

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6506929号
(P6506929)

(45) 発行日 平成31年4月24日 (2019. 4. 24)

(24) 登録日 平成31年4月5日 (2019. 4. 5)

(51) Int. Cl.

F I

G 0 6 Q 50/04 (2012. 01)

G 0 6 Q 50/04

G 0 6 F 17/50 (2006. 01)

G 0 6 F 17/50

6 0 4 G

G 0 6 F 17/50

6 0 4 A

請求項の数 13 外国語出願 (全 21 頁)

(21) 出願番号 特願2014-185168 (P2014-185168)
 (22) 出願日 平成26年9月11日 (2014. 9. 11)
 (65) 公開番号 特開2015-64876 (P2015-64876A)
 (43) 公開日 平成27年4月9日 (2015. 4. 9)
 審査請求日 平成29年9月4日 (2017. 9. 4)
 (31) 優先権主張番号 13306244.8
 (32) 優先日 平成25年9月11日 (2013. 9. 11)
 (33) 優先権主張国 欧州特許庁 (EP)

(73) 特許権者 500102435
 ダッソー システムズ
 DASSAULT SYSTEMES
 フランス国 7 8 1 4 0 ペリジー ピラ
 クブレー リュ マルセル ダッソー 1
 O
 (74) 代理人 110001243
 特許業務法人 谷・阿部特許事務所
 (72) 発明者 ジャン・フランソワ ラモー
 フランス 9 1 0 9 0 リス ル アグリ
 パ ドビニエ 3 6

審査官 木方 庸輔

最終頁に続く

(54) 【発明の名称】 二分木を用いてモデル化される工業製品を設計するためのコンピュータ実装方法

(57) 【特許請求の範囲】

【請求項 1】

基本ボリュームを表す葉ノード (L N) と、2 項可換演算組合せボリュームを表す非葉ノードと、を持つ二分木を用いてモデル化される工業製品を設計するためのコンピュータ実装方法であって、

葉ノード (L N) の追加または除去をトリガする設計動作を実行するステップと、

前記二分木を理想化するステップであって、すなわち、前記二分木を修正し、一方で、弱い葉ノード (W L N) の個数を最小化することにより前記二分木の最適性 (O B T) を維持する、ステップであって、弱い葉ノード (W L N) は、除去されると、結果として得られる二分木がもはや最適ではなくなるような葉ノードであり、最適二分木は、0 または 1 に等しい、任意の 2 個の葉ノード (L N) 間の深さの差の絶対値を有する、ステップと

を含むことを特徴とするコンピュータ実装方法。

【請求項 2】

前記設計動作が新しい葉ノード (L N) の前記追加であるとき、

最小数の葉ノード (L N) を持つ部分木を定義する次のノードを反復して選択することによって、前記二分木を、根ノード (R N) から既存の葉ノード (L N) にトラバースすることによりノードの経路を判定するステップと、

前記新しい葉ノード (L N) を、経路の終わりの前記既存の葉ノード (L N) において追加するステップと

を含むことを特徴とする請求項 1 に記載のコンピュータ実装方法。

【請求項 3】

前記設計動作が葉ノード (L N) の前記除去であるとき、

前記二分木上で、最小の深さを持つ前記葉ノード (L N) を、もはや最適ではない非最適葉ノード (L N) として判定するステップと、

前記非最適葉ノード (L N) から根ノード (R N) への経路において、部分木が非最適である第 1 のノードであるピボットノード (P N) を判定するステップと、

前記ピボットノード (P N) の非最適部分木において、最深の深さを持つ V 部分木を判定するステップであって、前記 V 部分木は 2 個の葉ノード (L N) を持つ部分木である、ステップと、

10

前記非最適葉ノード (L N) と前記 V 部分木とを入れ替えるステップと、
を含むことを特徴とする請求項 1 に記載のコンピュータ実装方法。

【請求項 4】

特定の工業製品を設計することに適合される二分木を入力するための既定義の木を提供するステップをさらに含むことを特徴とする請求項 1 から 3 のいずれか一項に記載のコンピュータ実装方法。

【請求項 5】

前記二分木の各ノードについて、その部分木の最大の深さがメモリに格納されることを特徴とする請求項 1 から 4 のいずれか一項に記載のコンピュータ実装方法。

【請求項 6】

20

前記二分木の各ノードについて、その部分木の最小の深さがメモリに格納されることを特徴とする請求項 1 から 5 のいずれか一項に記載のコンピュータ実装方法。

【請求項 7】

前記二分木の各ノードについて、その部分木の葉ノード (L N) の個数がメモリに格納されることを特徴とする請求項 1 から 6 のいずれか一項に記載のコンピュータ実装方法。

【請求項 8】

前記二分木が、非最適二分木を最適二分木 (O B T) に変換することにより取得されることを特徴とする請求項 1 から 7 のいずれか一項に記載のコンピュータ実装方法。

【請求項 9】

基本ボリュームを表す葉ノード (L N) の線形リストを、理想二分木 (I B T) に変換することにより、工業製品を表す任意の二分木を理想化するステップさらに含み、前記葉ノード (L N) は前記基本ボリュームであることを特徴とする請求項 8 に記載のコンピュータ実装方法。

30

【請求項 10】

葉ノード (L N) の個数の関連する範囲に従って、予め計算された理想二分木のライブラリを使用し、かつ、前記二分木の線形リストの基本ボリュームを表す葉ノード (L N) の個数を読み込んで、前記葉ノードの個数に対応する前記予め計算された理想二分木 (I B T) をロードし、かつ、前記予め計算された二分木の前記葉ノード (L N) を前記線形リストの前記基本ボリュームに設定することにより、工業製品を表す任意の二分木を理想化するステップさらに含むことを特徴とする請求項 8 に記載のコンピュータ実装方法。

40

【請求項 11】

請求項 1 から 10 のいずれか一項に記載の二分木を用いてモデル化される工業製品を設計するための方法を、コンピュータシステムに実行させるコンピュータ実行可能命令を有するコンピュータ可読記録媒体。

【請求項 12】

二分木を用いてモデル化される工業製品を設計するための、コンピュータ可読記録媒体に格納されるコンピュータプログラムであって、システムに請求項 1 から 10 のいずれか一項に記載の方法の前記ステップを行わせるためのコード手段を含むことを特徴とするコンピュータプログラム。

【請求項 13】

50

二分木を用いてモデル化される工業製品を設計するため装置であって、請求項 1 から 10 のいずれか一項に記載の方法の前記ステップを実装するための手段を含むことを特徴とする装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータプログラムの分野に関し、より詳細には、例えば、コンピュータ支援設計 C A D における、二分木を用いてモデル化される工業製品を設計するためのコンピュータ実装方法の分野に関する。

【背景技術】

10

【0002】

本発明は、コンピュータ支援設計 C A D を扱い、より正確には宣言型モデリング（機能モデリングとも呼ばれる）を扱う。宣言型モデリング C A D システムは、所定のヒストリ木を宣言素性（*declarative feature*）と共に含む。C A D システムのユーザが宣言素性をインスタンス化した後、結果として得られるソリッド（立体）をサポートするデータ構造は、和集合または共通集合などの可換演算を介して基本ボリュームを組み合わせた多くの部分木を含む、ヒストリ木となる。本発明は、この特別な状況を利用して、ユーザが変更を要求した後のソリッドのアップデート性能を向上させる。

【0003】

図 1 の左図に示すように、宣言型モデリングシステムの所定のヒストリ木を考え、また、所定のヒストリ木の入力パケットを考えると、宣言素性は、インスタンス化されると、基本ボリュームを入力パケット間に展開させる。その結果、いくつかの素性がインスタンス化された後、各入力パケットが可換演算（主に、「+」で示されるブール演算和）を介して複数の基本ボリュームを組み合わせる。現在、基本ボリュームの各組み合わせは、図 1 の右図に示すように、ボリュームが作成シーケンスに従って格納される番号付きリストによって、実装される。図において、根ノードが R N で示され、葉ノードが L N で示され、内部ノードが I N で示され、弧（*arc*）または辺（*edge*）がノードをつなぐ。

20

【0004】

宣言型ソリッドのアップデートには 2 つのステップが含まれることに留意されたい。第 1 のステップは、入力パケット（基本ボリュームの集合のリスト）をアップデートすることであり、第 2 のステップは、所定のヒストリ木をアップデートすることであり、ヒストリ木において、可換演算および非可換演算が格納される。工業試験によると、宣言的モデリング技法の初めての实装により、全体の計算時間が主に所定のヒストリ木のアップデート（入力パケットのアップデート時間ではなく）に費やされることが示されている。しかし、本分野における最近の性能向上（継続中のダッソーシステムズの特許：E P 2 4 7 4 9 2 8、E P 2 4 7 4 9 2 9、および E P 2 4 7 4 9 3 0 による）では、所定のヒストリ木のアップデート時間を減少させて、目下、基本ボリュームのアップデート時間が有意であるようにされ、従って最適化を行うに相応しい。

30

【0005】

以下で定義されるように、本発明は、「最適二分木」の概念を集中的に使用する。この概念は、辞書内の番号付けされたデータ（「レコード」と呼ばれる）をソートおよび探索する技術において周知である。ソートと探索の技術によると、新しいレコードが辞書に追加される度に、いわゆる「二分探索木」がアップデートされ、その構造は所与のレコードの探索を円滑にするべく設計される。探索の効率は二分木の均衡に基づく。そのため、新しいレコードが辞書に追加される度にこの均衡を維持するための精緻なアルゴリズムが存在する。探索の技術は、データベース管理のみならず計算幾何学にも使用される。著名な文献としては、D . K n u t h の「*Sorting and searching*」がある。

40

【発明の概要】

【発明が解決しようとする課題】

50

【 0 0 0 6 】

上記で説明したように、所定のヒストリ木の各入力バケットは、番号付きリストを介して基本ボリュームを組み合わせる。ユーザが、一部分の宣言素性を修正すると、基本ボリューム修正（削除、大きさの変更、位置変更．．．）が実行され、入力バケットのアップデートが必要となる。この結果、多大なアップデート時間を要することになり、何故なら、各リストのアップデートが、修正されるボリュームの世代（age）に比例する。アップデート時間の観点から、番号付きリストが最高の構造ではないことは、ソリッドモデリングの技術により周知である。

【 0 0 0 7 】

従来の技術の別の問題は、以下の通りである。宣言的モデリング技法は、素性作成順序とは無関係に動作するべく設計される。ダッソーシステムズの特許EP 1 5 0 1 0 2 6では、結果として得られるソリッドの形状が、素性作成順序とは無関係であることが保証される。しかし、入力バケットのリスト実装により、先の素性のアップデートが、後の素性のアップデートより長くなってしまい、これは、宣言動作が何であるべきなのかについてのユーザの予測と明らかに相反する。

10

【 0 0 0 8 】

探索の技術では、二分探索木の均衡が維持される。二分探索木のトポロジでは、葉ノードが番号付きレコードである、という事実に関与する構造が獲得される。例えば、典型的な要件は、左の部分木の全てのノードが、（レコードの番号付けに従うと）右の部分木の全てのノードより低いことである。これが、均衡アルゴリズムの複雑性の源である。

20

【 0 0 0 9 】

本発明の目的は、上記で言及した問題を克服するコンピュータ実装方法およびシステムを提供することである。

【課題を解決するための手段】

【 0 0 1 0 】

本発明の一態様によると、基本ボリュームを表す葉ノード（LN）と、2項可換演算組合せボリュームを表す非葉ノードと、弧でつながるノードと、を持つ二分木を用いてモデル化される工業製品を設計するためのコンピュータ実装方法が提案され、該方法は、

葉ノードの追加または除去をトリガする設計動作を実行するステップと、

30

二分木を理想化するステップであって、すなわち、二分木を修正し、一方で、弱い葉ノードの個数を最小化することにより、木の最適性を維持する、ステップであって、弱い葉ノードは、除去されると、結果として得られる二分木がもはや最適ではなくなるような葉ノードであり、最適二分木は、任意の2個の葉ノード（LN）間の深さの差の絶対値が、0または1に等しい、ステップとを含む。

【 0 0 1 1 】

そのような理想的な木を保持することにより、ユーザが変更を要求した後に、宣言型CADシステムがソリッドの新しい形状を計算するのに必要な時間を短縮することができる。さらに、宣言型CADシステムの応答時間が、より平滑になり、これは、素性のアップデート時間がその作成日時に左右されないということである。換言すると、任意の素性のアップデート時間が略同一となり、平均的に、従来技術の状態よりも短縮される。

40

【 0 0 1 2 】

迅速なアップデートは、有益な向上であり、何故なら、CAD産業においては、設計時間の80%が修正に費やされ、対照的に、設計時間の20%が新しいモデルを一から作成することに費やされる。性能の向上により時間が確保されることで、新しい製品の商品化までの時間を短縮させることができ、または、代替の設計の研究に時間を使うことができる。

【 0 0 1 3 】

アップデート時間全体を短縮することにより、また、素性間で不均衡なアップデート時

50

間を平滑にすることにより、本発明は、宣言型CADシステムをより快適に使用できるものにする。

【0014】

本発明にて、最適二分木を使用するのは、辞書内の番号付きレコードを探索することよりも優れてシンプルであるという文脈においてである。実際、二分木の全ての非葉ノードは、同一の可換演算（主に、ブール演算和）であり、また、全ての葉ノードが基本ボリュームである。これにより、状況が非常にシンプルとなり、本発明のアルゴリズムの効率は、その単純さの結果である。

【0015】

一実施形態によると、コンピュータ実装方法は、設計動作が新しい葉ノードの追加であるとき、

最小数の葉ノードを持つ部分木を定義する次のノードを反復して選択することによって、二分木を、根ノード（RN）から既存の葉ノードにトラバース（巡回）することによりノードの経路を定義するステップと、

新しい葉ノード（LN）を、経路の終わりの既存の葉ノードに追加するステップとを含む。

【0016】

従って、新しい葉ノードの追加により、木の構造を、その深さに比例する計算コストで、保存または向上させる。

【0017】

別の実施形態において、コンピュータ実装方法は、設計動作が葉ノードの除去であるとき、

二分木上で、もはや最適ではない非最適葉ノードを判定するステップであって、葉ノード（LN）は最小の深さを持つ、ステップと、

非最適葉ノードから根ノードへの経路において、部分木が非最適である第1のノードである、ピボットノードを判定するステップと、

ピボットノードの非最適部分木において、最深深さを持つV部分木を判定するステップであって、V部分木は、2個の葉ノードを持つ部分木である、ステップと、

非最適葉ノードとV部分木を入れ替えるステップとを含む。

【0018】

従って、ノード除去後の最適性の回復が、木の深さに比例する計算コストで実行される。

【0019】

一実施形態によると、コンピュータ実装方法は、特定の工業製品を設計することに適合される二分木を入力するための所定の木を提供するステップさらに含む。

従って、CADシステムは、宣言動作を特徴とし、ユーザに高レベルの設計特徴を提供する。

【0020】

一実施形態によると、二分木の各ノードについて、その部分木の最大の深さがメモリに格納される。

これにより、計算を制限することができる。

【0021】

一実施形態によると、二分木の各ノードについて、その部分木の最小の深さがメモリに格納される。

これにより、計算を制限することができる。

【0022】

一実施形態によると、二分木の各ノードについて、その部分木の葉ノードの個数がメモリに格納される。

これにより、計算を制限することができる。

10

20

30

40

50

換言すると、メモリは非一時的なコンピュータ可読媒体である。

【0023】

一実施形態によると、二分木が、非最適二分木を最適二分木に変換することにより取得される。

従って、本発明の方法によらずに作成された、工業製品または物体の二分木を使用することが可能である。

【0024】

例えば、該方法は、基本ボリュームを表す葉ノードの線形リストを、理想二分木に変換することにより、工業製品を表す任意の二分木を理想化する準備ステップさらに含み、葉ノードは基本ボリュームである。

10

【0025】

従って、本発明の方法によらずに作成された、工業製品または物体の二分木を使用することが可能である。

【0026】

例えば、該方法は、葉ノード個数の関連する範囲に従って、予め計算した理想二分木のライブラリを使用し、かつ、二分木の線形リストの基本ボリュームを表す葉ノードの個数を読み込んで、葉ノード個数に対応する予め計算された理想二分木をロードし、かつ、線形リストの基本ボリュームを予め計算された二分木の葉ノードに設定する、ことにより、工業製品を表す任意の二分木を理想化する準備ステップさらに含む。

【0027】

20

従って、本発明の方法によらずに作成された、工業製品または物体の二分木を使用することが可能である。

【0028】

また、本発明の別の態様によると、上述の二分木を用いてモデル化される工業製品を設計するための方法を、コンピュータシステムに実行させるコンピュータ実行可能命令を有するコンピュータ可読媒体が提案される。

【0029】

また、本発明の別の態様によると、二分木を用いてモデル化される工業製品を設計するための、コンピュータ可読媒体に格納されるコンピュータプログラム製品であって、システムに上述の方法のステップを行わせるためのコード化手段を含むことを特徴とする製品が提案される。

30

【0030】

また、本発明の別の態様によると、二分木を用いてモデル化される工業製品を設計するため装置であって、上述の方法のステップを実装するための手段を含むことを特徴とする装置が提案される。

【0031】

また、本発明の別の態様によると、上述の方法を用いて設計される工業製品が提案される。

【0032】

非制限的な例を用いて説明され、かつ、添付の図面により例示されるいくつかの実施形態を検討することにより、本発明がより良く理解されるであろう。

40

【図面の簡単な説明】

【0033】

【図1】従来の技術により、木構造の例を示す図である。

【図2】本発明の態様に係るコンピュータ実装方法を示す図である。

【図3】本発明の態様に係るコンピュータ実装方法を示す図である。

【図4】本発明の態様に係るコンピュータ実装方法を示す図である。

【図5】本発明の態様に係るコンピュータ実装方法を示す図である。

【図6】本発明の態様に係るコンピュータ実装方法を示す図である。

【図7】本発明の態様に係るコンピュータ実装方法を示す図である。

50

【図 8】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 9】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 10】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 11】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 12】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 13】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 14】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 15】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 16】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 17】本発明の態様に係るコンピュータ実装方法を示す図である。

10

【図 18】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 19】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 20】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 21】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 22】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 23】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 24】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 25】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 26】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 27】本発明の態様に係るコンピュータ実装方法を示す図である。

20

【図 28】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 29】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 30】本発明の態様に係るコンピュータ実装方法を示す図である。

【図 31】本発明を実装することができるコンピュータネットワークまたは同様のデジタル処理環境を示す図である。

【図 32】コンピュータの内部構造の図である。

【発明を実施するための形態】

【0034】

以下の図面により、本発明の機能がさらに詳細に説明される。

【0035】

30

図 2 に示すように、本発明では、図 1 の右図に示すような既存の番号付リストではなく、均衡のとれた二分木を介して、入力バケットの基本ボリュームの組み合わせが実装される。そして、各入力バケットのアップデート時間は、その基本ボリュームの個数の対数に比例するのであり、番号付きリストのアップデート時間がその基本ボリュームの対数に比例することとは対照的である。さらに、各均衡二分木は、その形状が、基本ボリュームの作成および削除を通じて最適に保たれるように、操作される。

【0036】

本発明の説明は、1 個の二分木の操作に着目して行う。もちろん、宣言的 CAD システムにおいて本発明を実装することにより、所定のヒストリ木構造に応じて必要な数の二分木がサービスに導入され、各二分木は本説明に従って操作される。

40

【0037】

使用する図式の定義は以下の通りである。

【0038】

木は、図 3 に示すように、非輪状の図形である。木には、厳密には弧の個数より 1 つ多いノードが含まれる。図 3 の最左の図形は木ではない（多くの輪が見られる）。図 3 の最右の図形は木である。

【0039】

二分木は、図 4 に示すように、2 個の弧を持つ根ノード RN という 1 個のノード、3 個の弧を持つ内部ノード IN という複数のノード、1 個の弧を持つ葉ノード LN という複数のノード、を特徴とする木である。従来、図 4 に示すように、製図においては、根ノード

50

R Nが最下位置に配置され、葉ノードL Nが最上位置に配置される。

【0040】

本願全体を通して、 p は葉ノードL Nの個数であり、 $p = 2^q + r$ と表され、ここで q および r は $0 \leq r < 2^q$ となる整数である。数値 q は $q = \log_2 p$ と表されることが多い。グラフ理論により周知であるように（オイラーの公式を二分木に適用することにより）ノードの総数および弧の個数が、葉ノードL Nの個数により完全に定義される。

【0041】

慣例により、図5に示すように、二分木の弧は葉ノードL Nから根ノードR Nに向くように、図面においては上部から下部に向くように、配置される。図5に示すように、根ノードR Nは出力弧を持たず2個の入力弧を持ち、葉ノードL Nは入力弧を持たず1個の出力弧を持ち、内部ノードI Nは1個の出力弧と2個の入力弧を持つ。この従来の配置向きは、本発明の文脈において、二分木により獲得されるデータフローに従って選択される。図面においては暗黙的なことである。

【0042】

図6に示すように、ノード x により定義される部分木は、ノード x の出力弧を切りとることにより取得される。その結果、ノード x は、その部分木の根ノードとなる。葉ノードは部分木としてとらえることができる。

【0043】

定義によると、2個の兄弟ノードが同一の出力ノードを持つ。図7では、点線の輪郭で、同一の出力ノード x を持つ2個の兄弟部分木が囲まれている。

【0044】

ノードの深さは、当該ノードから根ノードR Nまでの経路の弧の個数である。木の深さは、最大の葉ノード深さである。根ノードR Nの深さはゼロである。図8において、ノード x の深さは3であり、葉ノード y の深さは3であり、葉ノード z の深さは5である。

【0045】

葉ノード以外の各ノードは、従来的に「左」および「右」と呼ばれる2つの入力ノードを持つ。定義によると、葉ノード以外の所与のノードの左部分木は、その左入力ノードにより定義される部分木であり、同様に、右部分木は、その右入力ノードにより定義される部分木である。定義によると、均衡二分木では、葉ノード以外の各ノードについて、その左と右の部分木間の深さの差の絶対値が1より小さいかまたは1に等しい。図9の最左の木は不均衡であるが、最右の木は均衡である（ノードにはノードのそれぞれの部分木の深さが記されている）。

【0046】

最適二分木O B Tは特別な均衡二分木である。定義によると、最適二分木の任意の2個の葉ノードL N間の深さの差の絶対値が、0または1に等しい。換言すると、図10に示すように、葉ノードL Nについて最大で2つの深さレベルが存在する。図9の均衡二分木は、最適ではないが、図10の最左の均衡二分木は最適である。

【0047】

定義により、図10の最右に示すように、完全二分木の全ての葉ノードL Nは、同じ深さを持つ。完全二分木も葉ノードL Nの個数は、常に2の整数乗であり $p = 2^q$ を意味する。その結果、 $p = 2^q + r$ 個の葉ノードL Nを有する最適二分木の任意の葉ノードL Nの深さは、 q または $q + 1$ である。以下の特性は先行技術により周知である。最適二分木は、同じ個数の葉ノードL Nを持つ全ての二分木の中で、最小の深さであることを特徴とする。

【0048】

二分木に対して実行され、かつ、本発明において使用される演算は、「新しい葉ノードの追加」および「既存の葉ノードの除去」である。追加の演算は、以下のように行われる。象徴的に基本ボリュームを表す既存の葉ノード a を考え、また、 c という新しい基本ボリュームを考えると、追加の演算は、図11に示すように、葉ノード a を、根ノード $+$ と、葉ノード a および c とで定義される部分木で置き換えることである。

【 0 0 4 9 】

明らかに、葉ノードの追加により、二分木の深さを増加させることができる。

【 0 0 5 0 】

除去の演算は、以下のように行われる。二分木の葉ノード L_N の除去は、その出力ノードをその兄弟部分木の根ノード R_N で置き換えることである。例えば、図 1 2 において、葉ノード b を除去することは、ノード b の出力ノード $+$ を、葉ノード b の兄弟部分木 $a + c$ の根ノード $+$ で置き換えることである。

【 0 0 5 1 】

明らかに、葉ノード L_N の除去により、二分木の深さを減少させることができる。

【 0 0 5 2 】

今までのところ、全ての定義および特性は先行技術により周知である。それらは、参照するものによって異なり、何故ならそれらが厳密に呼び出されるからである。我々の知識では、次の定義は本発明に独創的なものである。

【 0 0 5 3 】

ここで、最適二分木の弱い葉ノード W_LN の考え方を定義する。最適二分木を考えると、弱い葉ノードは、定義によると、除去されると、結果として得られる二分木がもはや最適ではなくなるような葉ノードである。例えば、図 1 3 において、葉ノード x の除去により、最適性が壊されない

逆に、図 1 4 において、ノード y を除去することにより、実際に最適性が壊され、ノード y が弱い葉ノード W_LN となる。最右の木は最適ではなく、何故なら、3 つの葉ノードの深さ値 2、3、および 4 が存在するからである。

【 0 0 5 4 】

弱い葉ノードは以下のように特徴付けられる。その深さは q 、兄弟ノードは葉ノード L_N (部分木ではない) であり、二分木のどこかに深さ $q + 1$ のノードが存在する。

【 0 0 5 5 】

ここで、理想的な二分木 IBT を定義する。定義によると、理想的な二分木 IBT は、弱い葉ノードの個数が最小であることを特徴とする最適な二分木 OBT である。図 1 5 に示すように、全ての最適二分木 OBT が理想的であるというわけではない。最左の二分木は最適であるが理想的ではなく、一方、最右の二分木は理想的である。それら双方は 10 個の葉ノード L_N を持ち、最左の二分木が個の弱い葉ノード W_LN を持ち、最右の二分木が 4 個の弱い葉ノード W_LN を持つ。弱い葉ノード W_LN は白の正方形である。

【 0 0 5 6 】

$p = 2^q + r$ 個の葉ノード L_N を有する最適二分木 OBT の弱い葉ノードの個数 n の、下方の境界は弱い葉ノードの最小個数 m_{min} であり、かつ、上方の境界は弱い葉ノードの最大個数 m_{max} であり、すなわち、 $m_{min} \leq n \leq m_{max}$ であり、このとき、 $r = 0$ ならば $m_{min} = 0$ 、 $r > 0$ ならば $m_{min} = m_{max} \{ 0, 2^q - 2r \}$ 、また、このとき、 $r = 0$ ならば $m_{max} = 0$ 、 $r > 0$ ならば $m_{max} = 2^q - 2 \operatorname{div}(r + 1, 2)$ 、である。

$\operatorname{div}(a, b)$ という表記は、整数 b による整数 a のユークリッド除法の商を表す。

【 0 0 5 7 】

本発明の文脈において、葉ノード L_N はボリュームであり、非葉ノードは二項可換演算の組み合わせボリューム全体である。さらに、各非葉ノードは、その 2 個の入力ソリッドを組み合わせ得られるソリッド形状を格納する。ユーザが宣言素性を作成または削除する度に、新しい葉ノード L_N が二分木に対して追加または除去される。以下では、任意の版のシーケンスで二分木の最適性または理想性を維持するアルゴリズムについて説明する。理想的な二分木概念は、葉ノード削除の際の最適性の喪失を減少させるのに有用である。弱い葉ノード W_LN の個数を減らすことは、二分木が最適性を喪失して、システムが最適性を回復させる可能性を減らすことにもなる。

【 0 0 5 8 】

既存の二分木を考え、また、当該二分木に追加される新しい葉ノード L_N を考えると、

10

20

30

40

50

「理想的な成長アルゴリズム」が、新しい葉ノードを追加すべき最良の既存の葉ノード L_N を計算する。データ構造は、各ノードについて、その部分木の葉ノード L_N の個数を保存する。換言すると、各ノード x は、その部分木の葉ノード L_N の個数を知っており、 (x) で表され、 x が 1 個の葉ノード L_N ならば $(x) = 0$ である。アルゴリズムは、根ノード R_N で開始され、最小個数を有する現在ノードの入力ノードを参照して、ある葉ノード L_N に達するまで繰り返される。

【0059】

```
x:=根ノード
While xが葉ノードでない do begin
  y:=xの左入力ノード
  z:=xの右入力ノード
  If (y) (z) then
    x:=y
  Else
    x:=z
  End if
End while
Return x
```

10

【0060】

図 16 において、新しい葉ノードの最良の場所が矢印で指し示される。アルゴリズムが参照したノードに、それぞれの数が付されている。

20

【0061】

明らかに、新しい葉ノード L_N の最良の場所を見つけるための計算コストは、二分木の深さに比例する。

【0062】

二分木が理想的でない場合でも、先のアルゴリズムが働くことに留意されたい。理想的でない木から開始して、有限個数の追加葉ノードの後に理想性に達する。理想木 IBT で開始され、新しい葉ノード L_N をアルゴリズムにより規定される場所に追加することにより、新しい理想木 IBT が与えられる。

【0063】

30

図 17 には、最適性の回復について示される。

【0064】

弱い葉ノード WLN を最適二分木 OBT から除去すると、生じる二分木はもはや最適ではない。実際、除去演算により、いわゆる「非最適葉ノード」が作成され、これは、深さ $q - 1$ の一意的な葉ノード L_N であり、二分木内の他のどこかに、深さ q の少なくとも 1 個の葉ノード L_N 、および、深さ $q + 1$ の少なくとも 1 つの葉ノード L_N が存在するという事に留意されたい。

【0065】

最適性回復アルゴリズムの入力データは、

- ・適切なデータ構造（後述される）を備える二分木である、と共に
- ・ x で示される、その一意的な非最適葉ノード（へのポインタ）、である。

40

【0066】

アルゴリズムには、3 つのステップが含まれる。すなわち、第 1 のノードであってその部分木が非最適である、ピボットノード PN を見つけること、「 V 」部分木（ V 部分木は、2 個の葉ノードと 1 個の根ノードとを含む部分木である）を見つけて、および、非最適葉ノード L_N を V 部分木で入れ替えること、である。

【0067】

以下のデータ構造が使用される。各ノード s に 2 つの整数が付される。その部分木の最大の深さ $d_{max}(s)$ 、および、その部分木の最小の深さ $d_{min}(s)$ である。二分木が最適であるならば、任意のノード t について、以下の $c(t)$ で示す最適性条件が真

50

となる。

【 0 0 6 8 】

$0 \leq d_{\max}(u) - d_{\min}(v)$ かつ $0 \leq d_{\max}(v) - d_{\min}(u) \leq 1$
 ここで、 u 、 v はノード t の入力ノードである。ノード t が葉ノードならば、 $d_{\max}(t) = d_{\min}(t) = 0$ である。

【 0 0 6 9 】

ピボットノード PN を見つけるステップは、以下のように実現される。ピボットノードファインダアルゴリズムは、非最適葉ノード x で開始され、根ノード RN に向かって進み、 y で示すいわゆるピボットノード PN を見つける。ピボットノード PN は、偽の最適条件を特徴とする経路上にある第 1 のノードである。

10

【 0 0 7 0 】

```
s := x の出力ノード
While c(s) が真 do begin
  s := s の出力ノード
End if
End while
y := s
```

【 0 0 7 1 】

図 18 は、ピボットノード PN 発見処理を示す。見つかったピボットノードは y である。アルゴリズムが参照したノードには、 (d_{\max}, d_{\min}) 値が付されている。

20

【 0 0 7 2 】

最深深さを有する V 部分木を見つけるステップは、以下のように実現される。アルゴリズムは、ピボットノード y で開始され、葉ノードに向かい、最深の V 部分木を見つける。この最深 V 部分木の根ノードには、 z が付される。

【 0 0 7 3 】

```
アルゴリズムは以下の通りである。
s := y
While s が葉ノードでない do begin
  u := s の左入力ノード
  v := s の右入力ノード
  If  $d_{\max}(u) \geq d_{\max}(v)$  then
    s := u
  Else
    s := v
  End if
End while
z := s の出力ノード
```

30

【 0 0 7 4 】

図 19 は、 V 部分木探索を示す。アルゴリズムが参照するノードにそれぞれの d_{\max} 値が付されている。

40

【 0 0 7 5 】

入れ替えのステップが、図 20 に示され、 z で示す V 部分木の根ノードが非最適葉ノード LN で入れ替えられ、これにより、最適性が回復される。

【 0 0 7 6 】

図 21 は、可能性のある最悪の場合を示す。ピボットノード PN が根ノード PN であるということがあり得る。これは、除去された葉ノードが最深葉ノードから「遠く」に位置する場合に起こる。図 21 では、木の左側で葉ノードが先に除去され、非最適葉ノード x が生み出される。最深葉ノードが木の右側に位置するため、ピボットノード y が根ノードとなる。この最悪の場合においても、最適性回復の計算コストは、二分木の深さに比例する。

50

【 0 0 7 7 】

最適性回復では、いくつかのソリッドモデリングのアップデートが要求されて、新しいソリッド形状が得られる、という方法で二分木が認識される。繰り返される演算は、修正されたノードから根ノード R N への経路上に位置される。これらの演算は、図 2 2 では四角で囲んだ「+」である。換言すると、四角で囲んだ「+」で表されるノードは、変化に追隨して再計算すべきノードである。

【 0 0 7 8 】

図 2 3 は、最悪の場合の最適性回復を表す。明らかに、繰り返される「+」演算の個数が、二分木の深さの 2 倍を超えることは不可能である。ここでもまた、計算の複雑性が二分木の深さに比例する。

10

【 0 0 7 9 】

本発明の核心は、二分木の理想性を回復することである。

【 0 0 8 0 】

弱い葉ノード W L N を理想二分木 I B T から除去し、かつ、先のアルゴリズムを介して最適性を回復することにより、理想二分木 I B T を生み出すことができる（理想二分木は最適二分木 O B T より良好である、すなわち、最適二分木では、弱い葉ノードの個数が最小で、弱い葉ノードとは、除去されると得られる二分木がもはや最適ではなくなるという葉ノードである）。

【 0 0 8 1 】

ここで、本発明の裏付けを行う。

20

【 0 0 8 2 】

v を、 $p = 2^q + r$ 個の葉ノード有する最初の理想二分木 I B T の弱い葉ノード W L N とする。次に、3 つの事実に気づかなければならない。第 1 に、葉ノード v の深さは q である。第 2 に、ノード v の兄弟部分木は実際に、 x で示す別の葉ノードである（深さは同様に q ）。そうでなければ、 v の V 形状の兄弟部分木は、 v が除去された後に最適性を保存することになる。第 3 に、最初の二分木のどこかに、深さ $q + 1$ の少なくとも 1 個の葉ノード L N が存在する。このことを、図 2 4 に示す。

【 0 0 8 3 】

葉ノード v を除去した後、その前の兄弟葉ノード x は深さが $q - 1$ に下がり、そのため、最適性が壊れる。このことを、図 2 5 に示す。

30

【 0 0 8 4 】

最適性を回復するために、アルゴリズムは V 部分木を見つけ、その根ノード R N に z が付され、ノード z と x を入れ替える。重要な点は、 z の兄弟木が w で示される葉ノードであり、別の V 部分木ではないことである。（そうでなければ、最初の二分木は最適でないことになる。実際、我々は、ノード z の兄弟部分木が V 部分木であると想定することができる。そして、この V 部分木を、最初の二分木上の葉ノード v または x で入れ替えることにより、弱い葉ノード W L N の個数は減り、一方で、葉ノード L N の総数が保存され、これは最初の二分木の理想性の仮説と相反する。）従って、弱い葉ノード v が除去され、かつ、ノード x と z が入れ替えられた後、2 つの状況が可能となる。深さ $q + 1$ の葉ノード L N がもはや存在しない場合、結果として得られる二分木は完全であり、何故なら全ての葉ノード L N が同じ深さを持つからであり、これは特別な理想二分木である。木のどこかに深さ $q + 1$ の他の葉ノード L N が存在する場合、2 個の新しい弱い葉ノード W L N が作成される。これは、2 個の最初の弱い葉ノード x と v が、今度は 4 個の弱い葉ノードで置き換えられるからであり、この 4 個は、図面においては図 2 6 に示すように、 x 、 w 、および、V 部分木の、 c 、 d で示す 2 個の葉ノードである。

40

【 0 0 8 5 】

ここで、葉ノード L N の総数が $p - 1$ であるため、結果として得られる最適二分木 O B T は理想 I B T である。実際、この文脈において、最初（かつ理想的）の二分木 I B T の弱い葉ノード W L N の個数は、 $\min(p) = 2^q - 2r$ である。弱い葉ノード W L N の除去および最適回復の後、結果として得られる最適二分木 O B T の弱い葉ノード W L N

50

の個数は、 $m_{in}(p) + 2$ である。しかし、

$$\begin{aligned} m_{in}(p) + 2 &= 2^q - 2r + 2 \\ &= 2^q - 2(r - 1) \\ &= m_{in}(p - 1) \end{aligned}$$

これは、結果として得られる最適二分木 OBT は、 $p - 1$ 個の葉ノード LN を持つので、理想 IBT であるということを意味し、裏付けは終了する。

【0086】

図 27 に示すように、理想二分木 IBT の弱い葉ノードではないノードを削除すると、最適性が喪失する（最適性は保存される）ことに留意されたい。最左の二分木の指し示される葉ノードを削除することにより、中央の木が生み出される。この木の弱い葉ノードは、白い正方形である。これは最適であるが、理想的ではなく、何故なら、同じ個数の葉ノードを持つ理想二分木は、弱い葉ノード WLN を持たないからである。

10

【0087】

理想的な成長アルゴリズムでは、十分な個数の葉ノードが追加された後に、理想性が回復される。さらに、従前には最適であったが理想的ではなかった二分木の最適性を回復することにより、概して、理想二分木 IBT は生み出されない。換言すると、本発明では、二分木の最適性を保存する。理想性は成長および最適性の回復を通じて保存される。

【0088】

図 28 は、二分木の全ての可能性のあるステータスを捕えたものであり、理想二分木、最適（理想的ではない）二分木、および、非最適二分木、が含まれる。また、二分木に対して行われる演算を介した全ての可能性のあるステータスの変化も合わせて描かれ、葉ノード LN の追加（理想的な成長アルゴリズムによる）、弱い葉ノード WLN の除去、弱い葉ノード以外の葉ノード除去、および、最適性の回復、が含まれる。非最適のステータスは、一時的、であり、最適性が喪失するとすぐにアルゴリズムが最適性を瞬時に回復することを意味する。

20

【0089】

先の説明によると、最適二分木 OBT に対して行われる全ての演算は、葉ノード LN の個数の対数である、木の深さに比例する。これらの演算は、葉ノードの除去、葉ノードの追加、最適性の回復、またはソリッドのアップデート、である。反対に、線形リストに対して行われる演算は、葉ノードの総数に主に比例する。全ての基本ボリュームが、修正または削除される確立が同一である、という工業的な仮説の下で、以下の図では、最適二分木と番号付きリストとの間の平均計算成長を比較する。図 29 に示すように、二分木を選択する方が、番号付きリストに比べて明らかに有利である。

30

【0090】

葉ノード LN の個数と $p = 2^q + r$ で示すと、最適二分木 OBT の平均深さは以下の通り。

【0091】

【数 1】

$$h_{opt}(p) = q + \frac{2r}{p}$$

40

【0092】

番号付きリストの平均深さは以下の通り。

【0093】

【数 2】

$$h_{list}(p) = \frac{p+1}{2} - \frac{1}{p}$$

【0094】

適切なデータ構造は、二分木が各ノードに追加的な情報を格納しているものである。二

50

分木のデータ構造、ポインタ、およびナビゲーション機能は、当該技術において標準的である。これを超えて、本発明では、各ノードが以下の数値データを提供することができる、ことを必要とする。すなわち、ノードの部分木の最大の深さ、ノードの部分木の最小の深さ、およびノードの部分木の葉ノード L_N の個数、である。これらのデータは、演算（葉ノードの除去、葉ノードの追加、最適性の回復）により、最大でも深さに比例する程度のコストで、最新に維持される。アルゴリズムの全体的な対数計算の複雑性は保持される。換言すると、データ構造操作は複雑性を上昇させない。

【0095】

最適二分木 OBT の以下の特性は、実装を支援することができる。第1の点として、最適二分木の全ての部分木が、同様に最適二分木である。第2の点として、 $p = 2^q + r$ を最適二分木の葉ノード L_N の個数、 $0 \leq r < 2^q$ 、とすると、深さ q の葉ノードの個数は $2^q - r$ 、深さ $q + 1$ の葉ノードの個数は $2r$ である。よって、最適二分木の最大の深さおよび最小の深さは、葉ノードの個数から簡単に計算することができる。 p を与えて、 q および r を、 $p = 2^q + r$ かつ $0 \leq r < 2^q$ 、となるように計算する。 $r = 0$ ならば、 $d_{min} := q$ かつ $d_{max} := q$ かつ $d_{max} := q + 1$ である。

【0096】

この節では、図30に示すように、従来技術の記載に従って設計されたある一部分を（各入力バケットが線形リストを用いて入力される。図30の左図）、本発明のデータ形式に変換する（各入力バケットが理想二分木 IBT を用いて入力される。図30の右図）ことについて説明する。

【0097】

コンバータが、各入力バケットに対して、理想化アルゴリズムを実行し、これは、基本ボリユームの線形リストを、葉ノード L_N が当該基本ボリユームである理想二分木 IBT に変換することである。そして、コンバータは、その一部分のソリッドモデリングのアップデートを実行し、これにより、同一のソリッド形状が生み出される。

【0098】

理想化アルゴリズムでは、理想二分木を以下のように段階的に作成する：

新しい二分木を起動（根ノードまで縮小された空の二分木）

For 線形リストの各基本ボリユーム v do begin

 新しい葉ノード v を二分木に追加

End for

別のさらに速い理想化アルゴリズムでは、葉ノード個数の関連する範囲、典型的には、 $p = 2$ の葉ノードから $p = 50$ の葉ノードまで、に従って予め計算した理想二分木 IBT のライブラリが含まれる。各入力バケットに対して、本方法では、その線形リストの基本ボリユームの個数を読み込む、この葉ノード個数に対応する理想二分木の構造をロードする、および、線形リストの基本ボリユームを二分木の葉ノードに設定する、ことが含まれる。

【0099】

アルゴリズムの選択は、変換がどの程度大きいか、および、どの程度の頻度が、で決まる。

【0100】

データ変換の第1の使用は、以前のバージョンの CAD システム（本発明を実装していない）を使用して設計されたモデルを、本発明を実装する CAD システムのバージョンと互換性のあるモデルに変換することである。その場合、二分木は、非最適二分木を最適二分木 OBT に変換することにより得られる。

【0101】

データ変換の別の使用は、作成後の性能を向上させることである。

【0102】

図31は、本発明を実装することができるコンピュータネットワークまたは同様のデジタル処理環境を示す。

【 0 1 0 3 】

クライアントコンピュータ / 装置 C L およびサーバコンピュータ S V が、アプリケーションプログラム等を実行する処理装置、記憶装置、および、入出力装置を提供する。クライアントコンピュータ / 装置 C L はまた、通信ネットワーク C N E T を介して、他のクライアント装置 / 処理 C L およびサーバコンピュータ S V を含む他のコンピューティング装置にリンクされる。通信ネットワーク 7 0 は、リモートアクセスネットワーク、グローバルネットワーク（例えば、インターネット）、世界規模のコンピュータ収集、ローカルエリアネットワークまたはワイドエリアネットワーク、およびゲートウェイの一部であってよく、これらはそれぞれのプロトコルを現下で使用してお互いに通信を行う。他の電子装置 / コンピュータネットワークアーキテクチャも適切である。

10

【 0 1 0 4 】

図 3 2 は、図 3 1 のコンピュータシステムにおける、コンピュータ（例えば、クライアントプロセッサ / 装置 C L またはサーバコンピュータ S V ）の内部構造の図である。各コンピュータ C L 、 S V は、システムバス S B を含み、バスは、コンピュータまたは処理システムの構成要素間でデータ転送に使用されるハードウェアの線のセットである。バス S B は、本質的には、コンピュータシステム異なる要素（例えば、プロセッサ、ディスク記憶装置、メモリ、入出力ポート、ネットワークポートなど）を接続する共有導管であり、要素間の情報の転送を可能にするものである。

【 0 1 0 5 】

システムバス S B に付随するものには、種々の入力装置および出力装置（例えば、キーボード、マウス、ディスプレイ、プリンタ、スピーカ等）をコンピュータ C L 、 S V に接続する I / O 装置インターフェース D I がある。ネットワークインターフェース N I により、コンピュータはネットワーク（例えば、図 3 1 のネットワーク C N E T ）に付随する種々の他の装置に接続することができる。

20

【 0 1 0 6 】

メモリ M E M は、本発明の実施形態を実装するために使用されるコンピュータソフトウェア命令 S I およびデータ C P P のための、揮発性記憶装置を提供する（例えば、第 1 のパスビルダ P B 、第 2 のパスを計算するための手段 C M 、図 1 から 3 0 において検討された方法を実装するアップデーター U D 、および、上記で詳述したサポーティングコード）。

30

【 0 1 0 7 】

ディスク記憶装置 D S は、本発明の実施形態を実装するために使用されるコンピュータソフトウェア命令 S I およびデータ D A T のための、不揮発性記憶装置を提供する。中央処理装置（C P U ）もまた、システムバス S B に付随し、コンピュータ命令の実行を提供する。

【 0 1 0 8 】

一実施形態において、プロセッサルーチン S I およびデータ D A T は、コンピュータプログラム製品（一般的には、C P P と呼ばれる）であり、これにはコンピュータ可読媒体（例えば、1 つまたは複数の D V D - R O M 、 C D - R O M 、 ディスク、テープなどの着脱可能記憶媒体）が含まれ、本発明のシステムのソフトウェア命令の少なくとも一部を提供する。コンピュータプログラム製品 C P P は、従来技術において周知であるように、任意の適切なソフトウェアインストールプロシージャによりインストールされる。

40

【 0 1 0 9 】

別の実施形態において、ソフトウェア命令の少なくとも一部は、ケーブル、通信、および / または無線接続を介してダウンロードすることもできる。他の実施形態において、本発明のプログラムは、伝播媒体上の伝播信号上で具現化されるコンピュータプログラム伝播信号製品 S P である（例えば、無線波、赤外線波、レーザ波、音波、または、インターネットもしくは他のネットワークなどのグローバルネットワーク上を伝播される電波）。そのような搬送媒体または信号は、本発明のルーチン / プログラム C P P のためのソフトウェア命令の少なくとも一部を提供する。

50

【 0 1 1 0 】

代替実施形態において、伝播信号は、伝播媒体上で搬送されるアナログ搬送波またはデジタル信号である。例えば、伝播信号は、グローバルネットワーク（例えば、インターネット）、電気通信網、または他のネットワーク上を伝播される、デジタル化信号であってよい。

【 0 1 1 1 】

一実施形態において、伝播信号は、ある時間をかけて伝播媒体上を送信される信号であり、例えば、ミリ秒、秒、またはもっと長い時間をかけてネットワーク上を送信されるソフトウェアアプリケーションのための命令などである。

【 0 1 1 2 】

別の実施形態において、コンピュータプログラム製品ＣＰＰのコンピュータ可読媒体は、コンピュータシステムＣＬが、伝播媒体を受信すること、および、上述したように、コンピュータプログラム伝播信号製品のための伝播媒体において具現化される伝播信号を識別すること、により、受けとりかつ読むことができる伝播媒体である。

【 0 1 1 3 】

一般的に言えば、用語「搬送媒体」すなわち一時的な搬送、には、前述の一時的な信号、伝播信号、伝播媒体、記憶媒体などが包含される。

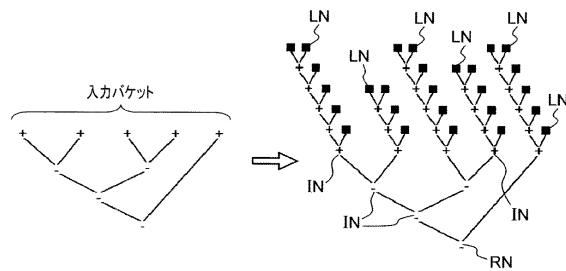
【 0 1 1 4 】

本発明を、その例示の実施形態を参照することで特に示して説明したが、当業者においては、添付の請求項により包含される本発明の範囲を逸脱することなく、形式および詳細の種々の変更がその中においてなされ得ることを理解するであろう。

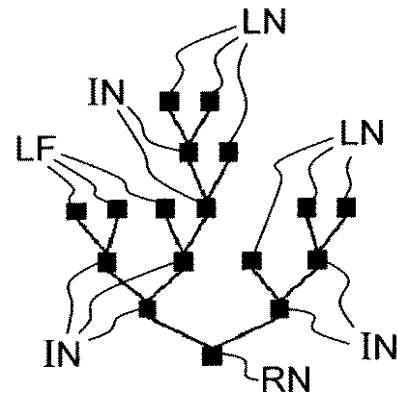
10

20

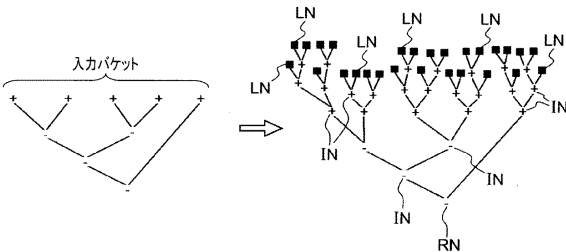
【 図 1 】



【 図 4 】



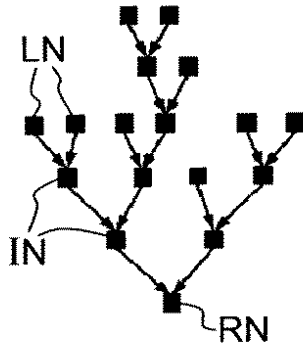
【 図 2 】



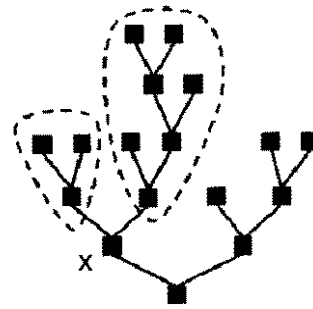
【 図 3 】



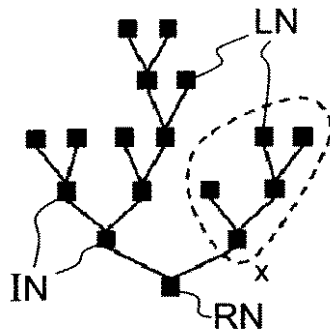
【図 5】



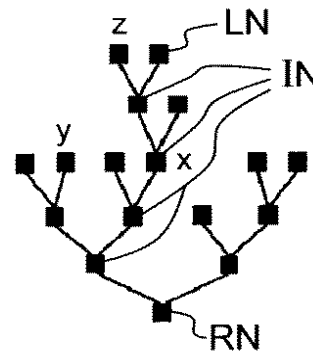
【図 7】



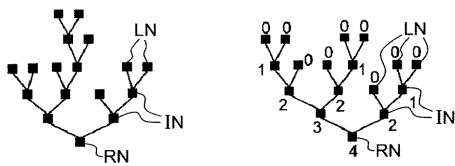
【図 6】



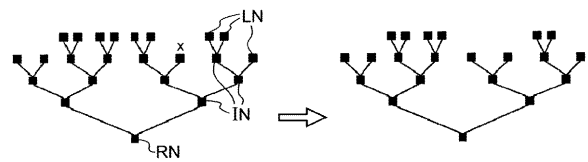
【図 8】



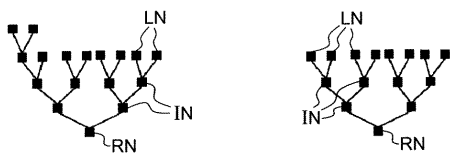
【図 9】



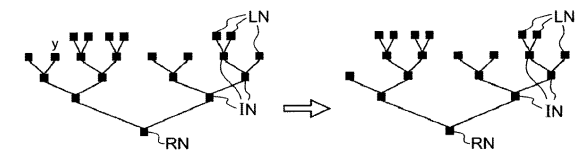
【図 13】



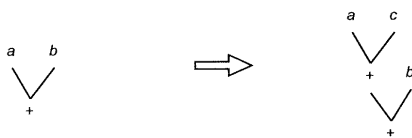
【図 10】



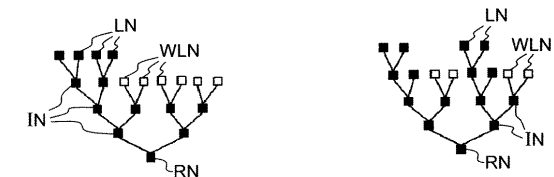
【図 14】



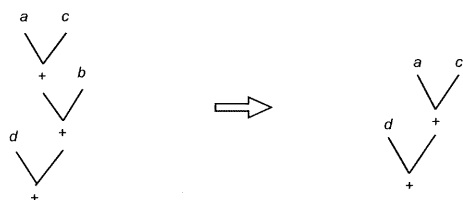
【図 11】



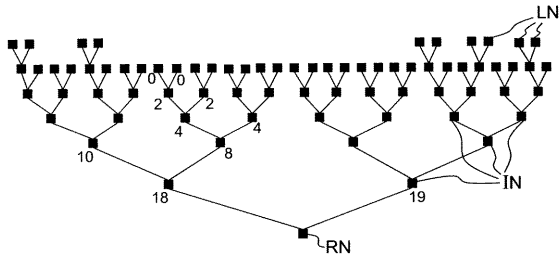
【図 15】



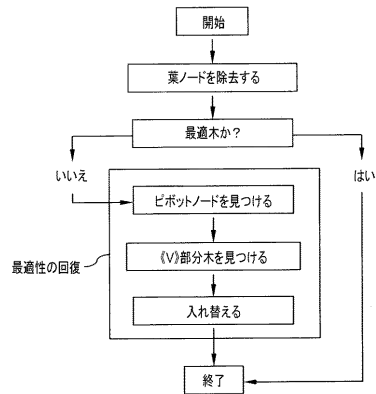
【図 12】



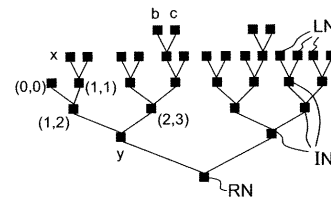
【図 16】



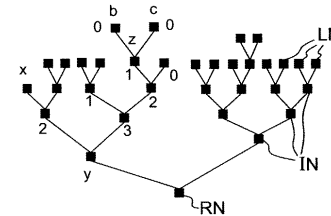
【図 17】



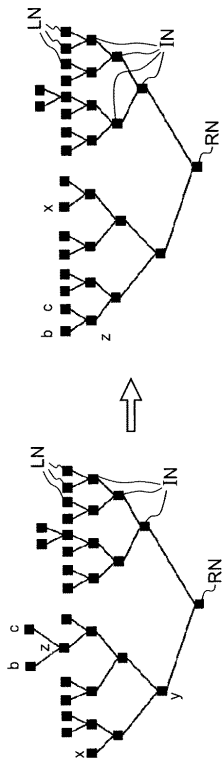
【図 18】



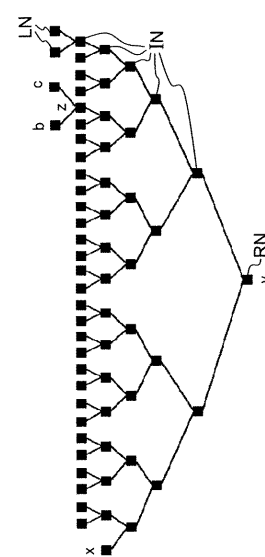
【図 19】



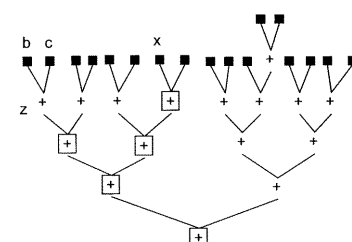
【図 20】



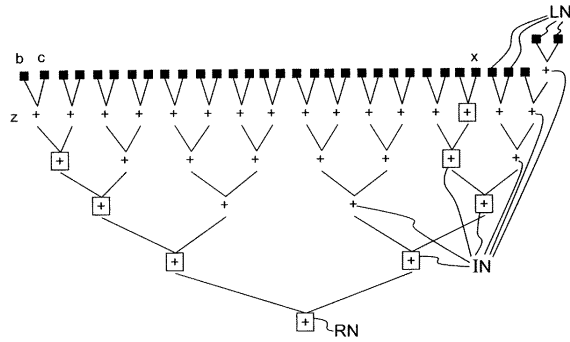
【図 21】



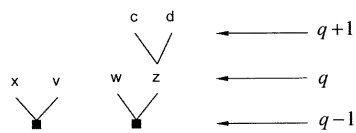
【図 22】



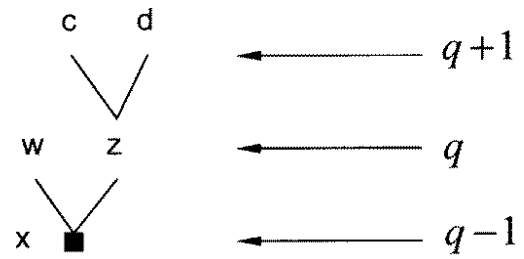
【図 23】



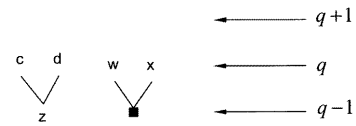
【図 24】



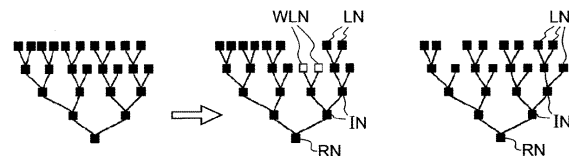
【図 25】



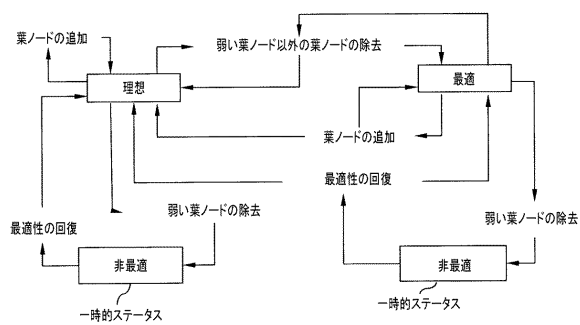
【図 26】



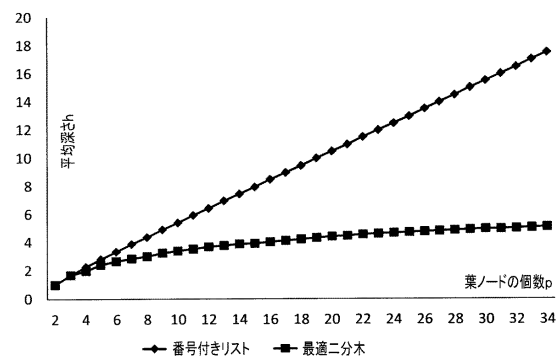
【図 27】



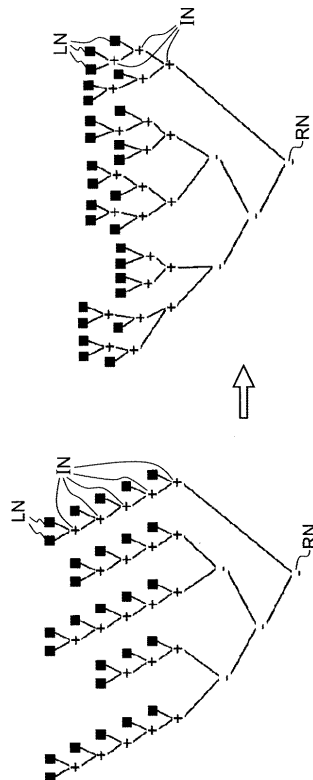
【図 28】



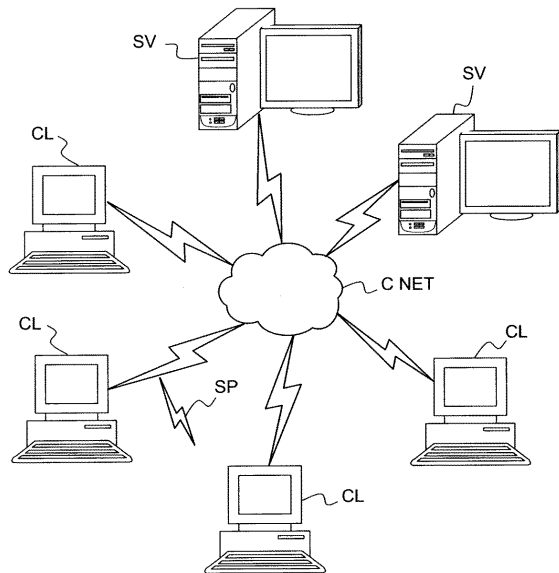
【図 29】



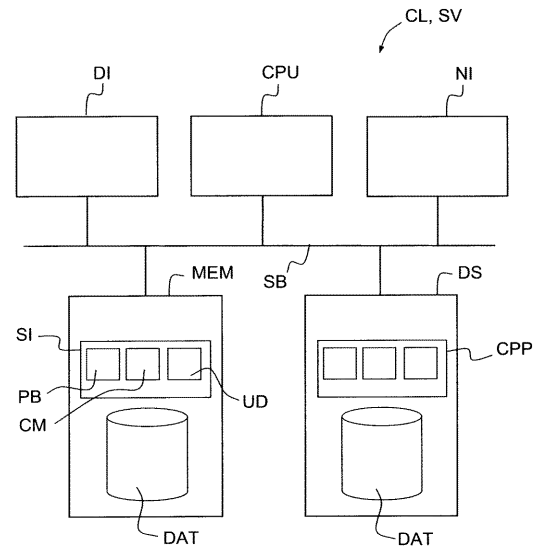
【図 30】



【図 3 1】



【図 3 2】



フロントページの続き

(56)参考文献 特開平 0 8 - 0 0 6 9 8 2 (J P , A)
特開平 0 9 - 1 8 5 7 2 7 (J P , A)
特開 2 0 0 3 - 1 6 2 5 5 0 (J P , A)
特開 2 0 0 5 - 0 4 4 3 6 5 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
G 0 6 Q 1 0 / 0 0 - 9 9 / 0 0 ,
G 0 6 F 1 7 / 5 0