(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-9156
(P2010-9156A)
(43)公開日　平成22年1月14日(2010.1.14)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 17/30 (2006.01)	G06F 17/30 419A	5B075

審査請求 有　請求項の数 7　ＯＬ　（全 17 頁）

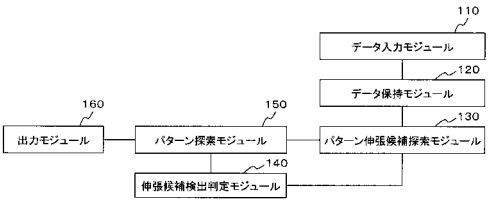
(21)出願番号　特願2008-165216 (P2008-165216)
(22)出願日　平成20年6月25日 (2008.6.25)

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂九丁目７番３号
(74)代理人　100115129
弁理士　清水　昇
(74)代理人　100102716
弁理士　在原　元司
(74)代理人　100122275
弁理士　竹居　信利
(72)発明者　林　千登
神奈川県足柄上郡中井町境４３０　グリーンテクなかい　富士ゼロックス株式会社内

Ｆターム(参考)　5B075 ND35

(54)【発明の名称】情報処理装置及び情報処理プログラム

(57)【要約】
【課題】複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減するようにした情報処理装置を提供する。
【解決手段】情報処理装置のパターン探索手段は、パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行い、伸張候補探索手段は、前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索し、伸張候補検出判定手段は、前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定し、判定結果保持手段は、前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を保持し、前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行う。
【選択図】図１



【特許請求の範囲】
【請求項１】
パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、
前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、
前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、
前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を保持する判定結果保持手段
を具備し、
前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行う
ことを特徴とする情報処理装置。
【請求項２】
パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、
前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、
前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、
前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を、前記パターンの候補が出現する木構造ごとに保持する判定結果保持手段
を具備し、
前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行う
ことを特徴とする情報処理装置。
【請求項３】
前記伸張候補検出判定手段は、
前記パターンの候補上のノードに新たに追加する子ノードの候補がなかったときに、発見可能性を無しと判定する
ことを特徴とする請求項１又は２に記載の情報処理装置。
【請求項４】
前記伸張候補検出判定手段は、
前記パターンの候補上のリーフノードに子ノードとして複数の候補がある、又は、一つ候補があり各木構造に出現箇所が複数あるものが予め指定された数以上あるときに、発見可能性をありと判定する
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情報処理装置。
【請求項５】
前記伸張候補探索手段は、
ノードの候補の探索をより上位のノードから行い、下位のノードでの処理は、上位のノードの探索によって更新された発見可能性の判定結果を利用して探索を行う
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情報処理装置。
【請求項６】
コンピュータを、
パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、

前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、
前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、
前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を保持する判定結果保持手段
として機能させ、
前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行う
ことを特徴とする情報処理プログラム。

【請求項７】
コンピュータを、
パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、
前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、
前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、
前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を、前記パターンの候補が出現する木構造ごとに保持する判定結果保持手段
として機能させ、
前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行う
ことを特徴とする情報処理プログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、情報処理装置及び情報処理プログラムに関する。
【背景技術】
【０００２】
コンピュータの処理能力、記憶装置の容量の飛躍的な増大に加え、ＩＴ化やネットワーク化が進んだことで大量な情報が容易に集められるようになってきた。集めた情報から市場機会やリスクに関する情報を早期に発見したり、隠れた知識を発見したりすることへの期待が高まっている。
しかし、集めた情報の量はしばしば人間の処理能力をはるかに超えるものとなる。このため、せっかく大量に集めた情報からリスクを発見したり、知識を抽出したりして活用することは実際には労力を伴う難しいものであった。
【０００３】
一方、パターン・マイニング等の技術の進展により、そのような大量の情報の中から例えば同時に購入される商品のパターンなどの情報が抽出可能となってきた。同時に購入される品物のパターンや購入される順序のパターンを抽出する技術が顧客の購買行動の分析などの需要から注目を集めて研究開発されてきたが、最近ではさまざまな情報の構造化、半構造化が進んできたこともあり、木構造のような構造を持つパターンを抽出するパターン・マイニングの技術が注目されてきている。構造情報を抽出するパターン・マイニングの技術の中でも、特に木構造はＸＭＬ（ｅＸｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ）をはじめとしてドキュメントの構造化や知識表現などさまざまな情報の構造化に用いられるためパターン抽出への期待も大きい。
【０００４】
木構造のデータ群から部分木のパターンを抽出する技術には大きく分けて、親子関係が

厳密に一致する構造だけを抽出するｉｎｄｕｃｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇの技術と、親子関係が多少乱れても先祖－子孫の関係があれば構造を抽出するｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇの技術がある。

現実社会で発生するデータ、例えば電子文書であるドキュメントの操作履歴などのように人の操作を記録したものでは、たとえ同じ内容の作業を行っても、作業者の操作順などが厳密には一致しないため操作の履歴データの親子関係が厳密に一致することは期待できない。このため、そのようなデータからパターンを抽出するためには、親子関係に揺れが生じたとしてもパターンが抽出できるｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇの技術を適用することが望ましい。また、人の操作の記録だけではなく、情報を整理した木構造データのようなものからも埋め込まれている隠れた構造を抽出するためには同様にｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇを用いる必要がある。

ｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇを実現する従来の技術として、例えばＴｒｅｅＭｉｎｅｒ、Ｄｒｙａｄｅ、ＭＢ３－ｍｉｎｅｒ、ＴＲＩＰＳ、ＰｒｅｆｉｘＴｒｅｅＥＳｐａｎなどの技術が開示されている。

【０００５】

これらに関連する技術として、例えば、特許文献１には、データの集合からその中に含まれる重要なパターンを検出する方法及びシステムを提供することを課題とし、木構造データで表されたデータ集合を含むデータベースから、集計対象となる候補パターンを用いて、頻出パターンを検出するシステムであって、（１）データベースから候補パターンにマッチするパターンを集計する手段と、（２）前記集計により出現頻度の高いパターンを検出する手段と、（３）前記検出したパターンから、次の集計対象となる候補パターンを生成する手段と、を有するように構成することが開示されている。

【０００６】

また、例えば、特許文献２には、順序木において頻出するパターンを抽出するのに好適な抽出装置等を提供することを課題とし、抽出装置の入力受付部は、一つ以上の順序木の入力を受け付け、変換部は、入力が受け付けられた順序木のそれぞれを系列表現へ変換し、抽出部は、変換された系列表現のそれぞれが含むパターンのうち、所定の頻度以上で出現するパターンを抽出し、系列表現は、順序木を深さ優先探索して、枝を進む際に通過する節はその名前を表すマークを、枝を戻る際はバックトラックマークを、それぞれ並べることによりでき、パターンは、系列表現であるマークの列中の名前を表すマークのいずれかを最初のマークとして、これから射影を０回以上繰り返したときに、最初のマークから最後のマークに至るまでに出会うマークの列をいい、射影が成立するか否かは、マークの列の列文脈と、射影文脈の値により判定することが開示されている。

【特許文献１】特開２００１－１３４５７５号公報

【特許文献２】特開２００４－３５５４５７号公報

【発明の開示】

【発明が解決しようとする課題】

【０００７】

本発明は、複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減するようにした情報処理装置及び情報処理プログラムを提供することを目的としている。

【課題を解決するための手段】

【０００８】

かかる目的を達成するための本発明の要旨とするところは、次の各項の発明に存する。

請求項１の発明は、パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を保持する判定結果

保持手段を具備し、前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行うことを特徴とする情報処理装置である。

【０００９】

請求項２の発明は、パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を、前記パターンの候補が出現する木構造ごとに保持する判定結果保持手段を具備し、前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行うことを特徴とする情報処理装置である。

【００１０】

請求項３の発明は、前記伸張候補検出判定手段は、前記パターンの候補上のノードに新たに追加する子ノードの候補がなかったときに、発見可能性を無しと判定することを特徴とする請求項１又は２に記載の情報処理装置である。

【００１１】

請求項４の発明は、前記伸張候補検出判定手段は、前記パターンの候補上のリーフノードに子ノードとして複数の候補がある、又は、一つ候補があり各木構造に出現箇所が複数あるものが予め指定された数以上あるときに、発見可能性をありと判定する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情報処理装置である。

【００１２】

請求項５の発明は、前記伸張候補探索手段は、ノードの候補の探索をより上位のノードから行い、下位のノードでの処理は、上位のノードの探索によって更新された発見可能性の判定結果を利用して探索を行う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情報処理装置である。

【００１３】

請求項６の発明は、コンピュータを、パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を保持する判定結果保持手段として機能させ、前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行うことを特徴とする情報処理プログラムである。

【００１４】

請求項７の発明は、コンピュータを、パターンの候補に複数の木構造内で複数回現れるラベルのノードを追加して、該パターンの候補を伸張することによって、該複数の木構造内で複数回現れるパターンの探索を行うパターン探索手段と、前記パターン探索手段によって伸張される前記パターンの候補のノードの候補を探索する伸張候補探索手段と、前記伸張候補探索手段によって探索されたノードの伸張候補の発見可能性を判定する伸張候補検出判定手段と、前記伸張候補検出判定手段によって判定された伸張候補の発見可能性を、前記パターンの候補が出現する木構造ごとに保持する判定結果保持手段として機能させ、前記パターン探索手段は、前記判定結果保持手段に保持された発見可能性の判定結果を用いて、パターンの探索を行うことを特徴とする情報処理プログラムである。

【発明の効果】

【００１５】

請求項１記載の情報処理装置によれば、複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減することができる

。
【００１６】
請求項２記載の情報処理装置によれば、複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減することができる。
【００１７】
請求項３記載の情報処理装置によれば、兄弟のノードを探索する場合に、存在しない枝を探索してしまう無駄な処理を削減することができる。
【００１８】
請求項４記載の情報処理装置によれば、子孫のノードを探索する場合に、存在しない枝を探索してしまう無駄な処理を削減することができる。
【００１９】
請求項５記載の情報処理装置によれば、本構成を有していない場合に比較して、より効率よくパターンの探索を行うことができる。
【００２０】
請求項６記載の情報処理プログラムによれば、複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減することができる。
【００２１】
請求項７記載の情報処理プログラムによれば、複数の木構造内で複数回現れるパターンの探索を行う場合において、存在しない枝を探索してしまう無駄な処理を削減することができる。
【発明を実施するための最良の形態】
【００２２】
まず、前述のｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇと、それを実現する従来の技術であるＴｒｅｅＭｉｎｅｒ、Ｄｒｙａｄｅ、ＭＢ３－Ｍｉｎｅｒ、ＴＲＩＰＳ、ＰｒｅｆｉｘＴｒｅｅＥＳｐａｎの技術について説明する。
【００２３】
ｅｍｂｅｄｄｅｄ　ｓｕｂＴｒｅｅ　ｍｉｎｉｎｇは、ｉｎｄｕｃｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇよりもパターンの抽出能力が高い反面、必要となる処理もメモリ量も膨大となるため、高価な計算リソースが必要となりさらにはそのような計算リソースを投入しても意味のあるパターンを抽出する前に処理が続行不可能となってしまう事態も珍しくなかった。
【００２４】
従来技術のうち、Ｄｒｙａｄｅは、出現数の同じ場合にはより大きなパターンだけを出力するというｃｌｏｓｅｄ　ｐａｔｔｅｒｎの抽出を実現するが、兄弟ノードに同じものを含めないという機能制限があり、そのような場面が頻出するドキュメントの操作履歴などの多くの現実のデータからのパターンの抽出には適さなかった。
【００２５】
ＭＢ３－Ｍｉｎｅｒは、同じ数のノードを含むパターン全てを使って、もう一つノードの数が多いパターンを求めるという横型探索の方法を採用している。ｅｍｂｅｄｄｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇではパターンの数は膨大なものとなるので、ＭＢ３－Ｍｉｎｅｒでは現実的な計算リソースを用いて抽出できるパターンの大きさが制限を受けてしまう。
【００２６】
ＴｒｅｅＭｉｎｅｒ、ＴＲＩＰＳは、見つけたパターン中の一つのリーフノードからルートノードにいたるパス上のノードに一つの新しいノードを加えたパターンを探すことを繰り返して頻出パターンの探索を実行する。これらはツリーデータの各ノードを深さ優先探索で辿って系列データとみなしたデータを用いて処理を行い、前記の選んだリーフノードよりその系列データ上で後ろにあるノードを調べることで次にパターンに加えるノード

を探索する。
しかし、パターン上のあるノードに対して新たな子供ノードとして加えることができるノードはｅｍｂｅｄｄｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇの場合にはｉｎｄｕｃｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇの場合に比べて格段に多くなってしまう上、パターンが大きくなって前記のリーフノードからルートノードにいたるパスが長くなってくると、一つのパス上のノードに付け加えられる可能性のあるノードの種類は膨大なものになってしまう。

【００２７】
さらに、パターンを伸張しながら探索していくパターン抽出処理は、再帰的に繰り返し実行される。このため、繰り返し処理が行われるたびに、パス上のノードに付け加えられる可能性のあるノードの種類と出現場所等の膨大な量の情報を、スタックのような記憶部に保存を重ねたまま処理が進んでいくことになる。このため、膨大なメモリ量を必要とすることになり、現実的な計算機の環境下では大きなパターンの抽出処理が行えないという問題が生じていた。

【００２８】
これに対して、ＰｒｅｆｉｘＴｒｅｅＥＳｐａｎではパターンを伸張するパス上の各ノードごとに、伸張した場合の射影データベースを作成して抽出処理を行っていく方法をとっている。これにより、前記の問題は緩和される。
しかし、パターンを伸張するパス上の各ノードごとに、その箇所から伸張できるノードを探索する処理が必要となってしまう。すなわち、パターン上の指定されたノードの子供ノードとなりえるツリーデータ内の特定の位置に出現する頻出ラベルを見つける必要が生じることになる。この処理は相応の負荷がかかる処理となる。

【００２９】
そして、例えば全てのツリーデータが枝の無い長い一本のパスからなっていたような場合にも、各パターンのパス上の各ノードについてパターン上で枝が生じる可能性があるノードを探索することを繰り返す必要が生じてしまい、膨大な回数の無駄な処理が発生してしまう。この無駄な処理は枝の全く無いデータのみで生じるのではなく、ある程度構造が決まっているツリーでも、例えば特定のラベルのノードが出現したあとは兄弟ノードは絶対に生じないというような場合にもこの無駄が発生する。すなわち、ありえない枝を探索する無駄な処理が生じる。
本実施の形態は、ツリーの分布に偏りがあるようなツリーデータを対象としたような場合であっても、前述した無駄な処理を削減して効率的にｅｍｂｅｄｄｅｄ　ｓｕｂｔｒｅｅ　ｍｉｎｉｎｇを実現するものである。

【００３０】
以下、図面に基づき本発明を実現するにあたっての好適な一実施の形態の例を説明する。
図１は、本実施の形態の構成例についての概念的なモジュール構成図を示している。
なお、モジュールとは、一般的に論理的に分離可能なソフトウェア（コンピュータ・プログラム）、ハードウェア等の部品を指す。したがって、本実施の形態におけるモジュールはコンピュータ・プログラムにおけるモジュールのことだけでなく、ハードウェア構成におけるモジュールも指す。それゆえ、本実施の形態は、コンピュータ・プログラム、システム及び方法の説明をも兼ねている。ただし、説明の都合上、「記憶する」、「記憶させる」、これらと同等の文言を用いるが、これらの文言は、実施の形態がコンピュータ・プログラムの場合は、記憶装置に記憶させる、又は記憶装置に記憶させるように制御するの意である。また、モジュールは機能にほぼ一対一に対応しているが、実装においては、１モジュールを１プログラムで構成してもよいし、複数モジュールを１プログラムで構成してもよく、逆に１モジュールを複数プログラムで構成してもよい。また、複数モジュールは１コンピュータによって実行されてもよいし、分散又は並列環境におけるコンピュータによって１モジュールが複数コンピュータで実行されてもよい。なお、一つのモジュールに他のモジュールが含まれていてもよい。また、以下、「接続」とは物理的な接続の他

、論理的な接続（データの授受、指示、データ間の参照関係等）の場合にも用いる。
また、システム又は装置とは、複数のコンピュータ、ハードウェア、装置等がネットワーク（一対一対応の通信接続を含む）等の通信手段で接続されて構成されるほか、一つのコンピュータ、ハードウェア、装置等によって実現される場合も含まれる。「装置」と「システム」とは、互いに同義の用語として用いる。「所定」という用語は、予め定められたの意の他に、そのときの状況・状態に応じて、又はそれまでの状況・状態に応じての意を含めて用いる。

【００３１】

本実施の形態は、図１に示すように、データ入力モジュール１１０、データ保持モジュール１２０、パターン伸張候補探索モジュール１３０、伸張候補検出判定モジュール１４０、パターン探索モジュール１５０、出力モジュール１６０を有している。
データ入力モジュール１１０は、データ保持モジュール１２０と接続されており、図示しない入力装置から木構造情報を含むデータ又は木構造に構成できる情報を含むデータを受け取り、その受け取った木構造データを本実施の形態による処理に適した形式のデータに変換してデータ保持モジュール１２０に送る。
データ保持モジュール１２０は、データ入力モジュール１１０、パターン伸張候補探索モジュール１３０と接続されており、データ入力モジュール１１０から木構造データや付随した情報を受け取り保持する。また、パターン伸張候補探索モジュール１３０の要求に応えてツリーデータに関する情報（以下、単にツリーデータともいう）を返す。

【００３２】

パターン伸張候補探索モジュール１３０は、データ保持モジュール１２０、伸張候補検出判定モジュール１４０、パターン探索モジュール１５０と接続されており、パターン探索モジュール１５０の指示に従って所定の条件にあうパターン伸張候補の探索を、データ保持モジュール１２０と通信して行う。この結果をパターン探索モジュール１５０に返すと共に、伸張候補検出判定モジュール１４０に送信し、また、データ保持モジュール１２０に保持させる。つまり、パターン伸張候補探索モジュール１３０は、パターン探索モジュール１５０によって伸張されるパターン候補（以下、単にパターンともいう）のノード候補を探索する。ここで伸張とは、現在のパターン候補内のあるノードから別のノードへの枝を発生させることであり、その伸張によってパターン候補は新たなパターン候補になる。また、ノード候補の探索をより上位のノードから行い、下位のノードでの処理は、上位のノードの探索によって更新された発見可能性の判定結果を利用して探索を行うようにしてもよい。

【００３３】

伸張候補検出判定モジュール１４０は、パターン伸張候補探索モジュール１３０、パターン探索モジュール１５０と接続されており、パターン伸張候補探索モジュール１３０からの情報を受け付けて、パターン探索モジュール１５０が指定した条件にて伸張候補が検出できるか否かを判定し、図示しない記憶装置にその判定結果を記憶し、パターン探索モジュール１５０にその記憶手段へアクセスするための情報を送信する。
つまり、伸張候補検出判定モジュール１４０は、パターン伸張候補探索モジュール１３０によって探索されたノードの伸張候補の発見可能性を判定する。ここで発見可能性とは、対象としているノードにおいて、伸張可能なものであるか否かを示すものであり、具体的には、２値（伸張可能、伸張不可能）を示す符号、その中間値を表す数値等であってもよい。ここでの記憶装置は、伸張候補検出判定モジュール１４０によって判定された伸張候補の発見可能性を保持する。また、その記憶装置は、伸張候補検出判定モジュール１４０によって判定された伸張候補の発見可能性を、パターン候補が出現するツリーデータごとに保持するようにしてもよい。
例えば、伸張候補検出判定モジュール１４０は、兄弟ノードに対する判定処理として、パターン候補上のノードに新たに追加する子ノードの候補がなかったときに、発見可能性を無しと判定する。そして、他の場合には発見可能性をありと判定する。また、伸張候補検出判定モジュール１４０は、子孫ノードに対する判定処理として、パターン候補上のリ

ーフノードに子ノードとして複数の候補がある、または一つ候補があり各ツリーデータに出現箇所が複数あるものが予め指定された数以上あるときに、発見可能性をありと判定する。そして、他の場合には発見可能性を無しと判定する。

【００３４】

パターン探索モジュール１５０は、パターン伸張候補探索モジュール１３０、伸張候補検出判定モジュール１４０、出力モジュール１６０と接続されており、パターン伸張候補探索モジュール１３０を制御してパターンを抽出して、抽出したパターンを出力モジュール１６０に送信する。この過程で、伸張候補検出判定モジュール１４０から受け取った情報をもとに、処理中のパターンの伸張箇所での伸張候補の検出可能性の情報を参照し、パターン伸張候補探索モジュール１３０の起動の有無を制御する。つまり、パターン探索モジュール１５０は、パターン候補に複数のツリーデータ内で複数回現れるラベルのノードを追加して、そのパターン候補を伸張することによって、その複数のツリーデータ内で複数回現れるパターンの探索を行う。そして、伸張候補検出判定モジュール１４０の記憶装置に保持された発見可能性の判定結果を用いて、パターンの探索を行う。

【００３５】

出力モジュール１６０は、パターン探索モジュール１５０と接続されており、パターン探索モジュール１５０から受け取ったパターン情報を出力する。この出力は可視化やテキスト化などの整形を経て、ディスプレイ装置や印刷装置等に送る場合もあるが、予め定めたフォーマットに必要なら整形してファイルやデータベース等に格納してもよい。

【００３６】

図２は、本実施の形態による処理例を示したフローチャートである。

処理を開始すると、頻出ラベルを列挙する（ステップＳ２０２）。つまり、データ保持モジュール１２０内の複数のツリーデータから頻出ラベルを捜索する。

そして、頻出ラベルのうち、未処理ラベルがなくなれば（ステップＳ２０４でｎ）、処理が終了する（ステップＳ２９９）。

未処理の頻出ラベルがあれば（ステップＳ２０４でｙ）、それらのうちから一つを選んで（ステップＳ２０６）、パターンを作成する（ステップＳ２０８）。最初におけるパターンとは、選択したラベルを持つ一つのノードからなるパターンである。

このパターンを指定して、ステップＳ２０８で作成したパターンにノードを付け加えて伸張した伸張パターンの探索の処理（ステップＳ２１０）に入る。

この繰り返しが未処理の頻出ラベルが無くなるまで繰り返される（ステップＳ２０４）。

【００３７】

図３は、伸張パターン探索（ステップＳ２１０）の処理例を示したフローチャートである。このフローチャートによる処理は、パターン探索モジュール１５０が行う処理、又はパターン探索モジュール１５０による指示によって、パターン伸張候補探索モジュール１３０、伸張候補検出判定モジュール１４０が行う処理である。また、この処理は再帰的な処理であり、ステップＳ３２２、ステップＳ３３２で、この処理そのものを呼び出す。

伸張するもとのパターンと、そのパターンに含まれるノードの出現位置などの情報を入力として処理を開始する。

処理が開始されると、入力されたパターンの情報を出力する。この入力パターンの出力処理は、伸張パターンの探索処理の開始直後でもよいし、処理が終了する直前でもよいし、別の時点で行われてもよく、処理の制御と深く関わらないので、図３に示す処理の流れのなかでの時点を示していない。

【００３８】

まず、ステップＳ３０２では、入力のパターンにノードを追加するパスを特定する。

特定したパスにはパターンの一つ以上のノードが存在する。これらのノードの中から、子供のノードを持つノードを列挙し（枝の追加を探す箇所の列挙）、未処理の箇所（ノード）があれば（ステップＳ３０４でｙ）続きの処理（ステップＳ３０６）を、なければ（ステップＳ３０４でｎ）リーフノードの子孫候補を探索する処理（ステップＳ３２４）に

移る。
枝の追加を探す未処理のノードがある場合（ステップＳ３０４でｙ）には、いずれかの未処理のノードを選択する（ステップＳ３０６）。このとき、選択は先祖子孫関係においてより先祖側のものから選択する、あるいはより子孫側のノードから順番に選択することが処理の効率がよくなる。特により先祖側のものから選択する場合が、本実施の形態による効率化の効果をより発揮することができる。
【００３９】
未処理のノード（枝の追加を探すノード）を選択すると、そのノードから追加の枝の候補が見つかる可能性チェックを行う（ステップＳ３０８）。これは、より前に起動された伸張パターンの探索処理により生成され、この伸張パターンの探索処理の開始時に入力された、伸張候補検出判定結果から、処理中の枝の追加を探すノードについての判定結果である発見可能性を参照することで行われる。つまり、伸張候補検出判定モジュール１４０の記憶装置に記憶されている発見可能性を利用する。パターンのノードが一つだけの場合のように事前の伸張候補検出判定結果が利用できない際には、発見可能性ありと判定してもよい。
発見の可能性が無い場合（ステップＳ３０８でｎ）には、この伸張パターンの探索処理から呼び出す伸張パターンの探索所に受け渡すためのデータとして、該当箇所のノードについて伸張ノードの発見が無いことを記録して（ステップＳ３１０）、次の繰り返しに移る（ステップＳ３０４に戻る）。
【００４０】
発見の可能性がある場合（ステップＳ３０８でｙ）には、枝の追加を探索する（ステップＳ３１２）。例えば、そのパターンに対して、新たなノードを追加してパターンを伸張することになるパスを特定する。より具体的には、そのパターンのノードに追加する枝となりえるツリーデータ上のノードを走査して頻出ラベル（頻出ラベルがあればパターン上での追加するノードとなる）を探す処理を行う。
次に、探索結果について、伸張ノード（ツリーデータ上での該当する箇所での頻出ラベルに対応し、パターン上の枝の追加を探しているノードに新たに追加するパターン上のノード）が存在したかしなかったかについて結果を記録する（ステップＳ３１４）。ここで未処理の伸張ノードが見つからなかった場合（ステップＳ３１６でｎ）には、この伸張パターンの探索処理から新たに起動される伸張パターンの探索処理においても同じ箇所に伸張ノードは見つからない。したがって、次の未処理の箇所に対する処理を行う（ステップＳ３０４に戻る）。
【００４１】
次に、未処理の伸張ノードがある間（ステップＳ３１６でｙ）は、ステップＳ３１８からステップＳ３２２による繰り返し処理が行われる。この繰り返し処理では、未処理の伸張ノードがある間、未処理の伸張ノードを選択し（ステップＳ３１８）、入力されたパターン上の枝の追加を探している箇所に、選択した未処理の伸張ノードを追加したパターンを作成し（ステップＳ３２０）、新たに伸張パターンの探索を起動すること（ステップＳ３２２）を繰り返す。そして、未処理の伸張ノードがなくなると（ステップＳ３１６でｎ）、枝の追加を探すために別の箇所の処理に移る（ステップＳ３０４に戻る）。
【００４２】
枝の追加を探す未処理の箇所がなくなると（ステップＳ３０４でｎ）、特定したパス上のリーフノードの子ノードの候補を探す処理（ステップＳ３２４）に移る。なお、「リーフノードの子ノード」の「リーフノード」とは、伸張する過程におけるパターンのリーフノードのことを指しており、そのパターンはリーフノードから伸張することになる。
この処理（ステップＳ３２４）も同様に指定された範囲のノードのラベルの頻出ラベルを探すことで実現することができる。ここでまた、伸張ノードの有無を記録する（ステップＳ３２６）。ここでの未処理の伸張ノードがあるという判定（ステップＳ３２８でｙ）は、単に頻出ラベルがあるということではなく、頻出ラベルが複数種類ある、あるいは、頻出ラベルは１種類だけだが、各ツリーデータにおいて指定された範囲での出現箇所が複

数あるものが指定された数以上ある場合に、未処理の伸張ノードがある（発見可能性がある）という判定となる。つまり、頻出ラベルが１種類でありかつ出現箇所が複数あるツリーが多くない（頻出であるとの判断に用いる閾値に満たない）場合に、未処理の伸張ノードがない（発見可能性なし）と判定する。したがって、頻出ラベルが複数種類ある場合は、発見可能性ありと判定する。頻出ラベルが１種類だが出現箇所が複数あるツリーが一定数以上ある（頻出であるとの判断に用いる閾値以上ある）場合に、発見可能性ありと判定する。頻出ラベルが１種類かつ、出現箇所が複数あるツリーが一定数未満である場合に、発見可能性なしと判定することとなる。

以降、同様に未処理の伸張ノードがある間（ステップＳ３２８でｙ）、未処理の伸張ノードを一つ選択して、同様に新たなパターンを作成して（ステップＳ３３０）、新たに伸張パターンの探索処理を起動すること（ステップＳ３３２）を繰り返す。

【００４３】

前記の伸張ノードの有無の判定は、伸張ノードを探す箇所ごとに行う例を示したが、この判定と判定結果の記録は例えば伸張ノードの種類ごと（頻出ラベルの種類ごと）に行ってもよい。例えば、兄弟ノードにラベルＺがあった場合にその後ろの兄弟には他の兄弟ノードはありえないというような場合に効果を発揮する。

また、頻出ラベルを探すときに参照するツリーデータごとに伸張ノードの有無の判定を行い、ツリーデータごとにその判定結果を保持してもよい。こうすることで、例えばデータ中９０％のツリーデータではラベルＺを持つノードのあとに兄弟ノードが無いというような場合に、９０％のデータについて兄弟ノードを探す処理を省略することができるようになる。

【００４４】

次に例を用いて処理を説明する。図４は、本実施の形態による処理対象の例を示した説明図であり、電子文書、紙媒体等に印刷された文書を含むドキュメントへの操作の履歴データの例である。例えば、図４（ａ）では、あるドキュメントが作成された後、修正が行われ、閲覧が行われる。修正されたドキュメントは別途確認され、印刷されて、承認されて、公式にファイルされる（登録）、一方で承認されたドキュメントはスキャンされており、スキャン結果を誰かが閲覧している。また、これらとは別に、修正されたドキュメントを確認処理と並行して誰かが後から閲覧している、といった記録をツリーデータとして表したものである。

【００４５】

図５は、本実施の形態による処理対象例の記号表記したものを示した説明図である。

図５（ａ）、（ｂ）、（ｃ）は、それぞれ図４（ａ）、（ｂ）、（ｃ）のラベルをＡ、Ｂ、．．．の記号に置き換えたものである。つまり、「作成」を「Ａ」、「閲覧」を「Ｂ」、「修正」を「Ｃ」のようにである。

また、各ノードにはそれぞれを参照するための識別子を記した。例えば、図５（ａ）、（ｂ）、（ｃ）のルートノードにはｖ０を割り当て、ｖ３の子孫であり一番左下に位置するリーフノードには図５（ａ）、（ｂ）、（ｃ）のツリーそれぞれにおいて、ｖ６、ｖ８、ｖ６を割り当てている。

【００４６】

以降、説明を簡単にするために、伸張すべき箇所であるノードが複数あるような場合でも、恣意的に選択して処理を説明する。したがって、複数の伸張すべきノードから選択する順序は、どのような順序であってもよい。パターンとして抽出する最低出現数（つまり、頻出数）を３として説明する。

最初に頻出ラベルを探索すると、ラベルＧとＩは１回のみ出現しているだけであり、他のラベルはそれぞれ３回以上出現しているので、ラベルＧとＩ以外の全てのラベルが頻出となる。このうち、ラベルＡを選択して伸張パターン探索の処理に入る例を、図６を用いて説明する。以降、説明の簡単のために、パターン上のノードについては、ラベルＡを持つノードを単にＡのノード、ノードＡ、あるいは単にＡということもある。

【００４７】

入力パターンがＡをラベルとして持つ一つのノードだけからなるツリー（図６（Ｐａｔ１）参照）であるとき、枝を追加できる箇所は無いため、リーフノードでもあるＡをラベルとして持つノードの子ノードの探索を行う。

この段階でも多くのラベルが頻出となるが、ここでＣを選択する。すなわち、次の入力パターンはＡの子供ノードにＣが加えられたツリーデータとなる（図６（Ｐａｔ２）参照）。

【００４８】

図６（Ｐａｔ２）では、ルートからリーフまで達するパスは一つしかないので、ノードＡからノードＣにいたるパスを選択する。未処理の枝を追加できる箇所、すなわち、既に子ノードを持つノードはノードＡだけであるので、未処理の枝を追加できる場所としてノードＡを選ぶ。ノードＡについて伸張ノードの発見可能性については、判定の方法により否定される構成も実現できるが、ここでの説明ではこの段階で否定されていないものとする。

【００４９】

次に追加できる枝を探索する。Ａのツリーデータ上での出現箇所は、図５（ａ）のｖ０、図５（ｂ）のｖ０、図５（ｃ）のｖ０である。Ｃのツリーデータ上での出現箇所は、図５（ａ）のｖ１、図５（ｂ）のｖ１、図５（ｃ）のｖ１と図５（ｃ）のｖ５である。ここで、Ａの出現箇所の子孫でＣの出現箇所の先祖でも子孫でもないものは、図５（ａ）と図５（ｂ）においては存在しない。図５（ｃ）においては、Ｃの出現箇所をｖ５としたときにｖ７、ｖ８、ｖ９、ｖ１０、ｖ１１が探索する範囲に当たるが、いずれのラベルも図５（ｃ）に示すツリーデータにしか出現しないため頻出となりえない。なお、ｖ２は不要である。これは、兄弟ノードに順序があることを前提として処理しているためである。図５に示す例では、兄弟ノードは左から右に走査する。また、Ｆ、Ｅ、Ｊ、Ｈ、Ｂも不要である。これは、図５（ａ）、（ｂ）では、いずれもこれらのノードがＣの子孫となっており、Ｃの兄弟ノードにはならないためである。

【００５０】

このため、図６（Ｐａｔ２）では、Ｃ以外のラベルを持つノードの枝がＡの子供として新たに現れることはない。この結果を発見可能性無しとして判定結果を記録する。

次に、枝を追加できる未処理の他のノードは無いので、パターン上のリーフノードＣについて子ノードの探索処理に移る。子ノードの出現箇所図５（ａ）のｖ１、図５（ｂ）のｖ１、図５（ｃ）のｖ１と図５（ｃ）のｖ５のそれぞれの子孫ノードの範囲での頻出ラベルを探す。頻出ラベルの中から、ここではラベルＢを選択する。ここでは他にも頻出ラベルとしてＤやＦなども列挙されるので、ノードＣについて、伸張ノードの存在可能性の判定では発見可能性は否定できないことを記録する。

【００５１】

そして、Ａ、Ｃ、Ｂが一直線につながったパターン（図６（Ｐａｔ３）参照）を作成し、伸張パターンの探索を起動する。

次の伸張パターンの起動において、入力パターンである図６（Ｐａｔ３）の中でルートからリーフノードに達するパスは一つしかないので、そのパス（Ａ、Ｃ、Ｂのノードが連なるパス）を特定する。そして、枝を追加するノードを選ぶ。

まず、ノードＡを選択する。頻出の可能性を検査すると、このパターンにおいてＡの子ノードとなる新たな枝は発見されないという情報が呼び出しもとの伸張パターン探索処理により得られているので、頻出ラベルの発見の無を記録して、次の処理に移る。

【００５２】

次にノードＣを選ぶ。発見の可能性は否定されていないので、枝の追加を探索する。Ｃのツリーデータ上での出現箇所は、図５（ａ）のｖ１、図５（ｂ）のｖ１、図５（ｃ）のｖ１と図５（ｃ）のｖ５であり、パターン上でノードＣの子供のノードとなるノードＢの出現箇所はツリーデータ上で、図５（ａ）のｖ２、図５（ａ）のｖ８、図５（ａ）のｖ９、図５（ｂ）のｖ２、図５（ｂ）のｖ８、図５（ｂ）のｖ１０、図５（ｂ）のｖ１１、図５（ｃ）のｖ２、図５（ｃ）のｖ６、図５（ｃ）のｖ１１である。これらの情報から、Ｃ

の出現箇所を先祖にもち、Ｂの出現箇所のいずれかを別のパス（左側のパス）に持つノードは、例えば図５（ａ）ではｖ３、ｖ４、．．．であり、図５（ｂ）ではｖ３、ｖ４、．．．であり、図５（ｃ）ではｖ３、ｖ４、．．．である。この範囲にいくつもの頻出と判定されるラベルがあるが、ここではラベルＤを選択する。

【００５３】

次に、Ａの子ノードにＣ、Ｃの子ノードにＢとＣが順に付いたツリーを入力パターン（図６（Ｐａｔ４）参照）として伸張パターン探索処理を起動する。

このとき、ノードを加えるパスには、リーフノードＤとルートノードＡを結ぶパスが選ばれるものとする。

ノードＡについて追加できる枝を探す処理は、図６（Ｐａｔ３）に対する処理と同様になる。これは、伸張ノードの発見可能性が否定されている情報を、呼び出しもとの伸張パターン探索処理より得ているためである。ただし、ノードＣについて追加できる枝を探す際に発見可能性について検査を行う場合に、判定処理の方式により発見可能性の有無が分かれる。例えば、単に呼び出しもとでの処理で頻出ラベルが一つも見つからなかったか否かで判定している場合には、ここでは発見可能性を否定できない。一方で、Ｄが出現した出現箇所以降にＣの子孫でＤの出現箇所を別のパス（左側のパス）上に持つようなノードで出現する頻出ラベルが存在しないことを判定している場合には、ここでは発見可能性が否定される。しかし、いずれの場合にもこの段階でＣには追加するノードは見つからないことが記録できる。

【００５４】

次にノードＤの子ノードを探索する。以降、同様の処理が施され、Ｆが追加されて図６（Ｐａｔ５）、Ｅが追加されて図６（Ｐａｔ６）、Ｈが追加されて図６（Ｐａｔ７）とパターンが伸張されたものとする。

次の処理では、入力パターンを図６（Ｐａｔ７）を入力として処理が行われる。

ノードを加えるパスは、Ａ、Ｃ、Ｄ、Ｆ、Ｅ、Ｈのノードが連なるパスとなり、枝の追加を探すノードはＡ、Ｃ、Ｄ、Ｆ、Ｅとなる。なお、Ｈがない理由は、Ｈはパターン上でのリーフノードであるためである。これらのうち、それまでの呼び出しもとの伸張パターン探索処理により、Ａ、Ｃ、Ｄ、Ｆについて伸張ノードが発見されないことがわかっている。また、判定の方式により、Ｅについても伸張ノードが発見されないことが呼び出しもとの伸張パターン探索処理で生成されて受け渡された伸張ノードの有無の判定結果を参照することでわかる。これにより、パターン中のノードＡ、Ｃ、Ｄ、Ｆ（場合によってＥも）について枝の追加の探索を省略することができる。そして、ここまでの説明でもパターン中のノード（例えばノードＡ）について、枝の追加の探索が省略できることに何度も触れているように、この省略される回数はパターンが大きくなるにつれて鼠算的に回数を増す。すなわち、本実施の形態の構成を採用しなかった場合に生じていた非常に多くの無駄な処理を、本実施の形態を採用すれば削減することになる。

【００５５】

次に、パターンを抽出する出現回数の最低数を２とした場合について説明する。前述の例と同様に図６（Ｐａｔ１）、図６（Ｐａｔ２）、図６（Ｐａｔ３）、図６（Ｐａｔ４）、図６（Ｐａｔ５）、図６（Ｐａｔ６）とパターンが抽出されたのち、ノードＥの子供ノードとしてノードＪが選ばれて、図６（Ｐａｔ７ｏ）に進んだとする。この段階では、図５（ａ）、（ｂ）、（ｃ）におけるツリーデータで、この図６（Ｐａｔ７ｏ）は出現する。そして、ノードＣは、ノードＤの後にラベルＢが図５（ａ）と（ｂ）のツリーデータの２つで出現するので、Ｃは伸張ノードの発見可能性は否定されていない。なお、Ｃの子供としての兄弟関係で「後」ということになるので、図５（ｃ）のツリーデータではノードＤの「後」にはノードは存在していない。

次に、Ｅの子ノードにノードＨを加えた図６（Ｐａｔ８ｏ）の処理に移った場合を想定する。この図６（Ｐａｔ８ｏ）は、図５（ｂ）のツリーデータでは出現しない。したがって、以降の伸張ノードを探す処理は、図５（ａ）と（ｃ）のツリーデータについて行う。

【００５６】

ここまでの説明と同様に、ノードＡ、Ｄ、Ｆ、Ｅについての伸張ノードの存在可能性は、呼び出しもとでの処理で否定されているため、伸張ノードの探索処理を行わないが、ノードＣについては頻出ノードの発見可能性は否定されていない。このため、図５（ｃ）のツリーデータにおいても、図６（Ｐａｔ８ｏ）のノードＣの出現箇所ｖ１について、ノードＤの出現箇所ｖ３の右のノードの子孫を存在しないにも関わらず捜索する処理を起動することになる。この様な場合にも、ツリーデータごとに頻出ノードの発見可能性のデータが保持されていれば、図５（ｃ）のツリーデータについてはパターン上のノードＣからの伸張ノードをさらに探す処理を省略させることができる。この方法では、多くのツリーデータの形が非常に似通っている中、一部のツリーデータに形が異なっているものが存在するような場合に、本実施の形態の効果を高めることができる。

【００５７】

枝の追加を探すノードの選択は、より上位のノードから実行されることが望ましいが、他の事情によりこれが実現できないこともある。例えば他の事情により枝の追加を探すノードの選択処理を下位のノードから上位のノードの順で行わなければならなかった場合には、それらの処理で判定された伸張ノードの有無のデータをまとめてリーフノードに子ノードを追加して呼び出す伸張パターンの探索処理に送信してもよい。この場合には、本実施の形態による効果の全ては発揮されないが、なお十分な効率化を得ることができる。

【００５８】

なお、本実施の形態としてのプログラムが実行されるコンピュータのハードウェア構成は、図７に例示するように、一般的なコンピュータであり、具体的にはパーソナルコンピュータ、サーバーとなりえるコンピュータ等である。パターン伸張候補探索モジュール１３０、伸張候補検出判定モジュール１４０、パターン探索モジュール１５０等のプログラムを実行するＣＰＵ７０１（この例では演算部としてＣＰＵを用いた）と、そのプログラムやデータを記憶するＲＡＭ７０２と、本コンピュータを起動するためのプログラム等が格納されているＲＯＭ７０３と、補助記憶装置であるＨＤ７０４（例えばハードディスクを用いることができる）と、キーボード、マウス等のデータを入力する入力装置７０６と、ＣＲＴや液晶ディスプレイ等の出力装置７０５と、通信ネットワークと接続するための通信回線インタフェース７０７（例えばネットワークインタフェースカードを用いることができる）、そして、それらをつないでデータのやりとりをするためのバス７０８により構成されている。これらのコンピュータが複数台互いにネットワークによって接続されていてもよい。

【００５９】

前述の実施の形態のうち、コンピュータ・プログラムによるものについては、本ハードウェア構成のシステムにソフトウェアであるコンピュータ・プログラムを読み込ませ、ソフトウェアとハードウェア資源とが協働して、前述の実施の形態が実現される。

なお、図７に示すハードウェア構成は、一つの構成例を示すものであり、本実施の形態は、図７に示す構成に限らず、本実施の形態において説明したモジュールを実行可能な構成であればよい。例えば、一部のモジュールを専用のハードウェア（例えばＡＳＩＣ等）で構成してもよく、一部のモジュールは外部のシステム内にあり通信回線で接続しているような形態でもよく、さらに図７に示すシステムが複数互いに通信回線によって接続されていて互いに協調動作するようにしてもよい。また、特に、パーソナルコンピュータの他、情報家電、複写機、ファックス、スキャナ、プリンタ、複合機（スキャナ、プリンタ、複写機、ファックス等のいずれか２つ以上の機能を有している画像処理装置）などに組み込まれていてもよい。

【００６０】

前記実施の形態においては、対象とするツリーデータとして、ドキュメントの操作履歴のデータを示したが、他の木構造であるデータを対象としてもよい。例えば、順序関係のある商品購入、ドキュメントの構造、遺伝子の配列、遺伝子の二次構造、たんぱく質の構造、画像や映像の特徴情報の構造、人の行動履歴構造、テキストデータの解析結果の構造、オントロジーなどの知識体系、ＸＭＬ等のツリー形式で構造化された情報等であっても

よい。

【００６１】

なお、説明したプログラムについては、記録媒体に格納して提供してもよく、また、そのプログラムを通信手段によって提供してもよい。その場合、例えば、前記説明したプログラムについて、「プログラムを記録したコンピュータ読み取り可能な記録媒体」の発明として捉えてもよい。

「プログラムを記録したコンピュータ読み取り可能な記録媒体」とは、プログラムのインストール、実行、プログラムの流通などのために用いられる、プログラムが記録されたコンピュータで読み取り可能な記録媒体をいう。

なお、記録媒体としては、例えば、デジタル・バーサタイル・ディスク（ＤＶＤ）であって、ＤＶＤフォーラムで策定された規格である「ＤＶＤ－Ｒ、ＤＶＤ－ＲＷ、ＤＶＤ－ＲＡＭ等」、ＤＶＤ＋ＲＷで策定された規格である「ＤＶＤ＋Ｒ、ＤＶＤ＋ＲＷ等」、コンパクトディスク（ＣＤ）であって、読出し専用メモリ（ＣＤ－ＲＯＭ）、ＣＤレコーダブル（ＣＤ－Ｒ）、ＣＤリライタブル（ＣＤ－ＲＷ）等、ブルーレイ・ディスク（Ｂｌｕｅ－ｒａｙ　Ｄｉｓｋ）、光磁気ディスク（ＭＯ）、フレキシブルディスク（ＦＤ）、磁気テープ、ハードディスク、読出し専用メモリ（ＲＯＭ）、電気的消去及び書換可能な読出し専用メモリ（ＥＥＰＲＯＭ）、フラッシュ・メモリ、ランダム・アクセス・メモリ（ＲＡＭ）等が含まれる。

そして、前記のプログラム又はその一部は、前記記録媒体に記録して保存や流通等させてもよい。また、通信によって、例えば、ローカル・エリア・ネットワーク（ＬＡＮ）、メトロポリタン・エリア・ネットワーク（ＭＡＮ）、ワイド・エリア・ネットワーク（ＷＡＮ）、インターネット、イントラネット、エクストラネット等に用いられる有線ネットワーク、あるいは無線通信ネットワーク、さらにこれらの組み合わせ等の伝送媒体を用いて伝送させてもよく、また、搬送波に乗せて搬送させてもよい。

さらに、前記のプログラムは、他のプログラムの一部分であってもよく、あるいは別個のプログラムと共に記録媒体に記録されていてもよい。また、複数の記録媒体に分割して記録されていてもよい。また、圧縮や暗号化など、復元可能であればどのような態様で記録されていてもよい。

【図面の簡単な説明】

【００６２】

【図１】本実施の形態の構成例についての概念的なモジュール構成図である。

【図２】本実施の形態による処理例を示したフローチャートである。

【図３】伸張パターン探索の処理例を示したフローチャートである。

【図４】本実施の形態による処理対象の例を示した説明図である。

【図５】本実施の形態による処理対象例の記号表記したものを示した説明図である。

【図６】本実施の形態による処理例を示した説明図である。

【図７】本実施の形態を実現するコンピュータのハードウェア構成例を示すブロック図である。

【符号の説明】

【００６３】

１１０…データ入力モジュール

１２０…データ保持モジュール

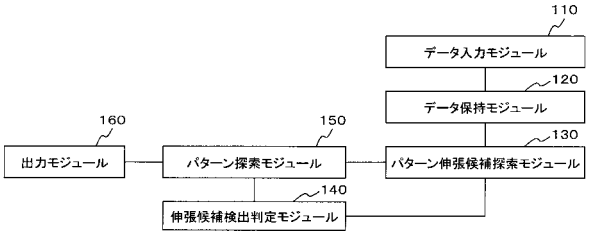
１３０…パターン伸張候補探索モジュール

１４０…伸張候補検出判定モジュール

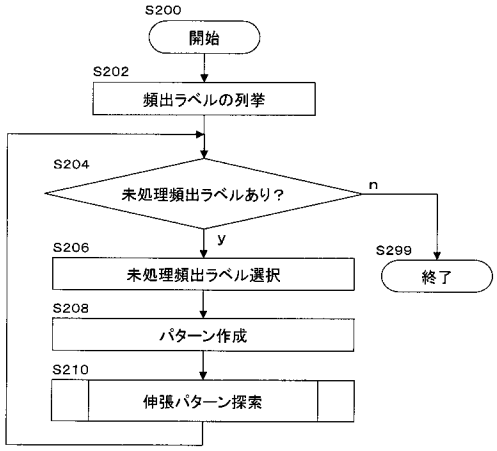
１５０…パターン探索モジュール

１６０…出力モジュール

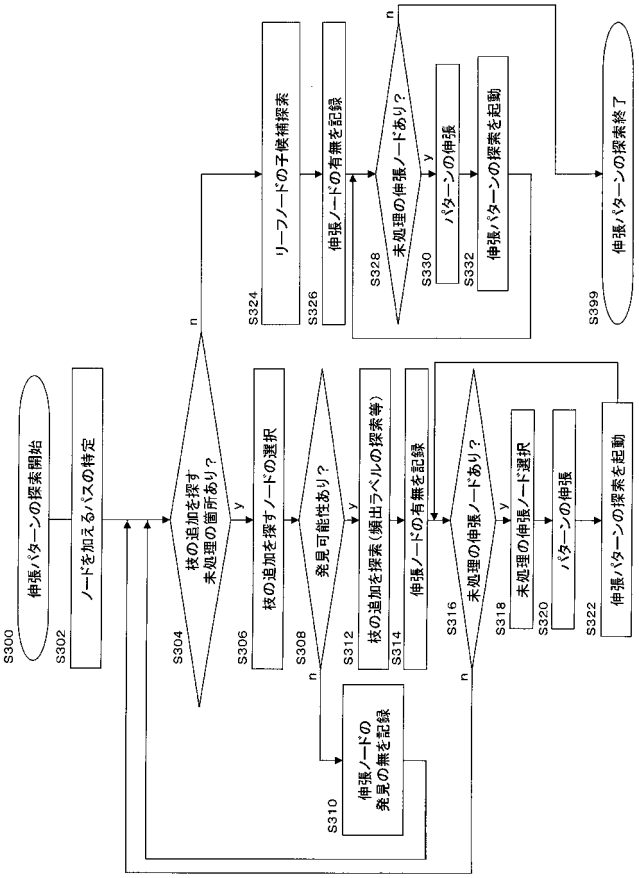
【図１】



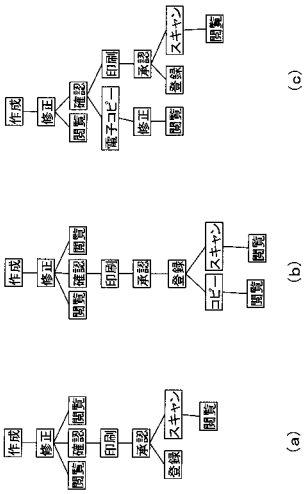
【図２】



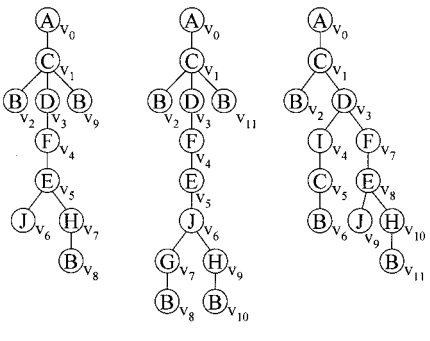
【図３】



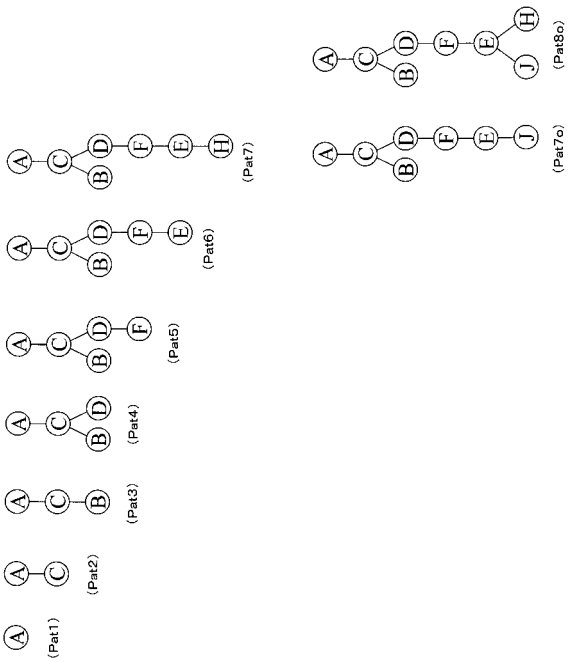
【図４】



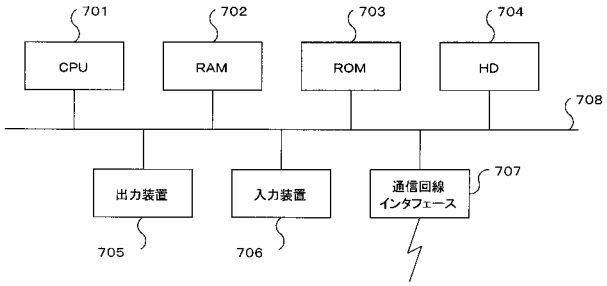
【図５】



【図６】



【図７】


701
CPU
702
RAM
703
ROM
704
HD
708
出力装置
705
入力装置
706
通信回線
インタフェース
707