knoten des ersten Graphen erstellt wurde, und

dass beim rekursiven Vergleich von Kindknoten in Schritt b) jeweils solche Kindknoten miteinander verglichen werden, denen derselbe Schlüssel zugewiesen wurde.

Ein besonders einfacher Vergleich von Graphen, die sich durch Abwandlung einer im wesentlichen identischen Vorlage ergeben, sieht vor, dass

bei Feststellung von Änderungen eines zweiten Graphen gegenüber einem ersten Graphen jeweils ein Änderungsdatensatz erstellt wird, der für die als geändert erkannten Knoten jeweils angibt, welche Änderung, insbesondere auch Hinzufügung oder Löschung von Kanten oder Kindknoten, im betreffenden Knoten des zweiten Graphen gegenüber dem jeweiligen Knoten des ersten Graphen vorgenommen wurde.

Eine einfache Verteilung von Änderungsinformationen, die mit bestehenden Datenstrukturen auskommt, sieht vor, dass die einen Knoten betreffenden Änderungen

sowie diejenigen Änderungen, die eine dem Knoten zugeordnete Kante betreffen, jeweils knotenweise separat zur Verfügung gestellt werden.

Gerade beim Vergleich von Graphen, die gleichen Ursprung haben, ist es mit dem erfindungsgemäßen Vorgehen sehr einfach möglich, einen azyklischen gerichteten Teilgraphen in den betreffenden zu vergleichenden Graphen zu finden, der in beiden Graphen annähernd dieselbe Gestalt hat. Gerade dann, wenn die einzelnen zu vergleichenden Graphen in großen Teilen miteinander übereinstimmen und nur einzelne Modifikationen einzelner Benutzer gegenüber einem vorab bekannten Graphen vorgenommen wurden, kann mit der erfindungsgemäßen Vorgehensweise auf einfache Weise ein einfaches Vergleichen und Versionieren von, insbesondere durch Bearbeitung derselben Datengrundlage erstellten, Graphen erreicht werden.

Eine besonders einfache Versionsverwaltung und Archivierung sieht vor, dass für den Fall, dass in den beiden Änderungsdatensätzen widersprüchliche Änderungsanforderungen enthalten sind, ein Konflikt erkannt wird, und insbesondere

- die Erstellung oder Anwendung des gemeinsamen Änderungsdatensatzes verweigert wird und/oder
- die einzelnen Knoten, für die Konflikte erkannt wurden, markiert werden.

Eine Konfliktbehandlung kann vorgenommen werden, indem zur Ermittlung widersprüchlicher Änderungsanforderungen jeweils nur Änderungsanforderungen auf Widersprüche untersucht werden, die demselben Knoten zugewiesen wurden.

Eine einfache Konfliktprüfung, die eine vorteilhafte Lokalisation von Konflikten ermöglicht, sieht vor,

- dass ein Server bereitgestellt wird, der einen Graphen oder einen Teilgraphen des Graphen zur Herunterladen zur Verfügung hält,
- dass die Schritte b) und d) mittels zumindest eines Clients ausgeführt werden, wobei
 - der Graph oder ein Teilgraph des Graphen auf den Client heruntergeladen wird,
 - der heruntergeladene Teilgraph vom Client modifiziert wird,
 - ein Änderungsdatensatz vom Client erstellt wird, der die vom Client vorgenommenen Änderungen beschreibt, und
 - der Änderungsdatensatz an den Server übermittelt wird,
- dass Schritt f) vom Server vorgenommen wird.

Über ein Computernetzwerk kann eine Kollaboration zwischen mehreren verteilt angeordneten Benutzern bzw. Clients vorgenommen werden, indem bei Feststellung eines allfälligen Konflikts

- vom Server eine die Verweigerung der Änderung enthaltende Meldung an den Client übermittelt wird, und/oder
- vom Server ein eine Konfliktmeldung erstellt wird, die angibt, für welche Knoten der Konflikt erkannt wurde.

Im Zusammenhang mit der Erfindung wird unter einem Graphen jede Datenstruktur verstanden, die eine Anzahl von Knoten und Kanten aufweist, wobei eine Kante jeweils zwei Knoten miteinander verbindet. Sowohl den Kanten als auch den Knoten können jeweils weitere Daten, insbesondere auch Daten in Form von Key-Value-Paaren zugewiesen sein. Unter einem gerichteten Graphen versteht man einen Graphen, bei dem den einzelnen Kanten eine innere Richtung zugeordnet ist, dh bei denen den beiden Knoten eine vorgegebenen Reihenfolge zugeordnet ist. Bei einem gerichteten Graphen kann die Ordnung der Knoten jeweils als Zuordnung einer Richtung verstanden werden, mit der angegeben werden kann, ob dass die betreffende Kante von dem einen Knoten zum anderen weist oder umgekehrt.

Unter einem Teilgraphen eines Graphen versteht man im vorliegenden Fall einen Graphen, der eine Untermenge der Knoten des Graphen enthält, allenfalls können die Knoten des Graphen auch den Knoten seines Teilgraphen entsprechen. Der Teilgraph enthält auch eine Untermenge der Kanten des Graphen, wobei im Teilgraphen ausschließlich Kanten des Graphen enthalten sein können, denen zwei im Teilgraphen enthaltene Knoten zugeordnet sind.

Darüber hinaus wird auch vorteilhafterweise davon ausgegangen, dass der vorliegende Graph insgesamt zusammenhängend ist, dh dass jeder Knoten über eine einen Pfad umfassend eine Anzahl von Kanten mit jedem anderen Knoten verbunden ist.

Im Zusammenhang mit der Erfindung werden Teilgraphen verwendet, die sämtliche Knoten des ursprünglichen Graphen aufweisen. Da der Graph im allgemeinen nicht notwendigerweise azyklisch sein braucht, dh auch Schleifen aufweisen kann, kann es zur Erstellung eines gerichteten azyklischen Teilgraphen erforderlich sein, einzelne Kanten zu streichen.

Ein azyklischer gerichteter Teilgraph eines derart festgelegten zusammenhängenden Graphen hat regelmäßig eine Baumstruktur, dh ausgehend von einem vorgegebenen Wurzelknoten kann durch einen Pfad entlang der Kantenrichtung jeder einzelne Knoten des Teilgraphen erreicht werden. Knoten, die entlang der vorgegebenen Kantenrichtung am Ende des jeweiligen gerichteten azyklischen Teilgraphen stehen, werden als Blattknoten bezeichnet.

Eine bevorzugte Ausführungsform der Erfindung wird anhand der folgenden Zeichnungfiguren ohne Einschränkung der Allgemeinheit näher dargestellt.

Fig. 1 zeigt einen Graphen. **Fig. 2** zeigt einen dem in Fig. 1 dargestellten Graphen zugeordneten azyklischen Teilgraphen. **Fig. 3** zeigt einen durch Abwandlung des in Fig. 1 dargestellten Graphen erstellten weiteren Graphen. **Fig. 4** zeigt einen dem in Fig. 3 dargestellten Graphen zugeordneten azyklischen Teilgraphen. **Fig. 5** zeigt einen durch Abwandlung des in Fig. 1 dargestellten Graphen erstellten weiteren Graphen. **Fig. 6** zeigt einen dem in Fig. 5 dargestellten Graphen zugeordneten azyklischen Teilgraphen. In dem in **Fig. 7** dargestellten Ausführungsbeispiel wird eine vorteilhafte gemeinsame Bearbeitung und Nutzung eines gemeinsamen Graphen durch eine Vielzahl von Benutzern auf einer Vielzahl unterschiedlicher Client-Rechner dargestellt. In **Fig. 8** ist ein Netzwerk umfassend einen Server sowie eine Clients dargestellt, mit dem ein kollaboratives Arbeiten mehrerer Benutzer an unterschiedlichen Versionen desselben Graphen möglich ist.

Ausgehend von dem in **Fig. 1** dargestellten Graphen 11 wird Bildung eines azyklischen gerichteten Teilgraphen 110 umfassend alle Knoten des ursprünglichen Graphen 11 näher dargestellt. Dabei wird der Knoten A als Wurzelknoten frei gewählt. Ebenso könnte auch ein anderer Knoten als Wurzelknoten angenommen werden. Die Wahl eines anderen Knoten als Wurzelknoten führt zu einem anderen azyklischen gerichteten Teilgraphen 110, ändert aber letztlich nichts am erfindungsgemäßen Vorgehen. Insgesamt ist es möglich ausgehend vom Wurzelknoten jeweils über einzelne dem azyklischen gerichteten Teilgraphen 110 zugehörige Kanten, wenn man sich vom Wurzelknoten aus entlang der Kantenrichtung bewegt, alle Knoten des jeweiligen Graphen 11 zu erreichen.

Die Erstellung eines azyklischen Teilgraphen 110 umfassend alle Knoten eines Graphen 11 erfolgt üblicherweise gemeinsam mit der Erstellung eines Graphen 11. Der Wurzelknoten wird als derjenige Knoten festgelegt, der als erster in den Graphen 11 eingefügt wurde.

Umfasst der Graph 11 bereits zumindest einen Knoten, so können dem Graphen 11 neue Knoten hinzugefügt werden, indem eine neue Kante hinzugefügt wird, die einen bereits bestehenden Knoten des Graphen 11 mit dem neuen Knoten verbindet. Eine solche neue Kante wird auch dem azyklischen gerichteten Teilgraphen 110 zugewiesen und zu dem neu hinzugefügten Knoten hin orientiert.

Darüber hinaus besteht auch die Möglichkeit, wenn der Graph 11 zumindest zwei Knoten aufweist, zwei bereits dem Graphen 11 zugehörige Knoten miteinander über eine neu einzufügende Kante zu verbinden. In diesem Fall wird eine Schleife im Graphen 11 erzeugt. Um einen azyklischen Graphen parallel mit der Erstellung eines Graphen 11 zu erzeugen, kann vorgesehen sein, solche Kanten nicht in den azyklischen Teilgraphen 110 aufzunehmen.

Durch das vorstehend dargestellte Einfügen neuer Knoten über Kanten kann beispielsweise ein erster Graph 11, wie in **Fig. 1** dargestellt, erreicht werden. Dabei wird initial ein Wurzelknoten A eingefügt. Anschließend werden

- ein Knoten B eingefügt und über die Kante a mit dem Knoten A verbunden,
- ein Knoten C eingefügt und über die Kante b mit dem Knoten A verbunden,
- ein Knoten D eingefügt und über die Kante c mit dem Knoten B verbunden,
- ein Knoten E eingefügt und über die Kante e mit dem Knoten C verbunden,
- ein Knoten F eingefügt und über die Kante f mit dem Knoten D verbunden,
- ein Knoten G eingefügt und über die Kante g mit dem Knoten E verbunden,
- eine Kante d eingefügt, die die beiden Knoten B und E verbindet, und
- eine Kante h eingefügt, die die beiden Knoten F und G verbindet.

Dieser erste Graph 11 umfasst somit insgesamt sieben Knoten A, B, C, D, E, F, G und acht Kanten a, b, c, d, e, f, g, h. Der in Fig. 1 dargestellte Graph 11 ist zyklisch. Er weist beispielsweise eine Schleife umfassend die Kanten a, b, d, e auf. In dem dem ersten Graphen 11 zugeordneten azyklischen Graphen (**Fig. 2**) ist eine dieser Kanten, im vorliegenden Fall die Kante d, nicht enthalten. Ebenso weist der Graph 11 eine Schleife umfassend die Kanten c, d, f, g, h auf. In dem dem ersten Graphen 11 zugeordneten

azyklischen Teilgraphen 110 ist eine dieser Kanten, im vorliegenden Fall die Kante h, nicht enthalten.

Die Kanten des azyklischen gerichteten Teilgraphen 110 des ersten Graphen 11 sind durch fette Linien sowie durch Angabe einer Richtung dargestellt. Beim ersten Graphen 11 ist beispielsweise die gerichtete Kante a so ausgerichtet, dass sie vom Wurzelknoten A zum Knoten B weist. Der azyklische gerichtete Teilgraph 110 umfasst alle Knoten A-G des ersten Graphen 11 und weist keine Zyklen auf, d.h. ausgehend vom Wurzelknoten A können über den Teilgraphen 110 bzw. dessen Kanten sämtliche Einzelknoten erreicht werden, jeder Knoten kann jedoch ausgehend vom Wurzelknoten A über gerichtete Kanten des azyklischen Teilgraphen 110 nur auf eine einzige Weise erreicht werden. Dementsprechend ist die Kante d nicht Teil des azyklischen gerichteten Teilgraphen 110, da, wenn diese dem azyklischen gerichteten Teilgraphen 110 zugeordnet wäre, ein zyklischer Teilgraph umfassend die Kanten a, b, d, e entstünde. Ebenso ist die Kante h nicht im azyklischen gerichteten Teilgraph 110 enthalten.

Diejenigen Knoten, die ausgehend von einem Knoten in Richtung der gerichteten Kanten des azyklischen Teilgraphs 110 unmittelbar erreicht werden können, werden als dessen unmittelbare Kindknoten bezeichnet. So hat der Wurzelknoten A des ersten Graphen zwei Kindknoten B, C, die unmittelbar über jeweils eine einzige gerichtete Kante a, b entsprechend der Kantenrichtung erreicht werden können. Im vorliegenden Ausführungsbeispiel ist – mit Ausnahme des Wurzelknotens – jeder Knoten Kindknoten genau eines anderen Knotens des azyklischen gerichteten Teilgraphs 110. Diejenigen Knoten, die über eine oder mehrere gerichtete Kanten des Teilgraphen 110 entlang der Kantenrichtung erreicht werden können werden als Kindknoten bezeichnet. Diejenigen Knoten, die keine Kindknoten aufweisen, werden üblicherweise als Blattknoten bezeichnet.

Im Folgenden wird beschrieben, wie ausgehend von dem in **Fig. 1** dargestellten Graphen 11 ein zweiter Graph 12 erstellt wird (**Fig. 2**), der durch Überarbeitung und Ergänzung des ersten Graphen 11 erstellt wurde. Einander entsprechende Knoten A-G;A'-G' und a-h;a'-h' Kanten werden durch identische Bezugszeichen dargestellt und mit einem Hochkomma (') versehen. Wie aus **Fig. 3 und 4** ersichtlich, verfügt der zweite Graph 12 über eine Anzahl weiterer Knoten und Kanten, die nicht im ersten Graphen 11 enthalten sind.

Auch der betreffende azyklische gerichtete Teilgraph 110 des ersten Graphen 11 und der azyklische gerichtete Teilgraph 120 des zweiten Graphen 12 stimmen bis auf die im Bezug auf den zweiten Graphen 12 vorgenommenen Einfügungen miteinander überein. Diese Einfügungen können durch die bereits beschriebenen Operationen erreicht werden, namentlich einerseits die Hinzufügung eines neuen Knotens und Verbinden des neuen Knotens über eine neue Kante des gerichteten Graphen von einem bestehenden Knoten aus und andererseits das Verbinden von zwei bestehenden Knoten über eine neu einzufügende und nicht dem Teilgraphen 120 zuzuweisende Kante.

In diesem Zusammenhang wurde für die Erstellung des zweiten Graphen 12 ausgehend vom ersten Graphen Folgendes eingefügt:

- der Knoten H' über die gerichtete Kante i' von C' aus,
- der Knoten J' über die gerichtete Kante l' von H' aus,
- der Knoten l' über die gerichtete Kante j' von B' aus,

Die neu eingefügten Kanten k', n', m' gehören nicht zum azyklischen gerichteten Graphen des zweiten Graphen 12; dies sind im vorliegenden Fall

- eine neue Kante n', die die Knoten H' und E' verbindet
- eine neue Kante k', die die Knoten E' und l' verbindet
- eine neue Kante m', die die Knoten G' und H' verbindet

Aufgrund der konkreten Erstellung der beiden Graphen 11, 12 ist es bereits möglich, einzelne in beiden Graphen 11, 12 vorhandene Knoten und Kanten A'-G', a'-h' vorab einander zuzuordnen; dies erfolgt praktisch beispielsweise durch die Vergabe von einheitlichen Schlüsseln oder IDs, die später miteinander identifiziert werden können.

Um die Graphen 11, 12 für einen Vergleich vorzubereiten, wird das folgende Vorgehen gewählt, das für den ersten Graphen 11 im Detail dargestellt ist. Dabei wird mittels eines rekursiven Vorgehens jedem einzelnen Knoten ein ihn charakterisierender Hash-Wert k_A, \dots, k_G zugewiesen. Für den zweiten Graphen 12 wird eine Bestimmung der Hash-Werte analog vorgenommen. Bei dieser Vorgehensweise werden die einzelnen Knoten mittels post-order-traversal traversiert.

Für die Bestimmung der Hash-Werte k_F, \dots, k_G von Blattknoten F, G werden jeweils dem dem jeweiligen Knoten F, G unmittelbar zugeordneten Informationen berücksichtigt, wie beispielsweise die ID des Blattknoten F, G sowie dem Knoten zugeordnete Informationen, wie insbesondere Key-Value-Paare.

Weiters werden bei der Bildung der Hash-Werts k_F, \dots, k_G der Blattknoten F, G auch Informationen betreffend Kanten berücksichtigt, die nicht zum azyklischen Teilgraphen 110, 120 gehören, bei den beiden Blattknoten F, G werden also Informationen hinsichtlich der Kante h berücksichtigt. Diese Information kann beispielsweise durch Verweis auf eine die Kante h eindeutig bezeichnende ID oder einen Schlüssel erfolgen. Darüber hinaus können auch dieser Kante unmittelbar zugeordnete Informationen, wie insbesondere Key-Value-Paare, berücksichtigt werden.

Informationen der beim Knoten F, G eingehenden Kanten f, g werden hingegen nicht für die Bildung des Hash-Werts k_F, \dots, k_G berücksichtigt.

Bei der Bestimmung des jeweiligen Hash-Werts k_A, \dots, k_E der übrigen Knoten A, B, C, D, E des Graphen 11, 12 werden neben den bereits genannten Informationen hinsichtlich der Knoten selbst sowie der mit dem Knoten verbundenen und nicht zum Teilgraphen 110, 120 gehörenden Kanten auch die Hash-Werte der unmittelbaren Kindknoten mit berücksichtigt sowie Informationen derjenigen Kanten verwendet, die den jeweiligen Knoten mit seinen unmittelbaren Kindknoten verbinden.

Wird beispielsweise eine dem Blattknoten G zugeordnete Information geändert oder wird der Blattknoten G mit einem anderen Knoten über eine weitere nicht zum Teilgraphen 110, 120 gehörende Kante verbunden, so ändert sich zunächst der ihm zugeordnete Hash-Wert k_G .

Bei der Berechnung des Hash-Werts k_E des Knotens E sind auch die Hash-Werte der Kindknoten des Knotens E, somit auch der Hash-Werts k_G des Blattknotens G mit zu berücksichtigen. Ändert sich der Hash-Wert k_G des Blattknotens G, so ändert sich notwendigerweise auch der Hash-Werts k_E des Elternknotens E. Ebenso wirkt sich eine Änderung des Hash-Werts k_G des Blattknotens G auf alle seine unmittelbaren und mittelbaren Elternknoten E, C, A aus, während die Hash-Werte der übrigen Knoten B, F, D von der Änderung nicht betroffen sind.

Die Berechnung des Hash-Werts eines Knotens B kann dabei so erfolgen, dass rekursiv zunächst die Hash-Werte der Kinder D des betreffenden Knotens ermittelt werden und erst wenn diese zur Verfügung stehen, der Hash-Wert des betreffenden Knotens B ermittelt wird. Im Rahmen der Berechnung der Hash-Werte der Kinder D des betreffenden Knotens B kann es erforderlich sein, erneut mittels Rekursion die Hash-Werte der Kinder

F des Kindknotens D zu bestimmen. Dieser Vorgang kann rekursiv so lange wiederholt werden, bis man bei den Blattknoten angelangt ist, die keine Kinder haben.

Sollen nun die beiden Graphen 11, 12 miteinander verglichen werden, so wird für sämtliche Knoten der beiden Graphen jeweils separat für die einzelnen Knoten nach der vorstehend genannten Vorgehensweise ein Hash-Wert bestimmt, der dem betreffenden Knoten zugeordnet wird. Anschließend werden zum Vergleich der beiden Graphen 11, 12 rekursiv ausgehend vom Wurzelknoten die Hash-Werte einander entsprechender Knoten miteinander verglichen. Dabei ergibt sich beim Vergleich der beiden Wurzelknoten A ein unterschiedlicher Hash-Wert.

Beim Vergleich der beiden Wurzelknoten A, A' ergeben sich schon aufgrund der unterschiedlichen topologischen Struktur der beiden Graphen unterschiedliche Werte. Da sich beim Vergleich dieser Wurzelknoten A, A' ein unterschiedlicher Hash-Wert $k_A, k_{A'}$ ergibt, kann nun gesucht werden, in welchem der einander zuordenbaren Kindknoten B, B'; C, C' sich jeweils ein unterschiedlicher Hash-Wert ergibt. Im vorliegenden Fall ergibt ein Vergleich der beiden Kindknoten B, B' sowie der beiden Kindknoten C, C' jeweils einen unterschiedlichen Hash-Wert $k_B, k_{B'}; k_C, k_{C'}$.

Da die Knoten B, B' sowie die Knoten C, C' jeweils unterschiedliche Hash-Werte $k_B, k_{B'}; k_C, k_{C'}$ aufweisen, werden ihre jeweiligen Kindknoten rekursiv auf Unterschiedlichkeit geprüft. Dabei ergibt sich, dass den beiden einander zugeordneten Knoten D, D' derselbe Hash-Wert zugeordnet ist, da sich bei der Erstellung des zweiten Graphen 12 gegenüber dem ersten Graphen 11 weder die dem Knoten D, D' noch die dem Kindknoten F, F' zugeordneten Informationen und Kanteninformationen geändert haben. Daher kann eine weitergehendere Untersuchung der Gleichheit der Hash-Werte der Kindknoten F, F' unterbleiben.

Sofern die Änderungen der beiden Graphen 11, 12 zueinander nur geringfügig sind und sämtliche Kindknoten eines Knoten unverändert geblieben sind, kann die rekursive Suche nach Unterschieden sehr rasch abgebrochen werden und sämtliche Kindknoten der miteinander übereinstimmenden Knoten als miteinander übereinstimmend erkannt werden.

In **Fig. 5 und 6** ist ein dritter Graph 13 dargestellt, der sich von den in **Fig. 1 und 2** dargestellten Graphen lediglich dadurch unterscheidet, dass ein Knoten K'' über eine Kante o' hinzugefügt wurde, die den Knoten K'' mit dem bestehenden Knoten C''

verbindet. Einander entsprechende Knoten $A-G;A''-G''$ und $a-h;a''-h''$ Kanten werden durch identische Bezugszeichen dargestellt und mit zwei Hochkommata (") versehen.

In diesem Fall kann bereits beim Vergleich der beiden Hash-Werte $k_B;k_B''$ der Knoten $B;B''$ festgestellt werden, dass sämtliche Kinder des Knotens B mit den Kindern des Knotens B'' hinsichtlich Information und Topologie übereinstimmen und eine weitere rekursive Suche nach Unterschieden entfallen kann.

Dagegen deuten die unterschiedlichen Hash-Werte $k_C;k_C''$ der Knoten $C;C''$ auf Unterschiede in der topologischen Struktur oder in den Informationen der betreffenden Knoten hin, sodass eine weitere rekursive Suche nach Unterschieden erforderlich wird. Eine tiefe rekursive Suche stellt einerseits das Einfügen des Knotens K'' über die Kante o'' fest, andererseits wird auch festgestellt, dass der Knoten E'' sowie seine Kinder gegenüber dem Knoten E unverändert sind, sodass eine weitere Rekursion zum Vergleich der Knoten G, G'' unterbleiben kann.

Die genannte Vorgehensweise ist auch Vorteil, um Speicherplatz und Übertragungskapazitäten zu sparen, wenn mehrere Versionen eines Graphen 11, 12, 13 an unterschiedlichen Orten von unterschiedlichen Benutzern bearbeitet werden sollen. Der Vorteil kann insbesondere auch dann genutzt werden, wenn lediglich diejenigen Informationen zwischen den Benutzern übertragen werden, die Änderungen gegenüber einem vorgegebenen Datenbestand repräsentieren.

Im vorliegenden Fall ist dabei vorgesehen, dass beim rekursiven Vergleich Knoten mit unterschiedlichen Hash-Werten als unterschiedlich markiert werden. Für den Fall, dass die Knoten als unterschiedlich markiert wurden, werden die betreffenden Änderungen der Knoten zueinander abgespeichert. Für den Fall, dass keine Veränderung des betreffenden Knoten gegenüber einer Vorgängerversion vorgenommen wurde, brauchen auch keine Änderungsinformationen abgespeichert werden.

Die hier beschriebenen Änderungen können insbesondere Änderungen von einem Knoten zugeordneten Informationen, Hinzufügung oder Löschung einzelner Knoten oder Kanten sein. Um vorteilhaft den Unterschied zwischen zwei Graphen mit möglichst geringem Speicheraufwand darzustellen und ein Versionsmanagement einzelner Graphen zu erleichtern kann es von Vorteil sein, nur derartige Änderungen abzuspeichern.

Wenn also zwei Benutzer ausgehend von dem in **Fig. 1** dargestellten ersten Graphen 11 jeweils Änderungen vornehmen, die den in **Fig. 2** bzw den in **Fig. 3** dargestellten zweiten Graphen 12 und dritten Graphen 13 ergeben, so brauchen nur die folgenden Informationen, im Folgenden als Änderungsdatensatz (22, 23) bezeichnet, abgespeichert werden:

| Änderungsdatensatz 22 für den zweiten Graphen 12 | Änderungsdatensatz 23 für den dritten Graphen 13 |
|---|---|
| Einfügen von H' über i' von C' aus | Einfügen von K'' über o'' von C'' aus |
| Einfügen von J' über l' von H' aus | |
| Einfügen von l' über j' von B' aus | |
| Kante n' von H' nach E' | |
| Kante k' von E' nach l' | |
| Kante m' von G' nach H' | |

Tabelle 1 – Änderungsdatensätze

In dem in **Fig. 7** dargestellten Ausführungsbeispiel wird eine vorteilhafte gemeinsame Bearbeitung und Nutzung eines gemeinsamen Graphen durch eine Vielzahl von Benutzern auf einer Vielzahl unterschiedlicher Client-Rechner dargestellt.

Dabei wird der in **Fig. 1 und 2** dargestellte Graph als Ursprungsgraph 11 angesehen, der einen initialen Zustand bzw. gemeinsamen Bearbeitungsstand darstellt. Der erste Benutzer lädt den Graphen – in seiner Gesamtheit oder mit den ihn interessierenden Teilen – auf seinen Client-Rechner und erstellt basierend auf diesem Ursprungsgraphen 11 den in Fig. 3 dargestellten Graphen, 13, der im Folgenden als erster Bearbeitungsgraph 13 bezeichnet wird.

Daneben werden von einem oder mehreren Benutzern Änderungen am Graphen 11 vorgenommen, sodass der gemeinsame Bearbeitungsstand durch den in **Fig. 3 und 4** dargestellten Graphen ersetzt wird, der im Folgenden als aktualisierter Graph 12 bezeichnet wird.

Will der Benutzer, der auf seinem Client-Rechner den Bearbeitungsgraphen 13 erstellt hat, seine Änderungen in den gemeinsamen Bearbeitungsstand einpflegen und den aktualisierten Graphen 12 erneut aktualisieren, so besteht das Problem, dass ein bloßes Überschreiben des aktualisierten Graphen 12 durch den Bearbeitungsgraphen 13 zur Rücknahme der Änderungen führt, die bereits vorgenommen wurden.

Aus diesem Grund wird sowohl für den aktualisierten Graphen 12 als auch für den Bearbeitungsgraphen 13 jeweils ein Änderungsdatensatz 22, 23 erstellt, der die Unterschiede zwischen dem Ursprungsgraphen 11 und den betreffenden Graphen 12, 13 angibt. Ein erster Änderungsdatensatz 22 gibt die Änderungen zwischen dem Ursprungsgraphen 11 und dem aktualisierten Graphen 12 an. Ein zweiter Änderungsdatensatz 23 gibt die Änderungen zwischen dem Ursprungsgraphen 11 und dem Bearbeitungsgraphen 13 an.

Anschließend werden die beiden Änderungsdatensätze 22, 23 miteinander derart kombiniert, dass ein gemeinsamer Änderungsdatensatz 24 erstellt wird, der für die einzelnen Knoten jeweils die im ersten und zweiten Änderungsdatensatz 22, 23 angegebenen Änderungen zusammenfasst. Dieser gemeinsame Änderungsdatensatz 24 wird zur Verfügung gehalten und kann insbesondere auf dem Ursprungsgraphen 11 angewendet werden, um die Änderungen der beiden Änderungsdatensätze übernehmen zu können.

Nach der Übernahme fungiert der gemeinsame Änderungsdatensatz 24 als neuer (erster) Änderungsdatensatz 22 zur Repräsentation des aktuellen Graphen 12, sodass weitere Änderungen in den Graphen eingepflegt werden können.

In der überwiegenden Mehrzahl von Fällen betreffen die Änderungen unterschiedlicher Benutzer sehr unterschiedliche Bereiche des Graphen, die nicht notwendigerweise miteinander in Konflikt kommen. Will also der Benutzer seine Bearbeitungen den übrigen Benutzern zur Verfügung stellen, so brauchen zu diesem Zweck lediglich die Änderungsdatensätze übermittelt werden.

Sofern die beiden Bearbeiter Änderungen an unterschiedlichen Bereichen des Ursprungsgraphen 11 vornehmen und unterschiedliche Knoten modifizieren, können die Änderungsdatensätze unabhängig voneinander auf dem betreffenden Ursprungsgraphen 11 angewendet werden.

| |
|------------------------------------|
| kombinierter Änderungsdatensatz |
| Einfügen von H' über i' von C' aus |
| Einfügen von J' über l' von H' aus |
| Einfügen von l' über j' von B' aus |
| Kante n' von H' nach E' |
| Kante k' von E' nach l' |

| |
|---------------------------------------|
| Kante m' von G' nach H' |
| Einfügen von K'' über o'' von C'' aus |

Tabelle 2 – kombinierter Änderungsdatensatz

Problematisch sind Änderungen, die von den einzelnen Benutzern hinsichtlich der selben Knoten vorgenommen werden. Wenn die Einzelnen im ersten und zweiten Änderungsdatensatz angegebenen Änderungen einander hinsichtlich einzelner Knoten widersprechen, werden Konflikte erkannt. Diese Knoten, in denen Konflikte erkannt wurden, können markiert werden, sodass insbesondere dem Bearbeiter, der eine Bearbeitung vornehmen möchte, die mit dem aktuellen Graphen 12 in Konflikt steht, darstellt, worin und im Bezug auf welche Knoten, Kanten oder Informationen der konkrete Konflikt besteht.

Zusätzlich kann in einem derartigen Versionsmanagementsystem natürlich auch vorgesehen sein, dass bei Detektion von Konflikten die Erstellung eines gemeinsamen Änderungsdatensatzes 24 verweigert wird. Darüber hinaus kann auch, wenn ein solcher Änderungsdatensatz erstellt wurde, die Anwendung dieses konfliktträchtigen Änderungsdatensatzes auf dem Ursprungsgraphen verweigert werden.

In **Fig. 8** ist ein Netzwerk umfassend einen Server 30 sowie eine Clients 31 dargestellt, mit dem ein kollaboratives Arbeiten mehrerer Benutzer an unterschiedlichen Versionen desselben Graphen möglich ist. Auf dem Server 30 ist ein aktueller Graph 12, wie in den vorstehend genannten Ausführungsbeispielen dargestellt, abgespeichert. Der Server 30 ist mit den Clients 31 über ein Interface verbunden, das es dem betreffenden Client ermöglicht, den gesamten Graphen oder einen Teilgraphen 110, 120 des Graphen abzurufen. Der Client 31 ermöglicht dem Benutzer die Abänderung des jeweils heruntergeladenen Graphen. Der Client 31 kann auf die vorstehend beschriebene Weise Änderungsdatensätze 22, 23 erstellen, die angeben, auf welche Weise sich die modifizierten Graphen von den die heruntergeladenen Graphen 11, 12, 13 oder Teilgraphen 110, 120, 130 unterscheiden. Die Änderungsdatensätze 23 können vom jeweiligen Client 31 an den betreffenden Server zurück übermittelt werden, woraufhin der Server den aktuellen Graphen 12 aufgrund des übermittelten Änderungsdatensatzes aktualisiert.

Der Server prüft den Änderungsdatensatz 23 gegebenenfalls auf Konflikte mit zwischenzeitlich erfolgten Änderungen und Aktualisierungen und gibt an den Client

gegebenenfalls eine Konfliktmeldung zurück, mit der der Client informiert wird, dass die gewünschte Änderung nicht vorgenommen werden kann.

Dies kann entweder in Form einer einfachen Meldung erfolgen, mit der der Server 30 die Verweigerung der Änderung dem Client 31 mitteilt. Andererseits kann der Server 30 dem betreffenden Client 31 auch eine Konfliktmeldung übermitteln, in der angegeben ist, für welche Knoten der Konflikt erkannt wurde.

Im Rahmen der vorliegenden Erfindung besteht die Möglichkeit, dass die Änderungsdatensätze als separate Datensätze, wie in den vorstehenden Tabellen dargestellt, übertragen werden. Darüber hinaus besteht alternativ auch die Möglichkeit, Änderungen im betreffenden Graphen so zu übertragen, dass alle Änderungsinformationen knotenweise zusammengefasst werden. Dies hat den Vorteil, dass ausschließlich knotenbezogene Informationen verteilt werden, dh zur Übermittlung von Änderungen werden ausschließlich Datenstrukturen übertragen, die Knoten entsprechen, und denen die Änderungsinformation und/oder die geänderte Information zugeordnet ist.

Soll ein Knoten gelöscht werden, reicht es aus, wenn im seinem zugeordneten Elternknoten die diesen Knoten betreffende Information gelöscht wird. Soll ein Knoten hinzugefügt werden, wird der neue Knoten sowie sein modifizierter Elternknoten übermittelt. Das Hinzufügen und Löschen von Kanten sowie das Ändern von Kanteninformationen wird durch die Änderung des die Kanteninformationen speichernden Knotens mitgeteilt.

Durch diese rein knotenweise Übermittlung von Änderungen ist es besonders einfach, Konflikte zu erkennen und zu visualisieren, da Konflikte aufgrund der knotenweisen Speicherung auch im betreffenden Knoten erkannt werden.

Patentansprüche:

1. Verfahren zum Vergleich von zwei Graphen,
 - wobei jeder der Graphen (11, 12) ein azyklischer gerichteter Teilgraph umfassend sämtliche der Knoten (A-F; A'-J') der jeweiligen Graphen (11, 12) zugeordnet ist,
 - wobei jedem Knoten (A-F; A'-J') der beiden Graphen (11, 12) jeweils Kanteninformationen hinsichtlich derjenigen Kanten (d, h) zugeordnet werden, die im jeweiligen Graphen (11, 12), aber nicht in dem dem Graphen (11, 12) zugeordneten azyklischen gerichteten Teilgraphen (110, 120) enthalten sind,
 - wobei jedem Knoten (A-F; A'-J') der beiden Graphen (11, 12) jeweils gegebenenfalls weitere Informationen zugeordnet sind,
 - wobei ausgehend von den Blattknoten (G, H), insbesondere mittels post-order-traversal, der gerichteten azyklischen Teilgraphen (110, 120) für den jeweiligen Knoten jeweils ein Hash-Wert basierend auf den dem Knoten zugeordneten Kanteninformationen und weiteren Informationen und die Hash-Werte seiner, insbesondere unmittelbaren, Kindknoten berücksichtigt werden,
 - wobei zum Vergleich der beiden Graphen (11, 12) rekursiv ausgehend vom Wurzelknoten (A, A') die Hash-Werte einander entsprechender Knoten verglichen werden, wobei
 - bei Identität der Hash-Werte die Identität aller Kindknoten des jeweiligen Knotens sowie aller dem jeweiligen Knoten zugeordneten Informationen festgestellt wird und
 - bei unterschiedlichen Hash-Werten
 - a) die dem Knoten zugeordneten Informationen sowie gegebenenfalls diejenigen Informationen, die den dem Knoten zugeordneten Kanten zugewiesen sind, verglichen werden und bei Vorliegen von Unterschieden dieser Informationen die beiden Knoten als unterschiedlich markiert werden, und
 - b) die Kindknoten der jeweiligen Knoten rekursiv nach vorstehend genannten Kriterien auf Unterschiedlichkeit geprüft werden.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass
 - den Knoten des ersten Graphen (11) bei der Erstellung jeweils Schlüssel zugewiesen wurden
 - der zweite Graph (12) durch Abänderung, Hinzufügung und/oder Löschung einzelner Knoten des ersten Graphen (11) erstellt wurde, und

dass beim rekursiven Vergleich von Kindknoten in Schritt b) jeweils solche Kindknoten miteinander verglichen werden, denen derselbe Schlüssel zugewiesen wurde.

3. Verfahren nach einem der vorangehenden Ansprüche, dadurch gekennzeichnet, dass bei Feststellung von Änderungen eines zweiten Graphen (12) gegenüber einem ersten Graphen (11) jeweils ein Änderungsdatensatz erstellt wird, der für die als geändert erkannten Knoten jeweils angibt, welche Änderung, insbesondere auch Hinzufügung oder Löschung von Kanten oder Kindknoten, im betreffenden Knoten des zweiten Graphen (12) gegenüber dem jeweiligen Knoten des ersten Graphen (11) vorgenommen wurde.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass die einen Knoten betreffenden Änderungen sowie diejenigen Änderungen, die eine dem Knoten zugeordnete Kante betreffen, jeweils knotenweise separat zur Verfügung gestellt werden.

5. Verfahren zur Aktualisierung eines ersten Graphen

a) wobei ein Ursprungsgraph (11) vorgegeben wird

b) wobei ein Bearbeitungsgraph (13) durch Modifikation des Ursprungsgraphen erstellt wird,

c) wobei der Ursprungsgraph (11) aktualisiert und derart durch einen aktualisierten Graphen (12) ersetzt wird,

d) wobei gemäß Anspruch 3 oder 4 ein erster Änderungsdatensatz (22) erstellt wird, der die Änderungen zwischen Ursprungsgraphen (11) und aktualisierten Graphen (12) angibt,

e) wobei gemäß Anspruch 3 oder 4 ein zweiter Änderungsdatensatz (23) erstellt wird, der die Änderungen zwischen Ursprungsgraphen (11) und Bearbeitungsgraph (13) angibt,

f) wobei ein gemeinsamer Änderungsdatensatz (24) erstellt wird, der für die einzelnen Knoten jeweils die im ersten und zweiten Änderungsdatensatz (22, 23) angegebenen Änderungen zusammenfasst, und dieser gemeinsame Änderungsdatensatz (24) zur Verfügung gehalten wird, insbesondere auf den Ursprungsgraphen (11) angewendet wird.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass für den Fall, dass in den beiden Änderungsdatensätzen (22, 23) widersprüchliche Änderungsanforderungen enthalten sind, ein Konflikt erkannt wird, und insbesondere

- die Erstellung oder Anwendung des gemeinsamen Änderungsdatensatzes (24) verweigert wird und/oder
- die einzelnen Knoten, für die Konflikte erkannt wurden, markiert werden.

7. Verfahren nach Anspruch 4 und 6, dadurch gekennzeichnet, dass zur Ermittlung widersprüchlicher Änderungsanforderungen jeweils nur Änderungsanforderungen auf Widersprüche untersucht werden, die demselben Knoten zugewiesen wurden.

8. Verfahren nach einem der Ansprüche 5 bis 7, dadurch gekennzeichnet,

- dass ein Server (30) bereitgestellt wird, der einen Graphen (11) oder einen Teilgraphen (110) des Graphen (11) zur Herunterladen zur Verfügung hält,

- dass die Schritte b) und d) mittels zumindest eines Clients (31) ausgeführt werden, wobei

der Graph (11) oder ein Teilgraph (110) des Graphen (11) auf den Client (31) heruntergeladen wird,

der heruntergeladene Teilgraph (110) vom Client (31) modifiziert wird,

ein Änderungsdatensatz (23) vom Client (31) erstellt wird, der die vom Client (31) vorgenommenen Änderungen beschreibt, und

der Änderungsdatensatz (23) an den Server (30) übermittelt wird,

- dass Schritt f) vom Server (30) vorgenommen wird.

9. Verfahren nach einem Anspruch 8, dadurch gekennzeichnet, dass bei Feststellung eines allfälligen Konflikts

- vom Server (30) eine die Verweigerung der Änderung enthaltende Meldung an den Client (31) übermittelt wird, und/oder

- vom Server (30) ein eine Konfliktmeldung erstellt wird, die angibt, für welche Knoten der Konflikt erkannt wurde.

Zusammenfassung

Die Erfindung betrifft ein Verfahren zum Vergleich von zwei Graphen,

- wobei jeder der Graphen (11, 12) ein azyklischer gerichteter Teilgraph umfassend sämtliche der Knoten (A-F; A'-J') der jeweiligen Graphen (11, 12) zugeordnet ist,
- wobei jedem Knoten (A-F; A'-J') der beiden Graphen (11, 12) jeweils Kanteninformationen hinsichtlich derjenigen Kanten (d, h) zugeordnet werden, die im jeweiligen Graphen (11, 12), aber nicht in dem dem Graphen (11, 12) zugeordneten azyklischen gerichteten Teilgraphen (110, 120) enthalten sind,
- wobei jedem Knoten (A-F; A'-J') der beiden Graphen (11, 12) jeweils gegebenenfalls weitere Informationen zugeordnet sind,
- wobei ausgehend von den Blattknoten (G, H), insbesondere mittels post-order-traversal, der gerichteten azyklischen Teilgraphen (110, 120) für den jeweiligen Knoten jeweils ein Hash-Wert basierend auf den dem Knoten zugeordneten Kanteninformationen und weiteren Informationen und die Hash-Werte seiner, insbesondere unmittelbaren, Kindknoten berücksichtigt werden,
- wobei zum Vergleich der beiden Graphen (11, 12) rekursiv ausgehend vom Wurzelknoten (A, A') die Hash-Werte einander entsprechender Knoten verglichen werden, wobei
 - bei Identität der Hash-Werte die Identität aller Kindknoten des jeweiligen Knotens sowie aller dem jeweiligen Knoten zugeordneten Informationen festgestellt wird und
 - bei unterschiedlichen Hash-Werten
 - a) die dem Knoten zugeordneten Informationen sowie gegebenenfalls diejenigen Informationen, die den dem Knoten zugeordneten Kanten zugewiesen sind, verglichen werden und bei Vorliegen von Unterschieden dieser Informationen die beiden Knoten als unterschiedlich markiert werden, und
 - b) die Kindknoten der jeweiligen Knoten rekursiv nach vorstehend genannten Kriterien auf Unterschiedlichkeit geprüft werden.

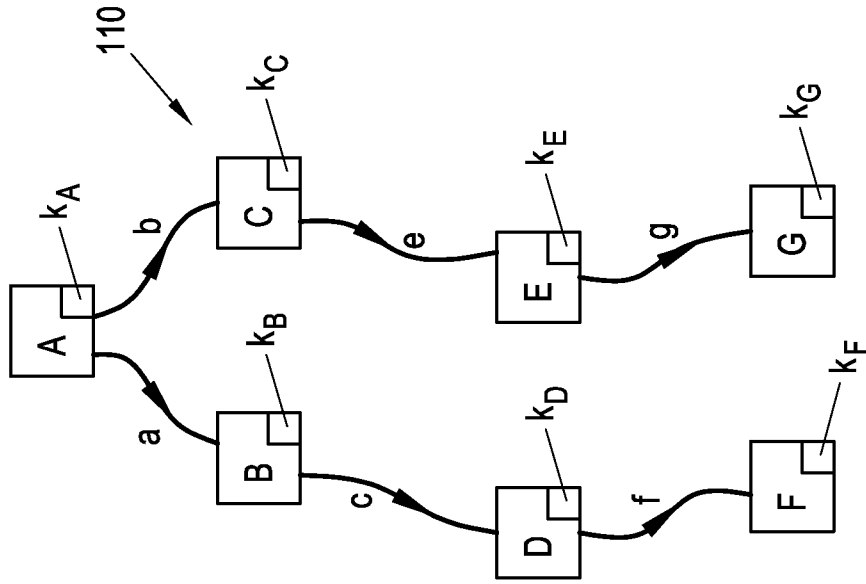


Fig. 2

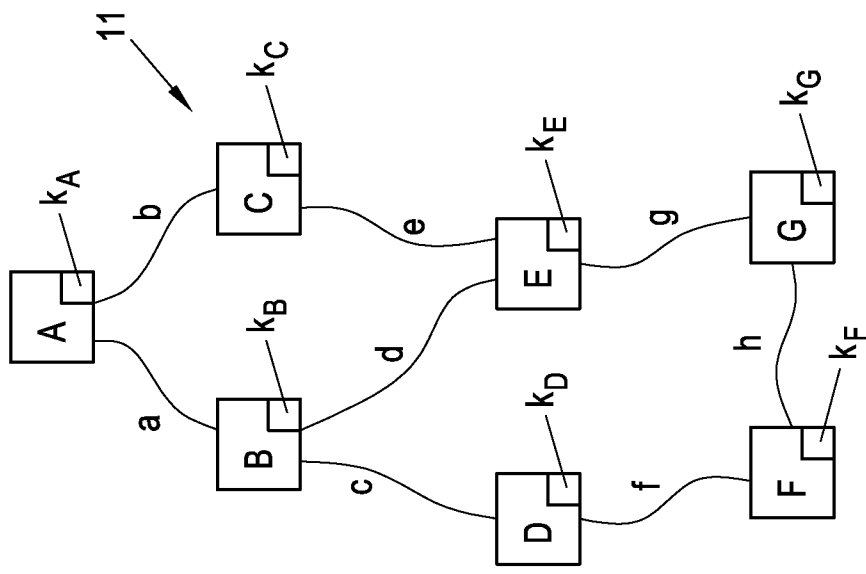


Fig. 1

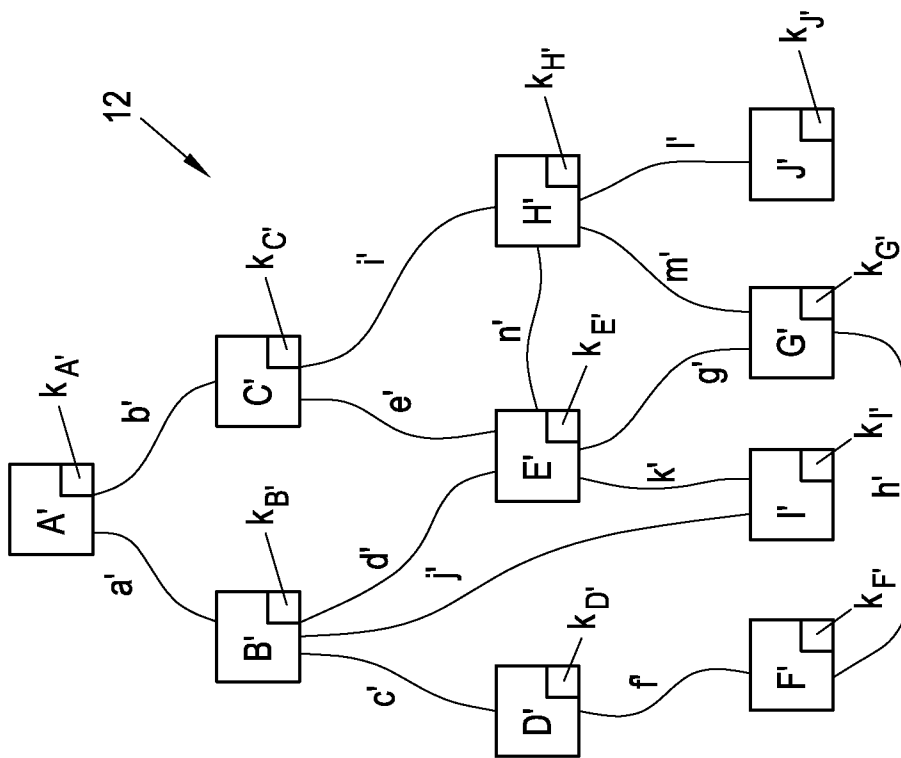
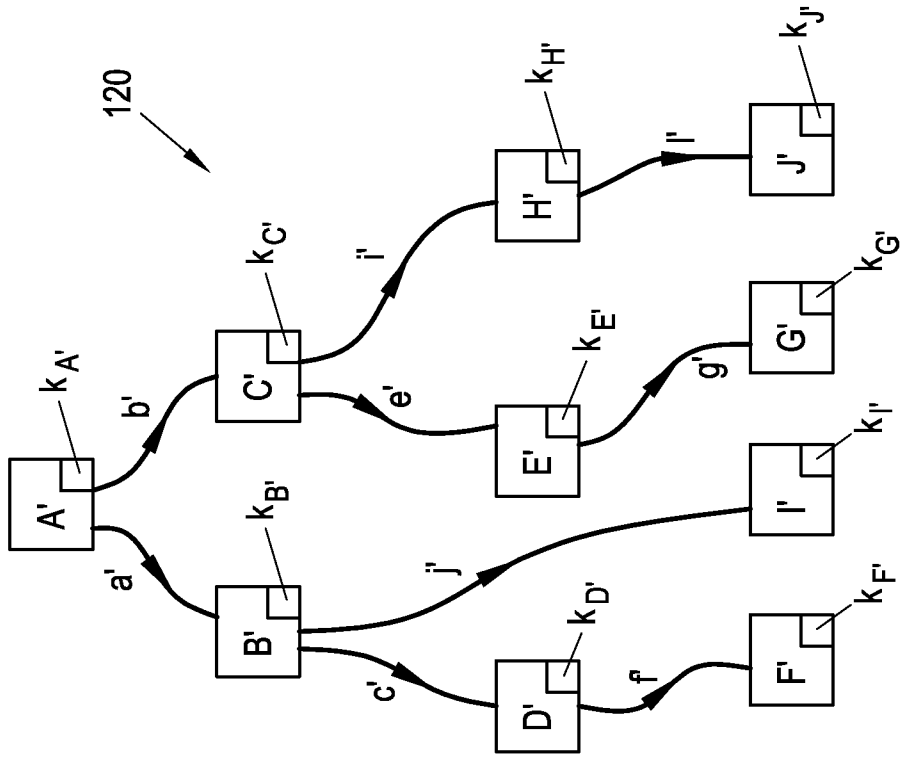


Fig. 4

Fig. 3

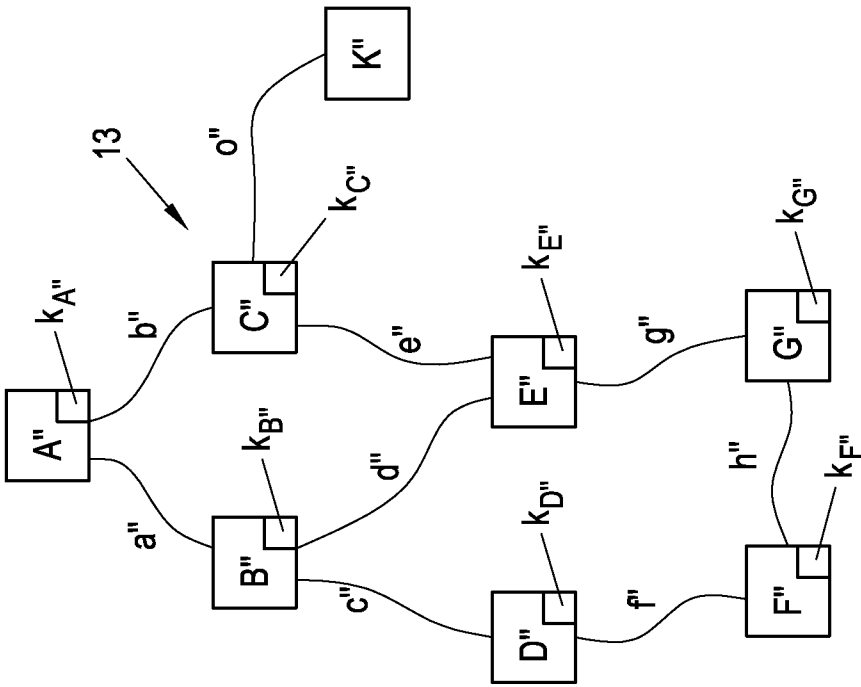
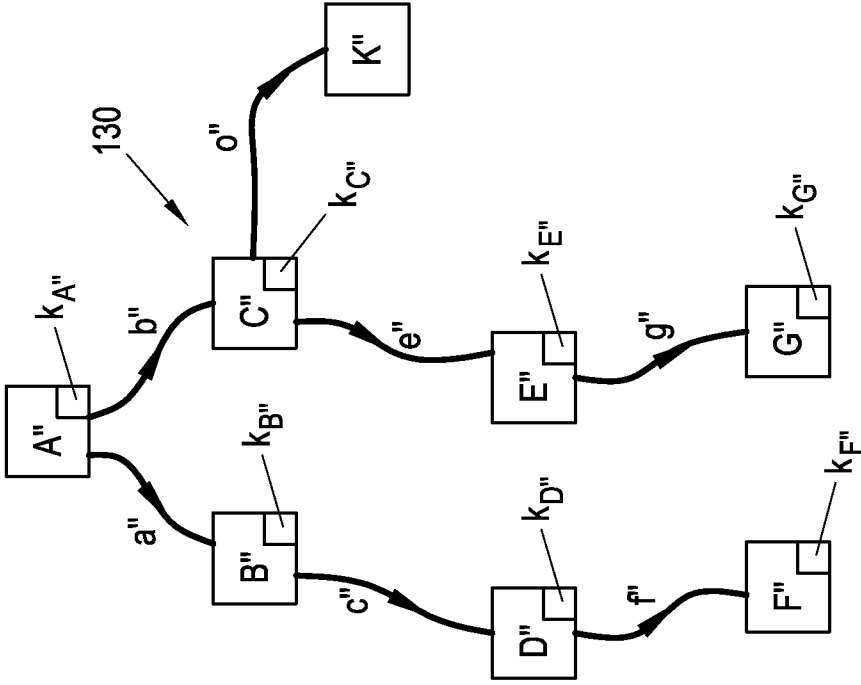


Fig. 6

Fig. 5

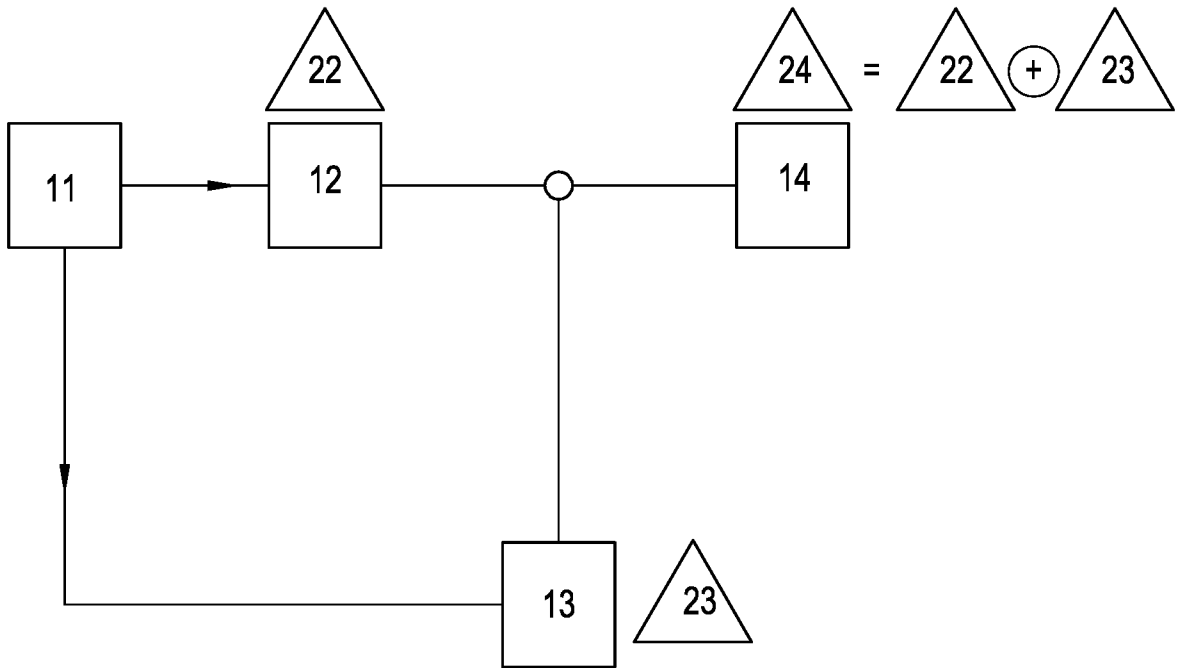


Fig. 7

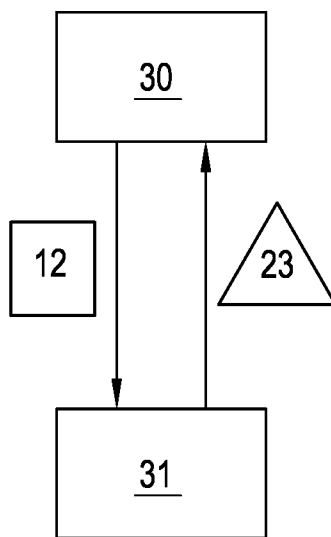


Fig. 8