

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4623446号
(P4623446)

(45) 発行日 平成23年2月2日(2011.2.2)

(24) 登録日 平成22年11月12日(2010.11.12)

(51) Int.Cl.

G06F 17/30 (2006.01)

F 1

G06F 17/30 330B
G06F 17/30 140
G06F 17/30 340D

請求項の数 20 (全 35 頁)

(21) 出願番号 特願2004-199927 (P2004-199927)
 (22) 出願日 平成16年6月8日 (2004.6.8)
 (65) 公開番号 特開2005-353024 (P2005-353024A)
 (43) 公開日 平成17年12月22日 (2005.12.22)
 審査請求日 平成19年6月8日 (2007.6.8)

特許法第30条第1項適用 平成16年2月17日 北
陸先端科学技術大学院大学主催の「修士論文発表会」に
て文書をもって発表

特許権者において、権利譲渡・実施許諾の用意がある。

(73) 特許権者 504259579
田島 敬史
京都府京都市左京区吉田本町 京都大学情
報学研究科内
(72) 発明者 田島 敬史
石川県能美郡辰口町旭台1-1 北陸先端科
学技術大学院大学内
(72) 発明者 福井 佳紀
愛知県西加茂郡三好町筋生鳥居前35番地
審査官 吉田 誠

最終頁に続く

(54) 【発明の名称】データ管理プログラムおよびデータ管理システム

(57) 【特許請求の範囲】

【請求項1】

一つまたは複数の木構造データを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって、

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で複数個、該入力装置に与えられた際に、

与えられた該複数パス式の全てからなる集合の空でない任意の部分集合と、与えられた該複数パス式の全ての接頭辞からなる集合の任意の部分集合との、全ての組合せについて、該複数パス式のうち前者の部分集合に含まれるもの全てと、該接頭辞のうち後者の部分集合に含まれるもの全てとの積と、

該複数パス式のうち前者の部分集合に含まれないもの全てと、該接頭辞のうち後者の部分集合に含まれないもの全てとの和の、

これらの積と和の差から、

さらに、前記与えられた複数パス式の全ての和にマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当するパス式を合成し、

これらの合成パス式の各々について、その解集合となる部分木集合を前記格納装置から取り出すことによって、

前記与えられた複数パス式のうちの一つ以上のパス式の解となる部分木のうち、該複数パ

10

20

式のうちのいずれかのパス式の他の解の一部をなす部分木とならないものの全てを，それがどのパス式の解集合には含まれどのパス式の解集合には含まれないか，および，それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかの，以上の二点に基づいて分類したものにあたる集合群を生成し，これらの集合群を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム。

【請求項 2】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，

10

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で一つ，該入力装置に与えられた際に，

該パス式の全ての接頭辞からなる集合の全ての部分集合について，

前記与えられたパス式と，該接頭辞のうち該部分集合に含まれるもの全てとの積と，

該接頭辞のうち該部分集合に含まれないもの全てとの和の，

これらの積と和の差から，

さらに，前記与えられたパス式にマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当するパス式を合成し，

これらの合成パス式の各々について，その解集合となる部分木集合を前記格納装置から取り出すことによって，

20

前記与えられたパス式の解となる部分木のうち，該パス式の他の解の一部をなす部分木とならないものの全てを，それが該パス式のどの接頭辞にはマッチし該パス式のどの接頭辞にはマッチしないかに基づいて分類したものにあたる集合群を生成し，

これらの集合群を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム。

【請求項 3】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，

30

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で複数個，該入力装置に与えられた際に，

該複数パス式を用いて該複数パス式の各々に対する解集合となる部分木集合を該格納装置から取り出した後に，

前記与えられた複数パス式のうちの一つ以上のパス式の解となる部分木のうち，該複数パス式のうちのいずれかのパス式の他の解の一部をなす部分木としても出現するものを取り除き，さらに，残った部分木の全てについて，それがどのパス式の解集合には含まれどのパス式の解集合には含まれないか，および，それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかを調べて，以上の二点の結果に基づいて分類することによって，これらの各分類にあたる集合群を生成し，

40

これらの集合群を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム。

【請求項 4】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で一つ，該入力装置に与えられた際に，

該パス式を用いてその解集合となる部分木集合を該格納装置から取り出した後に，

該パス式の解となる部分木のうち，該パス式の他の解の一部をなす部分木としても出現す

50

るものを取り除き，さらに，残った部分木の全てについて，それが該パス式のどの接頭辞にはマッチし該パス式のどの接頭辞にはマッチしないかを調べて，その結果に基づいて分類することによって，これらの各分類にあたる集合群を生成し，これらの集合群を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム。

【請求項5】

データの集合を受け取る入力装置に接続された計算機を，動作させるためのデータ管理プログラムであって，

請求項1または3に記載のデータ管理プログラムによって出力された前記各分類を表す部分木集合群を該入力装置で受け取り，該データ管理プログラムに当初与えられた前記複数パス式の各々に対する解集合を，該パス式の解集合に含まれるとされていた分類の全て，および，該パス式の各接頭辞にマッチするとされていた各分類中の部分木から，該パス式中で該接頭辞と対になる接尾辞を用いて取り出した部分木の集合の，これら全ての和集合を取ることによって合成するように計算機を動作させることを特徴とするデータ管理プログラム。

10

【請求項6】

データの集合を受け取る入力装置に接続された計算機を，動作させるためのデータ管理プログラムであって，

請求項2または4に記載のデータ管理プログラムによって出力された前記各分類を表す部分木集合群を該入力装置で受け取り，該データ管理プログラムに当初与えられた前記パス式の解集合を，該各分類の全て，および，該パス式の各接頭辞にマッチするとされていた各分類中の部分木から，該パス式中で該接頭辞と対になる接尾辞を用いて取り出した部分木の集合の，これら全ての和集合を取ることによって合成するように計算機を動作させることを特徴とするデータ管理プログラム。

20

【請求項7】

複数のデータを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，該格納装置に記憶されているデータの部分集合を取り出す検索式が複数個，該入力装置に与えられた際に，

該複数検索式全てからなる集合の全ての空でない部分集合について，該複数検索式のうち該部分集合に含まれるもの全ての積と，該複数検索式のうち該部分集合に含まれないもの全ての和の，これらの積と和の差に相当する検索式を合成し，これらの合成検索式の各々について，その解集合となるデータ集合を前記格納装置から取り出すことによって，

30

前記与えられた複数検索式のうちの一つ以上の検索式の解となるデータの全てを，それがどの検索式の解集合には含まれどの検索式の解集合には含まれないかに基づいて分類したものにあたる集合群を生成し，

これらの集合群を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム。

【請求項8】

データの集合を受け取る入力装置に接続された計算機を，動作させるためのデータ管理プログラムであって，

請求項7に記載のデータ管理プログラムによって出力された前期各分類を表す集合群を該入力装置で受け取り，該データ管理プログラムに当初与えられた前記複数検索式の各々に対する解集合を，該検索式の解集合に含まれるとされていた分類全ての和集合を取ることで合成するように計算機を動作させることを特徴とするデータ管理プログラム。

40

【請求項9】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，

50

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式が複数個，該入力装置に与えられた際に，

該複数検索式を用いて該複数検索式の各々に対する解集合となる部分木集合を該格納装置から取り出した後に，

該複数検索式のうちのある検索式の解となる部分木の一つが，該複数検索式のうちのいずれかの検索式の他の解の一部をなす部分木としても出現する場合には，他の解の一部としても出現する該部分木を解集合から取り除いた上で，各解集合を前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム．

【請求項 10】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置とに接続された計算機を動作させるためのデータ管理プログラムであって，

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式が複数個，該入力装置に与えられた際に，

与えられた該複数検索式の全てに適当な全順序を定め，

該複数検索式の各々について，

該検索式から，該検索式より該全順序で若い順番を持つ全ての検索式にマッチするデータを取り除いた差から，

さらに，前記与えられた複数検索式のいずれかにマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当する検索式を合成し，

これらの合成検索式の各々について，その解集合となる部分木集合を前記格納装置から取り出すことによって，

前記与えられた複数検索式のうちの一つ以上の検索式の解となる部分木のうち，該複数検索式のうちのいずれかの検索式の他の解の一部をなす部分木とならないものの全てを，重複なく取り出し，これらを前記出力装置に出力するように計算機を動作させることを特徴とするデータ管理プログラム．

【請求項 11】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置と，これらに接続された計算機とからなるデータ管理システムであって，

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で複数個，該入力装置に与えられた際に，

与えられた該複数パス式の全てからなる集合の空でない任意の部分集合と，与えられた該複数パス式の全ての接頭辞からなる集合の任意の部分集合との，全ての組合せについて，該複数パス式のうち前者の部分集合に含まれるもの全てと，該接頭辞のうち後者の部分集合に含まれるもの全てとの積と，

該複数パス式のうち前者の部分集合に含まれないもの全てと，該接頭辞のうち後者の部分集合に含まれないもの全てとの和の，

これらの積と和の差から，

さらに，前記与えられた複数パス式の全ての和にマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当するパス式を合成し，

これらの合成パス式の各々について，その解集合となる部分木集合を前記格納装置から取り出すことによって，

前記与えられた複数パス式のうちの一つ以上のパス式の解となる部分木のうち，該複数パス式のうちのいずれかのパス式の他の解の一部をなす部分木とならないものの全てを，それがどのパス式の解集合には含まれどのパス式の解集合には含まれないか，および，それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかの，以上の二点に基づいて分類したものにあたる集合群を生成し，

10

20

30

40

50

これらの集合群を前記出力装置に出力するように計算機が動作することを特徴とするデータ管理システム。

【請求項 1 2】

一つまたは複数の木構造データを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置と、これらに接続された計算機とからなるデータ管理システムであって、

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で一つ、該入力装置に与えられた際に、

該パス式の全ての接頭辞からなる集合の全ての部分集合について、

前記与えられたパス式と、該接頭辞のうち該部分集合に含まれるもの全てとの積と、

該接頭辞のうち該部分集合に含まれないもの全てとの和の、

これらの積と和の差から、

さらに、前記与えられたパス式にマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当するパス式を合成し、

これらの合成パス式の各々について、その解集合となる部分木集合を前記格納装置から取り出すことによって、

前記与えられたパス式の解となる部分木のうち、該パス式の他の解の一部をなす部分木とならないものの全てを、それが該パス式のどの接頭辞にはマッチし該パス式のどの接頭辞にはマッチしないかに基づいて分類したのにあたる集合群を生成し、

これらの集合群を前記出力装置に出力するように計算機が動作することを特徴とするデータ管理システム。

【請求項 1 3】

一つまたは複数の木構造データを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置と、これらに接続された計算機とからなるデータ管理システムであって、

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で複数個、該入力装置に与えられた際に、

該複数パス式を用いて該複数パス式の各々に対する解集合となる部分木集合を該格納装置から取り出した後に、

前記与えられた複数パス式のうちの一つ以上のパス式の解となる部分木のうち、該複数パス式のうちのいずれかのパス式の他の解の一部をなす部分木としても出現するものを取り除き、さらに、残った部分木の全てについて、それがどのパス式の解集合には含まれどのパス式の解集合には含まれないか、および、それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかを調べて、以上の二点の結果に基づいて分類することによって、これらの各分類にあたる集合群を生成し、

これらの集合群を前記出力装置に出力するように計算機が動作することを特徴とするデータ管理システム。

【請求項 1 4】

一つまたは複数の木構造データを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置と、これらに接続された計算機とからなるデータ管理システムであって、

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式がパス式の形で一つ、該入力装置に与えられた際に、

該パス式を用いてその解集合となる部分木集合を該格納装置から取り出した後に、

該パス式の解となる部分木のうち、該パス式の他の解の一部をなす部分木としても出現するものを取り除き、さらに、残った部分木の全てについて、それが該パス式のどの接頭辞にはマッチし該パス式のどの接頭辞にはマッチしないかを調べて、その結果に基づいて分類することによって、これらの各分類にあたる集合群を生成し、

これらの集合群を前記出力装置に出力するように計算機が動作することを特徴とするデータ

10

20

30

40

50

タ管理システム。

【請求項 1 5】

データの集合を受け取る入力装置と、これに接続された計算機とからなるデータ管理システムであって、

請求項 1 1 または 1 3 に記載のデータ管理システムによって出力された前記各分類を表す部分木集合群を該入力装置で受け取り、該データ管理システムに当初与えられた前記複数パス式の各々に対する解集合を、該パス式の解集合に含まれるとされていた分類の全て、および、該パス式の各接頭辞にマッチするとされていた各分類中の部分木から、該パス式中で該接頭辞と対になる接尾辞を用いて取り出した部分木の集合の、これら全ての和集合を取ることによって合成するように計算機が動作することを特徴とするデータ管理システム。10

【請求項 1 6】

データの集合を受け取る入力装置と、これに接続された計算機とからなるデータ管理システムであって、

請求項 1 2 または 1 4 に記載のデータ管理システムによって出力された前記各分類を表す部分木集合群を該入力装置で受け取り、該データ管理システムに当初与えられた前記パス式の解集合を、該各分類の全て、および、該パス式の各接頭辞にマッチするとされていた各分類中の部分木から、該パス式中で該接頭辞と対になる接尾辞を用いて取り出した部分木の集合の、これら全ての和集合を取ることによって合成するように計算機が動作することを特徴とするデータ管理システム。20

【請求項 1 7】

複数のデータを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置と、これらに接続された計算機とからなるデータ管理システムであって、該格納装置に記憶されているデータの部分集合を取り出す検索式が複数個、該入力装置に与えられた際に、

該複数検索式全てからなる集合の全ての空でない部分集合について、該複数検索式のうち該部分集合に含まれるもの全ての積と、該複数検索式のうち該部分集合に含まれないもの全ての和の、これらの積と和の差に相当する検索式を合成し、これらの合成検索式の各々について、その解集合となるデータ集合を前記格納装置から取り出すことによって、30

前記与えられた複数検索式のうちの一つ以上の検索式の解となるデータの全てを、それがどの検索式の解集合には含まれどの検索式の解集合には含まれないかに基づいて分類したものにあたる集合群を生成し、

これらの集合群を前記出力装置に出力するように計算機が動作することを特徴とするデータ管理システム。

【請求項 1 8】

データの集合を受け取る入力装置と、これに接続された計算機とからなるデータ管理システムであって、

請求項 1 7 に記載のデータ管理システムによって出力された前期各分類を表す集合群を該入力装置で受け取り、該データ管理システムに当初与えられた前記複数検索式の各々に対する解集合を、該検索式の解集合に含まれるとされていた分類全ての和集合を取ることで合成するように計算機が動作することを特徴とするデータ管理システム。40

【請求項 1 9】

一つまたは複数の木構造データを記憶する格納装置と、データ検索式を受け取る入力装置と、データを出力する出力装置と、これらに接続された計算機とからなるデータ管理システムであって、

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式が複数個、該入力装置に与えられた際に、

該複数検索式を用いて該複数検索式の各々に対する解集合となる部分木集合を該格納装置から取り出した後に、50

該複数検索式のうちのある検索式の解となる部分木の一つが，該複数検索式のうちのいずれかの検索式の他の解の一部をなす部分木としても出現する場合には，他の解の一部としても出現する該部分木を解集合から取り除いた上で，各解集合を前記出力装置に出力するよう計算機が動作することを特徴とするデータ管理システム．

【請求項 20】

一つまたは複数の木構造データを記憶する格納装置と，データ検索式を受け取る入力装置と，データを出力する出力装置と，これらに接続された計算機とからなるデータ管理システムであって，

該格納装置に記憶されている木構造データ中の様々な深さの階層にある部分木の集合を取り出す検索式が複数個，該入力装置に与えられた際に，

10

与えられた該複数検索式の全てに適当な全順序を定め，

該複数検索式の各々について，

該検索式から，該検索式より該全順序で若い順番を持つ全ての検索式にマッチするデータを取り除いた差から，

さらに，前記与えられた複数検索式のいずれかにマッチするデータの全ての子孫（自身を含まない）を取り除いた差

に相当する検索式を合成し，

これらの合成検索式の各々について，その解集合となる部分木集合を前記格納装置から取り出すことによって，

前記与えられた複数検索式のうちの一つ以上の検索式の解となる部分木のうち，該複数検索式のうちのいずれかの検索式の他の解の一部をなす部分木とならないものの全てを，重複なく取り出し，これらを前記出力装置に出力するよう計算機が動作することを特徴とするデータ管理システム．

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は，計算機に格納されているデータの一部をユーザからの要求に応じて取り出し，あるいはさらに，取り出したデータのネットワーク等を介しての送信や，ユーザ等への呈示や，記憶媒体等への保存等を行うシステムにおいて，データ中の冗長性を除去した状態で，取り出し，送信，呈示，保存等を行うことによって，取り出し処理，送信処理，保存領域等のコストの軽減ならびに，ユーザの閲覧利便性の向上を実現するためのプログラムおよびシステムに関する．

30

【背景技術】

【0002】

ユーザからのデータの取り出し要求に応じて，管理しているデータの中から，その取り出し要求への解となるデータの集合，すなわち解集合を取り出すデータ管理システムで，解集合中の重複を取り除くことを考慮するシステムに関しては，以下の四つの分類軸が考えられる．

【0003】

（1）同一のデータが複数回，解として出現するような重複のみを考慮するか，あるいは対象データが木構造として表現できるような入れ子構造を持つ場合には，ある解が別の解の部分木として出現するような重複も考慮するか．

40

【0004】

（2）取り出したデータを調べて重複を発見し取り除くか，あるいは，与えられたデータ取り出し要求を解析し，重複を含まない形でデータを取り出すようにこれらの取り出し要求を変形したものを合成し，これらを用いてデータを取り出すことで，重複を取り除くか．

【0005】

（3）一つのデータ取り出し要求に対する解集合内の重複のみを考慮するか，あるいは，複数のデータ取り出し要求を同時に受け取った場合は，これらのうちの異なる要求に対する解の間での重複についても考慮するか．

50

【0006】

(4) 重複を取り除いた状態のデータから、必要に応じて、当初の要求通りの、重複を含む解集合を合成する機能を有するかどうか。

【0007】

これら四つの分類軸の組合せから16通りの場合が考えられる。それらのうち、分類軸の(1)に関して、同一のデータが複数回、解となるような重複のみを考慮するものに分類される8通りのうちのいくつかについては、先行技術が存在する。

【0008】

まず、取り出したデータを調べて重複を発見する手法で、一つの取り出し要求のみを考慮するシステムについては、下記特許文献1や下記特許文献2において、ユーザからの検索要求に応じて、検索結果を表示するシステムで、検索結果の表示前にその内容を調べ、同じデータが複数回含まれる場合には、このうちの一つを残して削除するシステムが開示されている。同様に、取り出したデータを調べて重複を発見する手法で、複数の問い合わせを考慮する点についても、下記特許文献1において言及されている。

【特許文献1】特開平07-295994

【特許文献2】特開平10-143523

【0009】

また、データ取り出し要求を変形して重複を取り除く手法で、一つの要求のみを考慮する技術の例としては、関係データベースに対して「従業員の中で60歳以上の者の社員番号の集合と、従業員の中で勤続年数が40年以上の者の社員番号の集合の集合和を取り出せ」という要求を表すSQL検索式：

```
(select 社員番号 from 従業員 where 年齢 >= 60)
union
(select 社員番号 from 従業員 where 勤続年数 >= 40)
が与えられた時、これを「従業員の中で年齢が60歳以上か、もしくは、勤続年数が40年以上の者の社員番号の集合を取り出せ」という要求を表すSQL検索式：
```

```
select 社員番号 from 従業員
      where 年齢 >= 60 or 勤続年数 >= 40
```

へと変形することで、同じ従業員の取り出し処理を二回行う可能性を排除するといった手法が広く知られている。

【0010】

また、データ取り出し要求を変形して重複を取り除く手法で、複数の要求を考慮する技術の例としては、*reminder query*と呼ばれる手法が下記非特許文献1に開示されている。これは、以前に「年齢が60歳以上の従業員の社員番号の集合を取り出せ」という要求があった場合、この要求内容とそれに対する結果のデータを保存しておき、その後に「年齢が50歳以上の従業員の社員番号の集合を取り出せ」という要求があった場合には、これを、以前の要求の解集合との差分のみを取り出す「年齢が50歳以上かつ60歳未満の従業員の社員番号の集合を取り出せ」という要求に変形してデータを取り出し、その結果と、保存しておいた60以上の従業員の社員番号の集合との集合和を取って、年齢が50歳以上の従業員の社員番号の集合を生成する手法である。これにより、以前に取り出したことがある、60以上の従業員の情報を繰り返し取り出す処理を排除することができる。この例からわかるように、*reminder query*は(4)の分類軸で述べた、冗長性を取り除いて取り出したデータから、当初の要求通りの解となるデータを合成する機能についても、これを有していることがわかる。

【非特許文献1】S. Dar他著 "Semantic Data Caching and Replacement" International Conference on Very Large Databases (VLDB) 予稿集, 頁330-341, 1996

【0011】

同様に、木構造を持つデータであるXMLデータの管理においても、*reminder*

10

20

30

40

50

queryの手法を用いる研究が下記非特許文献2で示されている。しかし、非特許文献2で示されている手法では、扱うデータは木構造データであっても、過去の要求の解集合中のある要素が新しい要求の解ともなる場合のみを考慮していて、新しい要求のある解が過去の要求のある解の部分木になっているような場合については考慮しておらず、分類軸の(1)に関しては、通常のreminder queryの手法と同様に分類される。

【非特許文献2】L. Chen, E. A. Rundensteiner著 “ACE-XQ: A Cache-aware XQuery answeringsystem” International Workshop on the Web and Databases (WebDB) 予稿集, 頁31-36, 2002

【0012】

10

また、木構造データが別の木構造データの部分木として出現する場合の冗長性とは異なるが、ある要求に対する解集合中の異なる要素の間で、その値の一部分のみが共通である場合に、その部分の冗長性を取り除くための手法が下記特許文献3に開示されている。

【特許文献3】特開平07-121571

この手法では、取り出したデータが例えば

佐藤, 総務部

田中, 総務部

竹田, 人事部

石井, 人事部

20

のような表形式のデータであった場合、これを「同じ値」を表わす符合「」を用いて

佐藤, 総務部

田中, "

竹田, 人事部

石井, "

のような表現へと変形することで、冗長性を取り除く。

【発明の開示】

【発明が解決しようとする課題】

【0013】

しかしながら、上で分類軸(1)として挙げた点に関して、木構造として表現できるデータを管理する際に生じる、ある解が別の解の一部をなす部分木としても出現するような種類の重複についても考慮するような技術については、これまでに開示されていない。そのため、このような形での重複を含むデータを、取出し、送信、呈示、保存等を行うことによって、不必要に処理コストや必要記憶容量を増大させたり、ユーザの閲覧利便性を低下させるという問題がある。

30

【0014】

また、同一のデータが解として複数回出現するような重複のみを考えた場合に關しても、前述のreminder queryの手法では、重複を取り除いた状態でデータを取り出してから、本来の解集合を合成するということが、常に可能とは限らない。例えば、以前に「40歳以上60歳未満の従業員の社員番号の集合を取り出せ」という要求があり、その結果を保存してある時に、新たな要求として「30歳以上50歳未満の従業員の社員番号の集合を取り出せ」という要求が与えられたとする。この場合、40歳以上50歳未満の従業員の社員番号については、全て保存しているデータの中に既に含まれてあり、これを再度取り出すのは冗長である。しかし、保存している40歳以上60歳未満の従業員の「社員番号」のデータを見ただけでは、どの社員番号が50歳未満の社員の番号であるのか判定不可能であり、そのため、再び「30歳以上50歳未満の従業員の社員番号の集合を取り出せ」という要求でデータを取り出すしか方法がない。

40

【0015】

これは、reminder queryの手法が通常想定しているような、最初に「40歳以上60歳未満」という条件の要求が与えられた時点では、その後に「30歳以上50

50

「歳未満」という条件の要求が与えられることは予測不可能である状況では避けられない問題であるが、これら「40歳以上60歳未満」という条件の要求と「30歳以上50歳未満」という条件の要求の二つが同時に与えられるような場合には、これらの要求を「30歳以上40歳未満」と「40歳以上50歳未満」と「50歳以上60歳未満」の三つの要求に分割して実行すれば、重複を取り除いた形でデータを取得し、その後にこれらのデータを使って、当初の要求である「40歳以上60歳未満」の社員番号の集合と「30歳以上50歳未満」の社員番号の集合を生成することは容易である。よって、この例のように複数の取出し要求が「同時に」与えられる場合は、単純に `reminder query` の手法を適用するのでは、重複するデータの取出しを行わずに、かつ後から当初の解集合を合成する方法が存在するにも関わらず、不必要に重複するデータの取出しを行って処理コストを増大させてしまう場合があるという問題がある。10

【0016】

本発明の目的は、以上のような問題点に鑑み、一つまたは複数のデータ取出し要求が同時に与えられた場合にはいかなる場合においても、かつ、ある解が木構造データを持つ他の解の一部をなす部分木として出現するような種類の重複まで含めて、重複を取り除き、かつ、この重複を取り除いたデータから、必要に応じて当初の取出し要求の解集合を合成する機能を持つデータ管理プログラムおよびシステムを提供することによって、データの取出し、送信の処理コストの低減や、データの保存の記憶容量の節減、データをユーザへ呈示する際の閲覧利便性の向上を達成することにある。20

【課題を解決するための手段】

【0017】

本発明のデータ管理プログラムおよびシステムは、データの取出し要求の集合が与えられた際には、いずれかの取出し要求の解となるデータ全てを、それがどの取出し要求に対する解集合には含まれどの取出し要求に対する解集合には含まれないかに基づいて分類したものを生成する。これにより、同じデータが複数の取出し要求の解集合に重複して現れるという冗長性を除去することができると同時に、後述のように、これらの分類から当初の取出し要求の解集合を合成することができる。20

【0018】

本発明のデータ管理プログラムおよびシステムは、上記分類を、与えられた取出し要求を用いて各解集合を取り出した後に生成するか、または、与えられた取出し要求をもとに合成した取出し要求を用いて、直接、各分類を取り出す。後者の場合においては、各分類の取り出しを、その分類を解集合に含むとされる取出し要求の全ての積と、その分類を解集合に含まないとされる取出し要求の全ての和の、これら二つの差に相当する取出し要求を用いて行う。30

【0019】

そして、前述のように、本発明のデータ管理プログラムおよびシステムは、上述の各分類から、必要に応じて、当初の取出し要求の解集合を合成することができる。この合成は、該取出し要求の解集合に含まれるとされていた分類全ての和集合を取ることで行う。40

【0020】

また、本発明のデータ管理プログラムおよびシステムは、木構造データ中の様々な深さの階層にある部分木の集合の取出し要求が与えられて、該要求が取り出す部分木の一つが、該要求が取り出す別の部分木の中にその一部をなす部分木としても重複して出現する場合には、この前者の部分木を該要求に対する解集合から取り除く。ここで、ある要素がある要素の部分木であるとは、二つの要素が全く同じ木である場合も含む物とする。これにより、該取出し要求の解集合の中に存在する、木構造データ特有の部分木としての重複による冗長性を除去することができる。50

【0021】

同様に、本発明のデータ管理プログラムおよびシステムは、木構造データ中の様々な深さの階層にある部分木の集合の取出し要求が、複数個与えられて、該複数要求中のある要求が取り出す部分木の一つが、該複数要求中の別の要求が取り出すある部分木の中にその一

部をなす部分木としても重複して出現する場合には、この前者の部分木を対応する要求に対する解集合から取り除く。ここでも、ある要素がある要素の部分木であるとは、二つの要素が全く同じ木である場合も含む物とする。これにより、該複数取出し要求の解集合の間に存在する、木構造データ特有の部分木としての重複による冗長性を除去することができる。

【0022】

本発明のデータ管理プログラムおよびシステムは、上述の重複する部分木の除去を、与えられた取出し要求を用いて部分木集合を取り出した後に行うか、あるいは、与えられた取出し要求をもとに合成した取出し要求を用いて、重複が取り除かれた部分木集合を直接取り出すことで行う。

10

【0023】

そして、本発明のデータ管理プログラムおよびシステムは、上述のようにして、重複を取り除かれた部分木集合から、当初の取出し要求の解となる部分木集合を合成する機能を有する。

【0024】

また、本発明のデータ管理プログラムおよびシステムは、木構造データ中の様々な深さの階層にある部分木の集合の取出し要求がパス式の形で一つ与えられた際には、該パス式の解となる部分木のうち、該パス式の他の解の一部をなす部分木とならない物の全てを、それが該パス式のどの接頭辞にはマッチし該パス式のどの接頭辞にはマッチしないかに基づいて分類したものを生成する。これにより、該取出し要求の解集合中に存在する、木構造データ特有の部分木としての重複による冗長性を除去することができると同時に、後述のように、これらの分類から当初の取出し要求の解集合を合成することができる。ここで、パス式とは、木構造データから取り出したいノードの指定を、木構造データの根ノードを始点、取り出したいノードを終点とするパス上の各ノードが満たすべき条件を記述することで行うもので、通常、各ノードに関する条件を根ノードに近い側のノードの条件から順に並べた形で記述される。

20

【0025】

同様に、本発明のデータ管理プログラムおよびシステムは、木構造データ中の様々な深さの階層にある部分木の集合の取出し要求がパス式の形で複数個与えられた際には、いずれかのパス式の解となる部分木のうち、いずれかのパス式の他の解の一部をなす部分木とならない物の全てを、それがどのパス式の解集合には含まれどのパス式の解集合には含まれないか、および、それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかの、以上の二点に基づいて分類したものを生成する。これにより、該複数取出し要求の解集合の間に存在する、木構造データ特有の部分木としての重複による冗長性を除去することができると同時に、後述のように、これらの分類から当初の取出し要求の解集合を合成することができる。

30

【0026】

本発明のデータ管理プログラムおよびシステムは、上記の分類を、与えられたパス式を用いて全データを取り出した後に生成するか、または、与えられたパス式をもとに合成したパス式を用いて、各分類を取り出す。後者の場合においては、パス式が一つ与えられた場合には、各分類を取り出すパス式の合成を、与えられたパス式およびその分類とマッチするにとされていた接頭辞の全ての積と、その分類とマッチしないとされていた接頭辞の全ての和の、これら二つの差に相当するパス式を用いて行う。また、後者の場合において、パス式が複数個与えられた場合には、各分類を取り出すパス式の合成を、その分類を解集合に含むとされていた取出し要求およびその分類とマッチするとされていた接頭辞の全ての積と、その分類を解集合に含まないとされていた取出し要求およびその分類とマッチしないとされていた接頭辞の全ての和の、これら二つの差に相当するパス式を用いて行う。

40

【0027】

また、前述のように、本発明のデータ管理方法およびシステムは、上述の各分類から、必要に応じて当初のパス式の解集合を合成することができる。この合成は、該パス式の解集

50

合に含まれるとされていた分類の全てと，および，該パス式の各接頭辞にマッチするとされていた各分類中の部分木から，該パス式中で該接頭辞と対になる接尾辞を用いて取り出した部分木の集合との，これら全ての和集合を取ることによって行う。

【発明の効果】

【0028】

解の一部が重複する複数の検索式が同時に与えられた場合には，上述のような方法でデータを取り出すことによって，従来の `reminder query` の手法では対応できないような場合においても，重複を含まない形でデータを取り出すことが可能となり，従来の方法やシステムに比べて，データの取出しや送信の処理コストの軽減，データの保存のための記憶容量の節減，データを表示されたユーザの閲覧利便性向上を実現することができる。10

【0029】

また，木構造データを管理する場合に生じる，ある解が，他の解の一部をなす部分木としても出現するという種類の重複についても，これを取り除いた状態でデータの取出し，送信，表示，保存を行うことができ，従来の方法やシステムに比べて，木構造データを扱う場合の，データの取出しや送信の処理コストの軽減，データの保存のための記憶容量の節減，データを表示されたユーザの閲覧利便性向上を実現することができる。

【0030】

また，本発明のデータ管理方法またはシステムは，重複を取り除いた状態のデータから，当初の要求通りのデータを合成する機能を有するため，例えば，重複を取り除いた状態でネットワーク等を介してデータを送信し，これを受信した側で，当初の取出し要求で要求されていたデータを合成することによって，ユーザに表示する内容を当初の要求通りの物に維持しつつ，ネットワークの送信コストを節減することができる。同様に，重複を取り除いた状態でデータを保存し，後に必要になった時点で，当初の取出し要求で要求されていたデータを合成することによって，ユーザに表示する内容を当初の要求通りの物に維持しつつ，データの保存のための記憶容量を節減することができる。20

【発明を実施するための最良の形態】

【0031】

(第一の実施の形態)

次に，本発明の第一の実施の形態の詳細について，詳しく説明する。本説明においては，本発明を，JIS 規格である SQL 検索式によってデータの取出し要求を行う関係データベースに適用する場合の一実施形態について説明する。本実施形態においては，与えられた複数検索式の解集合が互いに重複する要素を含みうる。よって，システムは，与えられた複数検索式のうちのいずれかの検索式の解集合に含まれるような全データを，そのデータがどの検索式の解には含まれどの検索式の解には含まれないかに基づいて分類したものを作成する。これにより，同じデータが複数の取出し要求の解集合に重複して現れるという冗長性を除去することができ，さらに，後述のように，これらの分類から，必要に応じて，当初の検索式の解を合成することができる。一方，扱うデータは木構造データではないので，解集合中のある要素が，同じまたは別の解集合中の別の要素の部分木として重複して出現するような種類の重複は，当実施形態においては考慮する必要はない。30

【0032】

図 2 に，本実施形態におけるシステムの全体構成図を示す。本実施形態におけるシステムは，関係データを格納する格納装置 4 と，ユーザから SQL で記述された検索式を受け取る入力装置 1 と，入力装置 1 が受け取った一つまたは複数の検索式を受け取り，各検索式について，格納装置 4 中の関係データから，その検索式の解となるデータの集合を取り出す検索装置 3 と，検索装置 3 が取り出したデータを受け取って，これらのデータから，上述の各分類を生成する重複除去装置 10 と，重複除去装置が生成したデータを受け取って出力する出力装置 6 からなる。

【0033】

以下，本実施形態において，入力装置が検索式の集合を受け取った際の処理手順について40

10

20

30

40

50

説明する。例として、入力装置に三つの検索式が与えられた場合を考える。この場合、これらの検索式の解集合は互いに重複する要素を含みうる。一般には、三つの検索式が与えられた場合、それらの解集合を A_1, A_2, A_3 とすると、それら三つの集合の関係は、図 3 に示すベン図のようになる。そこで、本システムでは、重複除去装置は A_1, A_2, A_3 を受け取ると、これらに含まれるデータを、図 3 に示すベン図中の 7 つの領域に対応する 7 つのグループに分類したもの、すなわち、以下の式で定義される 7 つの集合を生成する。

$$\begin{array}{lll}
 A_1 & A_2 & A_3 \\
 (A_1 \cap A_2) & - & A_3 \\
 (A_1 \cap A_3) & - & A_2 \\
 (A_2 \cap A_3) & - & A_1 \\
 A_1 & - & (A_2 \cup A_3) \\
 A_2 & - & (A_1 \cup A_3) \\
 A_3 & - & (A_1 \cup A_2)
 \end{array}$$

10

ただし、ここで、 \cap は集合の積演算、 \cup は集合の和演算、 $-$ は集合の差演算を表わすものとする。これら 7 つのグループが互いに重複する要素を含まないことは、図 3 に示すベン図より明らかである。

【0034】

より一般には、 n 個の検索式を受け取った場合、それらの解集合を A_1, \dots, A_n とするとき、 $S = \{A_1, \dots, A_n\}$ の空でない任意の部分集合 T に対して、

20

【数1】

$$A(T) = \bigcap_{A_i \in T} A_i - \bigcup_{A_i \in S-T} A_i$$

で定義される集合を求めていいればよい。これは、別の言い方で言えば、 A_1, \dots, A_n のいずれかに含まれる全データを、それが A_1, \dots, A_n のうちのどれには含まれ、どれには含まれないかに基づいて分類していることに相当する。

【0035】

ユーザが、当初の検索式のいずれかの解集合中に含まれる全データを閲覧することさえできれば良い場合は、このようにして重複を除去した分類を出力することによって、ユーザは当初の三つの検索式の解そのものを閲覧する場合に比べて、同一のデータを複数回見ないで済むことになり、閲覧の利便性が向上する。

30

【0036】

また、このようにして生成された分類から、あとで必要に応じて、当初の検索式の解集合を合成することも可能である。例えば、前述の三つの検索式の解集合 A_1, A_2, A_3 の例では、一つ目の検索式の解集合 A_1 は、以下の四つの解集合

$$\begin{array}{lll}
 A_1 & A_2 & A_3 \\
 (A_1 \cap A_2) - A_3 \\
 (A_1 \cap A_3) - A_2 \\
 A_1 - (A_2 \cup A_3)
 \end{array}$$

40

の集合和を求めればよい。 A_2, A_3 についても同様である。この点も、図 3 から容易に確認できる。

【0037】

より一般には、ある解集合 A_i を合成するには、以下の式を用いればよい。

【数2】

$$A_i = \bigcup_{A_i \in T} A(T)$$

これは別の言い方で言えば、 A_i に含まれるとされていた分類の全ての和集合を取ることに相当する。上式が成り立つことについては、複数の集合をベン図で表わした際の、ベン

50

図中の各領域と各集合の間の関係に関する一般的に知られている数学的性質より明らかである。

【0038】

よって、上述の方法で生成した重複を含まない各分類を、ディスクに保存したり、ネットワーク等を介して他の計算機へ送信したりし、その後に、当初の検索式の解集合を合成してユーザに呈示することによって、ユーザに呈示される内容を当初の検索式の解集合から変えることなく、データの保存のための記憶領域やデータの送信のための通信コストを低減することができる。

【0039】

(第二の実施の形態)

10

次に、本発明の第二の実施の形態の詳細について、詳しく説明する。本実施形態では、第一の実施の形態同様、取出し要求をSQLによって行う関係データベースに本発明を適用する場合の一実施形態について説明する。第一の実施形態との違いは、データを取り出した後に上述のような分類を生成するのではなく、与えられた検索式をもとに合成した検索式を用いることで、直接、上述の分類に当たるデータを取り出す点である。

【0040】

図4に、本実施形態におけるシステムの全体構成図を示す。本実施形態におけるシステムは、関係データを格納する格納装置4と、ユーザからSQLで記述された検索式の入力を受け取る入力装置1と、入力装置1が受け取った一つまたは複数の検索式をもとに新たな検索式を合成する検索式合成装置2と、検索式合成装置2が生成した一つまたは複数の検索式を受け取って、各検索式について、格納装置4中の関係データから、その検索式の解となるデータの集合を取り出す検索装置3と、検索装置3が取り出したデータを受け取って、当初の検索式の解を合成する検索解合成装置5と、検索解合成装置5が生成したデータを受け取って出力する出力装置6からなる。

20

【0041】

これらの構成要素は全て同じ計算機上にあってもよいし、あるいは、図4に示すように、格納装置4と検索装置3のみが、サーバとなる計算機7の上にあって、入力装置1、検索式合成装置2、検索解合成装置5、出力装置6は、サーバ7とネットワーク8を介して接続されたクライアント計算機9の上にあるという配置でもよい。

30

【0042】

以下、本実施形態において、入力装置が検索式の集合を受け取った際の処理手順について説明する。例として、以下の三つの検索式が与えられた場合を考える。

SELECT 氏名 FROM 社員 WHERE 年齢 > 50

SELECT 氏名 FROM 社員 WHERE 年齢 > 40 AND 所属 = 経理

SELECT 氏名 FROM 社員 WHERE 年齢 > 30 AND 所属 = 営業

一つ目の検索式は、社員データの一覧から、年齢が50を超える全社員の氏名を取り出す要求を表わしている。また、二つ目の検索式は、社員データの一覧から、年齢が40を超え、かつ所属が経理であるような全社員の氏名を取り出す要求を表わしている。同様に、三つ目の検索式は、社員データの一覧から、年齢が30を超え、かつ所属が営業であるような全社員の氏名を取り出す要求を表わしている。

40

【0043】

この場合、これらの検索式の解集合は、互いに重複する要素を含みうる。例えば、年齢が50を超える経理所属の社員の氏名は、一つ目の検索式の解集合と二つ目の検索式の解集合の双方に、要素として現れる。同様に、年齢が50を超える営業所属の社員の氏名は、一つ目の検索式の解集合と三つ目の検索式の解集合の双方に、要素として現れる。

【0044】

よって、本システムでは、検索式合成装置がこれらの検索式から、以下の式で表現されるような解集合を取り出す検索式を合成して格納装置へと送り、これらを用いて重複を取り除いた形でデータの取り出しを行う。

```
( A1 A2 ) - A3
( A1 A3 ) - A2
( A2 A3 ) - A1
A1 - ( A2 A3 )
A2 - ( A1 A3 )
A3 - ( A1 A2 )
```

ただし，ここで，A1，A2，A3は，それぞれ上の三つの検索式の解集合を表わし，
は集合の積演算， は集合の和演算， - は集合の差演算を表わすものとする．このような
解集合を取り出すSQL検索式は，SQLのINTERSECT構文とUNION構文と
DIFFERENCE構文を用いて作成可能で，例えば，集合(A1 A2) - A3を取り出す検索式は，以下のようになる．

```
(SELECT 氏名 FROM 社員 WHERE 年齢 > 50
INTERSECT
SELECT 氏名 FROM 社員 WHERE 年齢 > 40 AND 所属 = 経理
) DIFFERENCE
SELECT 氏名 FROM 社員 WHERE 年齢 > 30 AND 所属 = 営業
```

【0045】

同様に，集合A1 - (A2 A3)を取り出す検索式は以下のようになる．

```
SELECT 氏名 FROM 社員 WHERE 年齢 > 50
DIFFERENCE
(SELECT 氏名 FROM 社員 WHERE 年齢 > 40 AND 所属 = 経理
UNION
SELECT 氏名 FROM 社員 WHERE 年齢 > 30 AND 所属 = 営業
)
```

他の6つについても同様である．

【0046】

実は，上の(A1 A2) - A3に対応する検索式は，以下のようなより簡単なSQLに等価変形することも可能である．

```
SELECT 氏名 FROM 社員 WHERE 年齢 > 50 AND 所属 = 経理
また，同じ社員データの所属の項目の値が，経理でもあり同時に営業でもあることはありえないことを考えれば，A1 A2 A3や(A2 A3) - A1に対応する検索式は，  
解集合が常に空集合であるため，必要ないことがわかる．しかし，このような，与えられた検索式の簡単化や解が常に空集合となる検索式の判定の技術は，先行技術が存在する，  
当発明とは独立した技術であるので，ここでは詳しくは述べない．
```

【0047】

上述のようにして作成した7つの検索式の解集合が，互いに重複する要素を含まないことは，第一の実施形態の説明の中でも述べたように，直感的には，これらの7つの検索式が図3に示す，A1，A2，A3の関係を表わすベン図中の7つの領域に対応する集合を解集合とする検索式であることから容易にわかる．

【0048】

上の7つの検索式の解を受け取った検索解合成装置は，これらを用いて，当初の三つの検索式の解集合A1，A2，A3を合成する．例えば，一つ目の検索式の解集合A1は，以下の四つの解集合

```
A1 A2 A3
( A1 A2 ) - A3
( A1 A3 ) - A2
A1 - ( A2 A3 )
```

の集合和を求めればよい．A2，A3についても同様である．この点も，図3から容易に確認できる．

【0049】

10

20

30

40

50

このようにして合成した検索式を用いてデータの取り出しを行うことで、検索装置が格納装置から同じデータを複数回取り出すことを避けることができる、検索装置の部分の処理の効率を向上することができる。例えば、単純に、与えられた三つの検索式を順番に実行した場合、年齢が50を超えていて所属が経理であるような社員の氏名のデータは、一つ目の検索式の実行の際と、二つ目の検索式の実行の際の、二回取り出されることになるが、上述の7つの検索式を用いてデータの取り出しを行った場合、いかなるデータも、二回以上、検索装置によって取り出されることはない。

【0050】

また、検索式合成装置や検索解合成装置がネットワーク等を介して接続されたクライアント上にある場合には、サーバからクライアントへは重複を取り除いたデータが送信され、クライアント上で、重複を含む当初の検索式の解集合を合成するため、ネットワークの通信コストを軽減することができる。10

【0051】

また、ユーザが、当初の検索式の解集合中に含まれる全データを閲覧することさえできれば、必ずしも当初の検索式の解集合の形に合成する必要がない場合は、検索解合成装置の機能をオフにし、合成された検索式の解をそのまま出力装置に送って出力してもよい。これにより、ユーザは、当初の三つの検索式の解を閲覧する場合に比べて、同一のデータを複数回見ないですむことになり、閲覧の利便性が向上する。

【0052】

ここまで検索式が三つの場合の例を用いて説明したが、一般的には以下のような手順で検索式の合成を行う。まず、検索式の集合 $S = \{ Q_1, \dots, Q_n \}$ が与えられたとする。その場合、 S の空でない任意の部分集合 T に対して、 T に含まれる全ての検索式の積（すなわち、全ての検索式の条件を満たすデータを取り出す検索式）と、 T に含まれない全ての検索式の和（すなわち、いずれかの検索式の条件を満たすデータを取り出す検索式）との、これら二つの検索式の差（すなわち、前者の検索式の条件は満たすが、後者の検索式の条件は満たさないデータを取り出す検索式）にあたる検索式を作成する。すなわち、 S の空でない任意の部分集合 T に対して、20

【数3】

$$Q(T) = \bigcap_{Q_i \in T} Q_i - \bigcup_{Q_i \in S-T} Q_i$$

30

で定義される検索式 $Q(T)$ を求める。ここでは、 \cap 、 \cup 、 $-$ はそれぞれ、検索式の積演算、和演算、差演算を表わし、SQLでは、前述のように、各々、INTERSECT構文、UNION構文、DIFFERENCE構文で表現できる。ここでは、取り出し要求の記述にSQL検索式を用いる場合を例にとっているが、他の多くの取り出し要求記述言語においても、同様の積、和、差にあたる取り出し要求の合成が容易に可能である。そして、全ての空集合でない部分集合 T に対する $Q(T)$ を使って、データを取り出す。上の式は、複数の集合をベン図で表わした際の、ベン図中の各領域に対応する集合を表わす式を、そのまま検索式の式に置き換えたものであり、よって、この式で定義される各検索式の解集合が重複を含まないことは明らかである。40

【0053】

そして、それらの解から当初の検索式 Q_i の解を合成するには、データの取り出しに用いた全ての検索式のうち、 Q_i を含む T を用いて作成された検索式の全ての解集合の和集合を取ればよい。すなわち、 Q_i の解集合を A_i 、 $Q(T)$ の解集合を $A(T)$ と書くことになると、

【数4】

$$A_i = \bigcup_{Q_i \in T} A(T)$$

である。上記の式が成り立つことについても、複数の集合をベン図で表わした際の、各領50

域と，各集合の間の関係に関する，一般的に知られている数学的性質より明らかである。

【0054】

また，ここで，前述のような検索式の簡単化や不要な検索式の除外を行った場合は，検索解合成装置は， Q_i の解の合成については， Q_i を含む T を用いて作成された検索式，またはそれを簡単化した検索式のうち，除外されなかったもの全てに対する解集合の和集合を取ることで，これを行う。

【0055】

(第三の実施の形態)

次に，本発明の第三の実施の形態の詳細について，詳しく説明する。本説明においては，本発明を，木構造で表現できるデータの一種である XML データを格納して，これに対するデータ取り出し要求の記述に XPath と呼ばれるパス式の一種（より詳しくは，下記非特許文献 3 参照）を用いる，XML データベースシステムに適用する場合の一実施形態について説明する。

【非特許文献 3】 J. Clark, S. DeRose 著 "XML Path Language XPath Version 2.0 - W3C Working Draft" <http://www.w3.org/TR/xpath20>

【0056】

X Path は，XML の木構造中の任意の深さのノードを根とする部分木の集合を取り出す要求を記述する言語であり，よって，この実施形態においては，第一，第二の実施形態で生じたような，同じデータが複数の検索式の解集合に重複して現れるという種類の重複に加えて，取り出し要求が取り出すある部分木が，その取り出し要求自身または同時に与えられた別の取り出し要求が取り出す別の部分木の，その一部をなす部分木として重複して出現するという種類の重複も起こりうる。よって，本実施形態においては，このような重複についても取り除く処理を行う。本実施形態においては，これらの重複を，取り出したデータを調べることによって発見する。

【0057】

本実施形態における XML データベースシステムの全体構成図は，第一の実施形態の場合と同様，図 2 に示すようになる。本実施形態における XML データベースシステムは，大きな XML データの木構造を格納する格納装置 4 と，ユーザから X Path で記述された検索式の入力を受け取る入力装置 1 と，入力装置 1 が受け取った一つまたは複数の検索式を受け取り，各検索式について，格納装置 4 中の XML データから，その検索式の解となる部分木の集合を取り出す検索装置 3 と，検索装置 3 が取り出したデータを受け取って，このデータを調べて発見した重複を取り除く重複除去装置 10 と，重複除去装置が生成したデータを受け取って出力する出力装置 6 からなる。

【0058】

以下，本実施形態において，入力装置が検索式または検索式の集合を受け取った際の処理手順について説明する。例えば，入力装置が以下の三つの X Path 検索式の集合を受け取ったとする。

/ a / b
/ a / c / *
/ a / * / d

一つ目の / a / b は，格納されている XML データの木構造の根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の b というタグ名を持つノードを取り出せという要求を表わす。一方，二つ目の検索式の中の * は「任意のタグ名」を表わし，/ a / c / * は，根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の c というタグ名を持つノードの，さらにその子供の任意のタグ名を持つノードを取り出せという要求を表わす。同様に，三つ目の / a / * / d は，根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の任意のタグ名を持つノードの，さらにその子供の d というタグ名を持つノードを取り出せという要求を表わす。

10

20

30

40

50

【0059】

よって、これら三つの検索式の解集合は、互いに重複して現れる要素を含みうる。具体的には、まず、/ a / c / d という検索式にマッチするようなノードは、全て / a / c / * と / a / * / d の双方の解集合の要素として重複して現れる。また、/ a / * / d の解集合中の要素のうち、/ a / b / d という検索式にマッチするようなノードは、/ a / b の解集合中のある要素の部分木としても重複して現れる。

【0060】

例えば、図 5 に示す木構造で表現されるような XML データに対して上の三つの検索式を評価した場合、それぞれの検索式に対する解集合は、図 6 に示したようになる。ここで、/ a / c / * の解集合中の二つの部分木のうち、左の物は、/ a / * / d の解集合中にも重複して現れている。また、/ a / * / d の解集合中の三つの部分木のうち、一番左の物は、/ a / b の解集合中の部分木の、その一部をなす部分木としても重複して出現している。

10

【0061】

よって、この場合は、入力装置が前述の三つの検索式を検索装置に送って、検索装置がこれらに対する解集合を取り出し、これを重複除去装置に送った後、重複除去装置は送られた解集合を調べて、これらの重複のいずれか一方を除去し、例えば、図 7 に示す結果を出力装置へと送り、これを出力装置が出力する。

【0062】

このように、同じ部分木が複数の解集合中に要素として現れるような重複や、ある解集合中のある要素が、別のあるいは同じ解集合中のある要素の部分木として現れるような重複を取り除いて出力することによって、ユーザが同じデータを複数回見ることがなくなり、ユーザの閲覧の利便性を増すことができる。

20

【0063】

上の例では、複数の検索式を受け取って、ある検索式の解が、別の検索式の解、またはその一部をなす部分木として重複して出現する例を示したが、XPath で用いられる構文「|」や「//」を含む検索式が与えられた場合には、ある一つの検索式の解集合中のある部分木が、同じ解集合中の別の部分木の一部をなす部分木として重複して出現する場合もある。例えば、以下の検索式が与えられたとする。

/ a / b | / a / * / d

30

「|」は、二つの検索式の「和」を表わし、上の検索式の場合、/ a / b と / a / * / d の二つの検索式の解集合の和集合が、解集合となる。すなわち、図 6 中の左端の集合と右端の集合の和集合が解集合となる。よって、右端の集合の一番左の部分木は、他の解である、左端の集合中の部分木の、その一部をなす部分木としても重複して出現することになる。よって、重複除去装置は、このような一つの解集合中の重複についても、これを除去する。

【0064】

同様に、「//」を含む検索式の例として以下の検索式を考える。

/ a / b // *

「//」は、直感的には「その下の任意の深さの子孫の」という意味を表わし、上の検索式は、根からスタートして、その子供の a というタグ名を持つノードの、その子供の b というタグ名を持つノードの、その下の任意の深さの子孫の、任意のタグ名を持つノードを取り出せという要求を表わす。よって、この検索式を図 5 に示したデータに対して評価した場合、図 8 に示す部分木集合が解集合となる。この解集合中の右に示した解は、左に示した解の部分木としても重複して出現する。よって、重複除去装置は、このような重複についても、これを除去する。

40

【0065】

(第四の実施の形態)

次に、本発明の第四の実施の形態の詳細について、詳しく説明する。本説明においては、本発明を、第三の実施形態と同様に、木構造で表現できるデータの一種である XML デー

50

タを格納して、これに対するデータ取り出し要求の記述に X P a t h と呼ばれるパス式の一種（より詳しくは、上記非特許文献 3 参照）を用いる、X M L データベースシステムに適用する場合の一実施形態について説明する。

【 0 0 6 6 】

第三の実施の形態の場合と同様、この実施形態においては、第一、第二の実施形態で生じたような、同じデータが複数の検索式の解集合に重複して現れるという種類の重複に加えて、取り出し要求が取り出すある部分木が、その取り出し要求自身または同時に与えられた別の取り出し要求が取り出す別の部分木の、その一部をなす部分木として重複して出現するという種類の重複も起こりうる。よって、本実施形態においては、これら双方の種類の重複を取り除く処理を行う。また、本実施形態においては、与えられた検索式をもとに合成した検索式を用いてデータを取り出すことで、これらの重複を取り除いた形でデータを取り出す。

【 0 0 6 7 】

図 9 に、本実施形態における X M L データベースシステムの全体構成図を示す。本実施形態における X M L データベースシステムは、大きな X M L データの木構造を格納する格納装置 4 と、ユーザから X P a t h で記述された検索式の入力を受け取る入力装置 1 と、入力装置 1 が受け取った一つまたは複数の検索式をもとに新たな検索式を合成する検索式合成装置 2 と、検索式合成装置 2 が生成した一つまたは複数の検索式を受け取って、各検索式について、格納装置 4 中の X M L データから、その検索式の解となる部分木の集合を取り出す検索装置 3 と、検索装置 3 が取り出したデータを受け取って出力する出力装置 6 からなる。

【 0 0 6 8 】

以下、本実施形態において、入力装置が検索式または検索式の集合を受け取った際の処理手順について説明する。例として、入力装置が前述の第三の実施の形態の場合と同様、以下の三つの X P a t h 検索式の集合を受け取ったとする。

```
/ a / b
/ a / c / *
/ a / * / d
```

【 0 0 6 9 】

この場合、検索式合成装置は、まず、異なる検索式の解集合間のデータの重複を取り除くため、上記の三つの検索式を、以下のように変形する。

```
/ a / b
/ a / c / * - / a / b
/ a / * / d - / a / b - / a / c / *
```

より一般には、全検索式に適当な順番を決め、各検索式の末尾に、その検索式より順番の若い検索式全てを「-」でつないで並べれば良い。このようにして生成された検索式を、さらに、ある解が他の解の部分木として現れるような重複を取り除くため、以下のように変形する。

```
/ a / b - / a / b / / * - / a / c / * / / * - / a / * / d / / *
/ a / c / * - / a / b
- / a / b / / * - / a / c / * / / * - / a / * / d / / *
/ a / * / d - / a / b - / a / c / *
- / a / b / / * - / a / c / * / / * - / a / * / d / / *
```

より一般には、各検索式の後ろに、全検索式の末尾に / / * を追加したものを、- でつなないで並べる。このように変形した検索式を用いてデータを取り出すことで、重複を除去した状態でデータを取り出すことができ、データの取り出し処理のコストの低減や、ユーザの閲覧の利便性の向上が期待できる。

【 0 0 7 0 】

ここで、上に示した変形後の検索式に対しては、簡単化や、解集合が常に空集合となる検索式の除去が可能である。しかし、前述の通り、これらは本発明とは独立した技術である

10

20

30

40

50

ので，ここでは詳しくは述べない．また，ここでは，木構造データからの部分木集合の取り出し要求の記述に X P a t h を用いる場合を例にとって，検索式の合成方法について説明したが，他の多くの取り出し要求記述言語においても，同様の合成が可能である．

【 0 0 7 1 】

(第五の実施の形態)

次に，本発明の第五の実施の形態の詳細について，詳しく説明する．本説明においては，本発明を，第三，第四の実施形態と同様に，木構造で表現できるデータの一種である X M L データを格納して，これに対するデータ取り出し要求の記述に X P a t h と呼ばれるパス式の一種（より詳しくは，上記非特許文献 3 参照）を用いる， X M L データベースシステムに適用する場合の一実施形態について説明する．

10

【 0 0 7 2 】

この実施形態においても，第一，第二の実施形態で生じたような，同じデータが複数の検索式の解集合に重複して現れるという種類の重複に加えて，第三，第四の実施の形態で生じたような，取り出し要求が取り出すある部分木が，その取り出し要求自身または同時に与えられた別の取り出し要求が取り出す別の部分木の，その一部をなす部分木として重複して出現するという種類の重複が起こりうる．よって，本実施形態においては，これら双方の種類の重複を取り除く処理を行う．また，本実施形態においては，与えられた検索式をもとに合成した検索式を用いてデータを取り出すことで，これらの重複を取り除いた形でデータを取り出す．さらに，本実施形態では，そのようにして重複を取り除いたデータから，必要に応じて，当初の検索式の解集合を合成することが可能である．

20

【 0 0 7 3 】

図 1 に，本実施形態における X M L データベースシステムの全体構成図を示す．本実施形態における X M L データベースシステムは，大きな X M L データの木構造を格納する格納装置 4 と，ユーザから X P a t h で記述された検索式の入力を受け取る入力装置 1 と，入力装置 1 が受け取った一つまたは複数の検索式をもとに新たな検索式を合成し，それとともに，合成された検索式の解からの，当初の検索式の解の取り出し方法を指示する記述を生成する検索式合成装置 2 と，検索式合成装置 2 が生成した一つまたは複数の検索式を受け取って，各検索式について，格納装置 4 中の X M L データから，その検索式の解となる部分木の集合を取り出す検索装置 3 と，検索装置 3 が取り出したデータ，および，検索式合成装置 2 が生成した，当初の検索式の解の取り出し方法を指示する記述を受け取って，当初の検索式の解を合成する検索解合成装置 5 と，検索解合成装置 5 が生成したデータを受け取って出力する出力装置 6 からなる．

30

【 0 0 7 4 】

これらの構成要素は全て同じ計算機上にあってもよいし，あるいは，格納装置 4 と検索装置 3 のみが，サーバとなる計算機 7 の上にあって，入力装置 1 ，検索式合成装置 2 ，検索解合成装置 5 ，出力装置 6 は，サーバ 7 とネットワーク 8 を介して接続されたクライアント計算機 9 の上にあるという配置でもよい．

【 0 0 7 5 】

以下，本実施形態において，入力装置が検索式または検索式の集合を受け取った際の処理手順について説明するが，ここでは，これを理解の容易さを考えて，三つの場合に分けて説明する．まず最初に，受け取った X P a t h 検索式が，前述の「 / / 」という構文を含まず，かつ，全ての検索式が X M L データの木構造中の同じ深さのノードを取り出す要求である場合の処理手順について説明する．この場合の処理手順は，当発明を第一の実施形態のように木構造データ以外のデータ管理システムに適用している場合など，ある解が別の解の部分木として重複する場合を考慮しない場合の処理手順と同様のものになる．二番目に，検索式が「 / / 」を含まないが，各検索式が取り出すノードの深さは全て同じではない場合の処理手順について説明する．最後に，検索式が「 / / 」を含む場合の処理手順について説明する．以上，理解の容易さのために三つの場合にわけて説明するが，實際には，最後に説明する処理手順が，全ての場合に対応する最も一般的な処理手順になっており，この処理手順を，一番目，二番目に取り上げるような特殊な場合に適用すれば，一番

40

50

目，二番目に説明する処理手順と同等の最終結果を得ることができる。

【0076】

受け取った X P a t h 検索式が「/ /」を含まず，検索式全てが同じ深さのデータを取り出す場合の処理手順は，以下のようになる。例えば，入力装置が以下の三つの X P a t h 検索式の集合を受け取ったとする。

```
/ a / b
/ a / *
/ a / * [ c / d ]
```

一つ目の / a / b は，格納されている X M L データの木構造の根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の b というタグ名を持つノードを取り出せという要求を表わす。一方，二つ目の / a / * は，根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の任意のタグ名を持つノードを取り出せという要求を表わす。また，三つ目の検索式の中の [c / d] は「その下に， c / d というパスを持つ」という条件を表わしており， / a / * [c / d] という検索式は，根からスタートして，その子供の a というタグ名を持つノードの，さらにその子供の任意のタグ名を持つノードであって，しかも，その下に c という子ノードを持ち，さらに，その c という子ノードが d という子ノードを持っているようなものを取り出せという要求を表わす。

【0077】

よって，これら三つの検索式の解集合は，互いに重複して現れる要素を含みうる。具体的には， / a / b または / a / * [c / d] の解集合の要素となるノードは，全て / a / * の解集合の要素としても現れる。また， / a / b と / a / * [c / d] の解集合は，同じ要素を重複して持ちうる。一方，この例の場合は，三つの検索式とも木構造中の深さ 2 のノードを取り出すため，一つの解集合のある要素が，同じまたは他の解集合の要素の部分木として重複して出現することはありえない。

【0078】

よって，本システムではこの場合，与えられた検索式集合の任意の空でない部分集合，すなわちこの場合で言うと

```
{ / a / b , / a / * , / a / * [ c / d ] }
{ / a / b , / a / * }
{ / a / b , / a / * ' c / d ' }
{ / a / * , / a / * [ c / d ] }
{ / a / b }
{ / a / * }
{ / a / * [ c / d ] }
```

という七つの部分集合の各々に対して，該部分集合中の全検索式の積と，該部分集合に含まれない検索式の集合である

```
{ }
{ / a / * [ c / d ] }
{ / a / * }
{ / a / b }
{ / a / * , / a / * ' c / d ' }
{ / a / b , / a / * [ c / d ] }
{ / a / b , / a / * }
```

に含まれる全検索式の和との，両者の差に相当する七つの検索式

/ a / b	/ a / *	/ a / * ' c / d '	(検索式 1)
(/ a / b	/ a / *)	- / a / * [c / d]	(検索式 2)
(/ a / b	/ a / * ' c / d ') - / a / *	(検索式 3)
(/ a / *	/ a / * [c / d])	- / a / b	(検索式 4)
/ a / b	- (/ a / * / a / * [c / d])		(検索式 5)

/ a / * - (/ a / b / a / * [c / d]) (検索式 6)
 / a / * [c / d] - (/ a / b / a / *) (検索式 7)

を検索式合成装置が求め、これら七つの検索式を検索装置に送って検索を行う。これは、前述の第二の実施の形態の場合と同様の手順である。

【0079】

ここで、 \cdot は二つの検索式の積、すなわち、二つの検索式双方の条件を満たすノードを取り出す要求を表わす。同様に、 $+$ は二つの検索式の和、すなわち、二つの検索式双方の条件のいずれかを満たすノードを取り出す要求を表わし、また $-$ は二つの検索式の差、すなわち、前者の検索式の条件は満たすが、後者の検索式の条件は満たさないノードを取り出す要求を表わす。これらの記法は、実際の XPath での記法とは異なる（例えば、 \cdot は実際の XPath では、前述のように「|」と記述される）が、ここでは説明のわかりやすさのため、このような記法を用いることとする。上の式 1 ~ 7 に対する解集合が、互いに重複する要素を含まないことは、第二の実施の形態の場合とほぼ同様である。このように合成した検索式を用いてデータを取り出すことにより、重複したデータの取り出し、あるいは重複したデータのネットワークを介しての送信を避けることができる。10

【0080】

検索装置は、上のような合成された検索式を受け取ってその解を検索し、検索解合成装置へ対応する七つの解集合を送る。検索解合成装置は、これらの解集合のうち、 $/ a / b$ を含む部分集合を用いて作成された、検索式 1, 2, 3, 5 に対する解集合の集合和を取ることによって、当初の検索式である $/ a / b$ に対する解集合を生成する。また、 $/ a / *$ の解は、検索式 1, 2, 4, 6 に対する解集合の集合和を、 $/ a / * [c / d]$ の解については、検索式 1, 3, 4, 7 に対する解集合の和集合を取ることによって生成する。20

【0081】

また、この例の場合、式 1 は、より単純な検索式 $/ a / b [c / d]$ と等価であり、式 3 は解集合が常に空集合となる検索式である。他の式についても、同様な簡単化が可能なものや、解が常に空となるものがある。よって、検索式合成装置は、このような検索式の簡単化や、解が空となる検索式の除外を行ってもよい。このような、パス式の形で記述されている検索式の簡単化や、解集合が常に空集合となる検索式の判定のための技術については、下記非特許文献 4, 5 に示されているものなど、様々な先行技術があり、当発明とは独立した技術であるので、ここでは、詳しくは述べない。30

【非特許文献 4】 G. Mikkilau, D. Suciu 著 "Containment and Equivalence for an XPath Fragment" ACM Symposium on Principles of Database Systems (PODS) 予稿集, 頁 65 - 76, 2002

【非特許文献 5】 M. Benedikt 他著 "Structural Properties of XPath Fragments" International Conference on Database Theory (ICDT) 予稿集, 頁 79 - 95, 2003

【0082】

上述のような検索式の簡単化や除外を行った場合は、検索解合成装置は、 $/ a / b$ を含む部分集合をもとに生成された検索式の簡単化によって得られ、かつ除外されなかったものの全ての解集合の和集合を求めて $/ a / b$ の解を生成する。他の検索式の解についても同様である。これについても、前述の第二の実施形態の場合と同様である。40

【0083】

ここまででは、特定の例を用いて手順を説明したが、より一般的には、第二の実施の形態の場合と同様、与えられた検索式集合の部分集合で空でないもの全てに対して、該部分集合を T として数式 3 で定義される検索式を生成し、これら全ての検索式によってデータの取り出しを行う。また、取り出したデータからの当初検索式の解の合成も、第二の実施の形態の場合と同様、数式 4 に基づいて合成を行う。

【0084】

50

次に，検索式が「／／」を含まないが，全ての検索式が同じ深さのノードの取り出し要求ではない場合の処理手順について説明する．この場合，検索式合成装置は，以下の手順で検索式の合成を行う．

(1) まず，与えられた検索式全てに Q_1, Q_2, \dots という識別子を割り当て，さらに与えられた検索式全てをパスの長さでグループに分類する．

(2) 上の(1)で得られたグループ各々について以下の処理を行う．

(3) そのグループに含まれる全検索式の集合を S とし， S の空でない全ての部分集合の各々を T として，以下の処理を行う．

(4) そのグループより長い検索式全てから，現在考えているグループの長さの接頭辞を取り出し，これらの接頭辞の集合を P とし， P の全ての部分集合の各々を R として，以下の処理を行う．

(5) 現在のグループより短い検索式全てについて，その末尾に $/ * / \dots / *$ を追加して現在のグループと同じ長さにした検索式を作成し，これら全ての和を E とする．以上を用いて以下の式

【数5】

$$Q(T, R) = \left(\bigcap_{Q_i \in T} Q_i - \bigcup_{Q_i \in S-T} Q_i \right) \cap \left(\bigcap_{p \in R} p - \bigcup_{p \in P-R} p \right) - E$$

にあたるパス式を作成する．また，このようにして生成された各パス式に対して，順に，
V1, V2...という識別子を割り当てていく．

(6) また，それと同時に， T に含まれる検索式 Q_x, Q_y, \dots の各々について，今，作成した V_a の解となる部分木が，そのまま， Q_x, Q_y, \dots の解として取り出されるべきであることを表わす，

$Q_x (V_a, *)$

$Q_y (V_a, *)$

⋮

⋮

という項を生成する．同様に， R に含まれる接頭辞 p の各々について，その接頭辞を取り出した検索式を Q_x ，また， Q_x からその接頭辞 p を取り除いた残りの部分にあたる接尾辞を $p p$ として， V_a の解となる部分木の根からパス $* / p p$ にしたがってたどって到達するノードが， Q_x の解として取り出されるべきであることを表わす，

$Q_x (V_a, * / p p)$

⋮

⋮

という項を生成する．

【0085】

例として，以下の三つの検索式が与えられた場合について説明する．

$/ a / b / c$

$/ a / * / c / d$

$/ a / b / c [d] / *$

一つ目の検索式は，根からスタートして， a というタグ名を持つ子ノードの，その子供で b というタグ名を持つノードの，その子供で c というタグ名を持つノードを取り出す．二つ目の検索式は，根からスタートして， a というタグ名を持つ子ノードの，その任意の子ノードの，その子供で c というタグ名を持つノードの，その子供で d というタグ名を持つノードを取り出す．また，三つ目の検索式は，根からスタートして， a というタグ名を持つ子ノードの，その子供で b というタグ名を持つノードの，その子供で c というタグ名を持ち，かつ， d というタグ名の子供を持つノードの，その任意の子ノードを取り出す．

【0086】

これら三つの検索式の解集合は，以下ののようなデータの重複を含みうる．まず，二つ目の

10

20

30

40

50

検索式の解集合と三つ目の検索式の解集合とには、同じ要素が重複して出現しうる。さらに、二つ目と三つ目の検索式の解集合中の一部の要素は、一つ目の検索式の解集合中のある解の部分木として出現しうる。例えば、システムに現在格納されている XML データ中に、/ a / b / c / d という検索式にマッチするようなノードがあった場合、このノードは二つ目の検索式の解となるが、一方で、そのノードの親ノードは一つ目の検索式の解となるので、このノードは二つ目の検索式の解集合中の要素として出現すると同時に、一つ目の検索式の解集合中のある要素の部分木としても重複して出現することになる。

【 0 0 8 7 】

そこで、検索式合成装置は以下の手順で検索式の合成を行う。まず、これらの検索式に対して、識別子 Q 1, Q 2, Q 3 を割り当てる。次に、これらの検索式を、長さ 3 のグループ { Q 1 } と長さ 4 のグループ { Q 2, Q 3 } に分類する。まず、長さ 3 のグループについて考えると、その空でない部分集合は { Q 1 } のみなので、これを T として (4) 以下の処理を行う。次の (4) において、このグループより長い全ての検索式から長さ 3 の接頭辞を取り出した集合 P は { / a / * / c, / a / b / c [d] } となるので、この部分集合

```
{ / a / * / c, / a / b / c [ d ] }
{ / a / * / c }
{ / a / b / c [ d ] }
{ }
```

の各々を R とする四通りの場合について (5) を行う。今の場合、現在のグループより短い検索式はなく、E に該当する検索式は存在しない。よって、R として { / a / * / c, / a / b / c [d] } を選んだ場合は、以下のパス式が作成される。

```
/ a / b / c      / a / * / c      / a / b / c [ d ]
```

同時に、この検索式に V 1 という識別子を割り当てる。また、この検索式は検索式生成に用いる T としては { Q 1 } を用いていたので、V 1 の解が、そのまま Q 1 の解としても取り出されるべきであることを表わす項である

```
Q 1 ( V 1, * )
```

が生成される。さらに、この検索式は、検索式生成に用いる R としては、Q 2 から取り出した接頭辞である / a / * / c と Q 3 から取り出した接頭辞である / a / b / c [d] がなる集合を用いていたので、V 1 の解となる部分木の根から、/ a / * / c に対応する接尾辞である d というパスをたどって到達する要素も Q 2 の解として取り出すべきであることを表わす項である

```
Q 2 ( V 1, * / d )
```

と、V 1 の解となる部分木の根から、/ a / b / c [d] に対応する接尾辞である * というパスをたどって到達する要素も Q 3 の解として取り出すべきであることを表わす項の

```
Q 3 ( V 1, * / * )
```

が生成される。

【 0 0 8 8 】

また、ここで、V 1 は / a / b / c [d] へと簡略化可能であるので、検索式合成装置はそのような簡略化を行ってもよいが、既に述べたように、簡略化や不要な検索式の除外の具体的な手法については、当発明とは独立した技術なので、ここでは詳しく述べない。

【 0 0 8 9 】

同様に、手順 (4) で残りの三通りの R の選び方、すなわち、

```
R = { / a / * / c }
R = { / a / b / c [ d ] }
R = { }
```

として (5) を実行した場合、それぞれ、

```
V 2 : / a / b / c      / a / * / c      -      / a / b / c [ d ]
Q 1 ( V 2, * )
Q 2 ( V 2, * / d )
```

10

20

30

40

50

V 3 : / a / b / c / a / b / c [d] - / a / * / c
 Q 1 (V 3 , *)
 Q 3 (V 3 , * / *)
 V 4 : / a / b / c - / a / * / c - / a / b / c [d]
 Q 1 (V 4 , *)

という検索式 V 2 , V 3 , V 4 , および , それらからの当初の検索式の解の取り出し方法を指示する項が生成される .

【 0 0 9 0 】

次に , 長さ 4 の検索式のグループ { / a / * / c / d , / a / b / c [d] / * } について考える . このグループの空でない部分集合は ,

{ / a / * / c / d , / a / b / c [d] / * }
 { / a / * / c / d }
 { / a / b / c [d] / * }

である . よって , これらの各々を T として (4) を行う . このグループより長い検索式はないので , 接頭辞の集合 P は空集合となり , その部分集合は空集合のみである . よって , いざれを T とした場合についても , R を空集合として (5) を行う . E については , 現在のグループより短い検索式は , 長さが 3 の / a / b / c のみなので , これの長さを 4 とするために , 末尾に / * を追加して / a / b / c / * とする . その結果 , 上の三つの集合のうちの { / a / * / c / d , / a / b / c [d] / * } を T とした場合は ,

V 5 : / a / * / c / d / a / b / c [d] / * - / a / b / c / *

という検索式 V 5 , および ,

Q 2 (V 5 , *)
 Q 3 (V 5 , *)

という , V 5 からの当初の検索式の解の取り出し方法を指示する項が生成される . 同様にして , { / a / * / c / d } および { / a / b / c [d] / * } を T とした場合は ,

V 6 : / a / * / c / d - / a / b / c [d] / * - / a / b / c / *
 Q 2 (V 6 , *)
 V 7 : / a / b / c [d] / * - / a / * / c / d - / a / b / c / *
 Q 3 (V 7 , *)

という検索式 V 6 , V 7 , および , それらからの当初の検索式の解の取り出し方法を指示する項が生成される .

【 0 0 9 1 】

以上から , 当初与えられた三つの検索式に対して , 七つの検索式と , その解からの当初の検索式の解の取りしが生成される . これら七つの検索式の解集合は , 互いに重複する要素を含まず , またある解集合中のある要素が , 同じまたは別の解集合中のある要素の部分木として重複して出現することもない . 一方で , これら七つの検索式の解集合は , 当初の三つの検索式の解集合の合成に必要な全てのデータを含んでおり , これら七つの検索式の解集合から , 上で示された取りしが方法にしたがって , 当初の三つの検索式の解集合を合成することができる .

【 0 0 9 2 】

任意の検索式集合に対して , 前述の手順で作成された検索式が , 上のような性質を満たす理由については , 以下のように説明できる . まず , これらの検索式の解が互いに重複する要素を含まない点について説明する . 解集合中に重複する解を含みうるのは , 長さが同じ検索式同士のみである . 一方 , 検索式の生成に用いられる数式 5 中の

【 数 6 】

$$(\bigcap_{Q_i \in T} Q_i - \bigcup_{Q_i \in S-T} Q_i)$$

の部分は , 長さが同じ検索式のグループについて , 前述の第二の実施形態の場合や , 与えられた検索式が全て同じ長さだった場合において行ったのと同様の , ベン図の各領域に対

10

20

30

40

50

応する解集合への分類を行っていることになる。よって、数式5によって作成された各検索式は、その解集合中に重複する要素を含まない。

【0093】

次に、ある検索式の解集合中のある要素が、同じ検索式または別の検索式の解集合中の別の要素の部分木として出現しないという点について説明する。ある検索式のある解が、ある検索式のある解の部分木として重複して出現しうるのは、前者の検索式が、後者の検索式より長い場合である。しかし、数式5においては、Tに含まれる当初検索式より短い全ての検索式について、その後に*/.../*を加えて同じ長さにした物の和をEとして、これに含まれる物を-Eによって取り除いており、よって、より短い検索式の解の部分木として現れている物は必ず取り除かれることになる。

10

【0094】

次に、当初検索式の解となるノードは、必ずいずれかの合成検索式の解集合中に出現しているという点について説明する。これは、そのノードの先祖ノードの中に、いずれかの当初検索式の解となるものがある場合と、そのような先祖ノードが全くない場合の二通りに分けられる。そのような先祖ノードがある場合は、それらの先祖ノードの中で最も根に近いノードが、ある一つの合成検索式の解集合中に含まれる。よって、現在問題にしているノードは、その解の一部をなす部分木として出現する。なお、その先祖ノードが解として出現する合成検索式は、その先祖ノードにマッチする検索式全ての集合をTとし、T中の検索式より長い検索式の接頭辞のうち、その先祖ノードとマッチするもの全ての集合をRとして生成された検索式である。一方、そのような先祖ノードが全くない場合は、現在問題としているノード自身が、ある一つの合成検索式の解集合中に出現する。その合成検索式は、その部分木にマッチする全ての検索式の集合をTとし、T中の検索式より長い検索式の接頭辞のうち、そのノードとマッチするもの全ての集合をRとして生成された検索式である。

20

【0095】

次に、上述の方法で生成された各検索式の解から、上述の手法によって、正しく当初検索式の解を合成できる点について説明する。合成に用いる数式5のうちの数式6の部分は、当初検索式のうちのいずれかの解となるノード全てを、そのノードが当初検索式のうちのどれの解集合には含まれどれの解集合には含まれないかに基づいて分類していることに相当する。同様に、数式5中の

30

【数7】

$$(\bigcap_{p \in R} p - \bigcup_{p \in P-R} p)$$

の部分は、これらのノードを、そのノードを根とする部分木からは、他の、より長い検索式のうちのどれとどれの解を、どんなパス式に基づいて取り出せばよいかという観点から分類していることに相当する。よって、数式5は、必要な全ノードを、それをどの検索式の解とすればよいか、そこからどのような部分木を他の解として抜き出せば良いかに基づいて、分類していることに相当し、よって、正しく当初検索式の解を取り出すことができる。

40

【0096】

次に、検索式が「//」という構文を含む場合の処理手順について説明する。X Pathにおける「//」という構文は「その下の任意の深さの」という条件を表しており、例えば、/a//bはデータの根からスタートして、aというタグ名を持つ子ノードの、その下の任意の深さの子孫で、bというタグ名を持つノードを取り出せという要求を表す。また、//a/bは、データ中の任意の深さにあるaというタグ名を持つノードの、その子ノードでbというタグ名を持つものを取り出せという要求を表す。

【0097】

このような「//」という構文を含む検索式が与えられた場合の処理手順を記述するために、まず、検索式のi-接頭辞という概念を以下のように定義する。今、与えられた検索

50

式が

/ 1 p 1 / 2 p 2 ... / m p m

という形をしていたとする。ここで、/ 1, ..., / m は、それぞれ / であるか、または、// であるかのいずれかであり、p 1, ..., p m は、それぞれ、それらの間に挟まれた、/ と // を含まない部分である。この時、0 ≤ i ≤ m - 1 である i に対して、この検索式の i - 接頭辞は以下のように定義される。

(1) i = 0 で / i + 1 が / である場合 :

/ 1 p 1 ... / i p i

(2) i = 0 で / i + 1 が // である場合 :

/ 1 p 1 ... / i p i / 1 p 1 ... / i p i // *

10

(3) i = 0 で / 1 が / である場合 :

(いかなるノードにもマッチしない特別なパス式)

(4) i = 0 で / 1 が // である場合 :

// *

例えば、// a という長さ 1 の検索式に対しては、0 ≤ i ≤ 1 である i、すなわち、i = 0 のみについて i - 接頭辞が定義され、この検索式の 0 - 接頭辞は // * である。また、/ b // c という長さが 2 である検索式に対しては、0 ≤ i ≤ 2 である i、すなわち、i = 0 と i = 1 について i - 接頭辞が定義され、この検索式の 0 - 接頭辞は // * である、1 - 接頭辞は / b / b // * である。

【0098】

20

上の i - 接頭辞の定義は、構文「//」が「/」、または、「* /」、または、「* / * /」、または、「...」という無限個の場合を表現するものであることを考慮して、通常のパスの接頭辞の概念を自然に拡張した物になっている。

【0099】

上のように、i - 接頭辞を定義すると、構文「//」を含む検索式に対する処理手順は、以下のようになる。

(1) まず、与えられた検索式全てに、Q 1, Q 2, ... という識別子を割り当てる。

(2) 与えられた全検索式の集合を S とし、S の空でない全ての部分集合の各々を T として、以下の処理を行う。

(3) 与えられた各検索式に対して定義される全ての i - 接頭辞の集合を P とし、P の全ての部分集合の各々を R として、以下の処理を行う。

30

(4) 全ての検索式について、その末尾に // * を追加した検索式を作成し、これら全ての和を E として、以下の式

【数 8】

$$Q(T, R) = \left(\bigcap_{Q_i \in T} Q_i - \bigcup_{Q_i \in S-T} Q_i \right) \cap \left(\bigcap_{p \in R} p - \bigcup_{p \in P-R} p \right) - E$$

にあたるパス式を作成する。また、このようにして作成された各パス式に対して、順に、V 1, V 2, ... という識別子を割り当てていく。

40

(5) また、それと同時に、T に含まれる検索式 Q x, Q y, ... の各々について、今、作成した V a の解となる部分木が、そのまま、Q x, Q y, ... の解として取り出されるべきであることを表わす、

Q x (V a, *)

Q y (V a, *)

:

:

という項を生成する。同様に、R に含まれる i - 接頭辞 p の各々について、その接頭辞を取り出した検索式を Q x、また、Q x からその接頭辞 p を取り除いた残りの部分にあたる接尾辞を p p として、V a の解となる部分木の根からパス * p p にしたがって到

50

達するノードが， $Q \times$ の解として取り出されるべきであることを表わす，

$Q \times (V a, * pp)$

:

:

という項を生成する。

【0100】

上の手順は，構文「//」を含む検索式はその長さ，すなわち取り出す部分木の現れる深さが可変になる点を考慮して，前述の「//」を含まない検索式集合に対する手順を自然に拡張した物になっている。また，上の手順は，別の言い方をすると，与えられたパス式のいずれかの解となる部分木のうちで，他の解の一部をなす部分木とはならない物の全てを，それがどのパス式の解集合には含まれどのパス式の解集合には含まれないかという点と，それがどのパス式のどの接頭辞にはマッチしどのパス式のどの接頭辞にはマッチしないかという点の，以上の二点に基づいて分類し，この各分類を取り出すパス式を生成していることに相当する。

10

【0101】

例として，以下の二つの検索式が与えられた場合について説明する。

// a
/ b // *

一つ目の検索式は，任意の深さにあるノードで a というタグ名を持つものを根とする部分木を取り出す。二つ目の検索式は，データの根からスタートして，b というタグ名を持つ子ノードの，その下の任意の深さにある任意のタグ名を持つノードを根とする部分木を取り出す。この二つの検索式は，以下ののようなデータの重複を含みうる。まず，/ b // a という検索式にマッチするようなノードがあった場合，そのノードは上の二つの検索式の双方の解集合中に要素として含まれる。また，/ b // * // a という検索式にマッチするようなノードがあった場合，そのノードは一つ目の検索式の解集合中の要素として出現すると同時に，二つ目の検索式の解集合中の要素の部分木としても重複して出現することになる。同様に，/ b // a // * という検索式にマッチするようなノードがあった場合，そのノードは二つ目の検索式の解集合中の要素として出現すると同時に，一つ目の検索式の解集合中の要素の部分木としても重複して出現することになる。

20

【0102】

30

そこで，検索式合成装置は，以下の手順で検索式の合成を行う。まず，これらの検索式に Q_1 ， Q_2 という識別子を割り当てる。次に，検索式の集合 $\{ // a, / b // * \}$ の空でない部分集合

$\{ // a, / b // *\}$
 $\{ // a \}$
 $\{ / b // *\}$

の各々に対して，これを T として(3)以下の処理を行う。与えられた検索式に対して定義される全ての i - 接頭辞の集合は

$\{ // *, / b / b // *\}$

であるので，この全ての部分集合に対して，その各々を R として(4)以下の処理を行う。例えば $\{ // a, / b // * \}$ を T ， $\{ // *, / b / b // *\}$ を R とした場合，

$V : // a / b // * // * (/ b / b // *)$
- $(// a // * / b // * / / *)$

40

という検索式 V と， V からの当初の検索式の解の取り出し方法を表す項の，

$Q_1 (V, *)$
 $Q_2 (V, *)$
 $Q_1 (V, * / / a)$
 $Q_2 (V, * / b / / *)$
 $Q_2 (V, * / / *)$

50

が生成される。ここで、//aはQ1の0-接頭辞に対する接尾辞、/b//*はQ2の0-接頭辞に対する接尾辞、//*はQ2の1-接頭辞に対する接尾辞である。

【0103】

ただし、上のVは、全てのノードにマッチしない を積の成分として含んでいるため、常に解集合が空集合となる検索式である。よって、このように を積の成分として含んでいて、解集合が空集合となることが容易に判定可能なものについては、検索式合成装置で取り除くことにもよい。同様に、//*が負の成分として含まれる検索式も、常に解集合が空集合となることが容易に判定可能なので、検索式合成装置が取り除くことにもよい。さらには「//*」と「-」は省略しても検索式全体の意味が変わらないことが容易にわかるため、これらを省略することによる検索式の簡単化を、検索式合成装置が行うことにもよい。ここでは、これらの特に容易な検索式の除去と簡単化については検索式合成装置が行うものとする。その場合、Rとしては{ //*, /b / b//* }と{ //* }の場合のみを考えればよい。結果として、三通りのTと二通りのRの組み合わせで、以下の六つの検索式が生成される。

V 1 1 :	//a	/b//*	(/b / b//*)	
			- (//a//* / b//*//*)	
V 1 2 :	//a	/b//*	- (/b / b//*)	
			- (//a//* / b//*//*)	
V 1 3 :	//a	(/b	/b//*) - /b//*	
			- (//a//* / b//*//*)	20
V 1 4 :	//a	- /b//*	- (/b / b//*)	
			- (//a//* / b//*//*)	
V 1 5 :	/b//*	(/b	/b//*) - //a	
			- (//a//* / b//*//*)	
V 1 6 :	/b//*	- //a	- (/b / b//*)	
			- (//a//* / b//*//*)	

さらに、これらから当初のQ1とQ2の解の取出し方法を表す項として、

Q1 (Vi, *) ただし $i = 11, 12, 13, 14$

Q2 (Vi, *) ただし $i = 11, 12, 15, 16$

Q1 (Vi, *//a) ただし $i = 11, 12, 13, 14, 15, 16$

Q2 (Vi, *//*) ただし $i = 11, 13, 15$

が生成される。ここで、//aはQ1の0-接頭辞である//*に対する接尾辞、//*はQ2の1-接頭辞である/b / b//*に対する接尾辞である。

【0104】

実は、ここで、上のV11～V16は以下のようにさらに簡単化可能である。

V 1 1 :	/b/a			
V 1 2 :				
V 1 3 :				
V 1 4 :	//a - /b//* - //a//*			
V 1 5 :	/b//* - //a - //a//*			
V 1 6 :				

よって、検索式合成装置は、このようなさらなる簡単化を行ってもよい。しかし、前述の通り、このような簡単化のための具体的な技術については、ここでは詳しくは述べない。

【0105】

以上の合成された検索式を用いてデータの取出しを行うことで、重複したデータの取出しを排除することができる。この点について、以下に具体的なデータの例を使って示す。図10に示したデータに対して、//aと/b//*を実行した場合の解は、図11に示す通りになる。図11からわかるように、これらの解は多くのデータの重複を含み、非常に冗長である。一方、図10に示したデータに対してV11～V16を実行した場合には、V12, V13, V14, V16の解は空集合となり、V11, V15の解は図12に示

すようになる。そして、これらの解から上述の項に従って、//aと/b//*の解を取り出すと、正しく図11と同じ解が取り出せることがわかる。

【0106】

同様の重複の問題は、検索式が「//」を含む場合には、検索式が一つだけ与えられた場合にも生じる。例として、図13に示すデータに対して、

//a//*//*

という検索式が与えられたとする。この場合の解は、図14に示すようになる。しかし、この図からわかるように、この解にはfを根とする部分木が、一度は解として、一度は他の解の一部をなす部分木として、重複して現れている。

【0107】

そこで、上述の手順によって検索式の合成を行うことを考える。まず、上の検索式をQ1とする。検索式が一つしかないため、Tの選び方はT = {Q1}のみである。接頭辞の集合は、{//*, //a, //a//*}であり、この部分集合の選び方は8通りある。しかし、//*については、前述の例の場合と同じように、これをRに含めない時に生成される検索式は必ず解が空集合となる。よって、Rについては、

R = { //* , //a , //a//* }
 R = { //* , //a }
 R = { //* , //a//* }
 R = { //* }

の四通りを考えればよく、以下の四つの検索式が生成される。

V21 : //a//*//* //* //a //a//* - //a//*//*//*
 V22 : //a//*//* //* //a - //a//* - //a//*//*//*
 V23 : //a//*//* //* //a//* - //a - //a//*//*//*
 V24 : //a//*//* //* - //a - //a//* - //a//*//*//*

さらに、これらから当初のQ1の解の取り出し方法を表す項として、

Q1 (Vi, *) ただし i = 21, 22, 23, 24
 Q1 (Vi, *//a//*) ただし i = 21, 22, 23, 24
 Q1 (Vi, *//*//*) ただし i = 21, 22
 Q1 (Vi, *//*) ただし i = 21, 23

が生成される。二番目の項中の//a//*//*は、0-接頭辞//*に対応する接尾辞であり、同様に、三番目と四番目の項中の//*//*と//*は、1-接頭辞と2-接頭辞に対する接尾辞である。

【0108】

実は、V21～V24の検索式は以下のようにさらに簡略化できるので、検索式合成装置はそのような簡略化を行ってもよいが、前述の通り、ここでは詳しくは述べない。

V21 : //a/a/a - //a//*//*//*
 V22 : //a//a/a - //a//* - //a//*//*//*
 V23 : //a/a/* - //a - //a//*//*//*
 V24 : //a//*/* - //a - //a//* - //a//*//*//*

これらの検索式を図13に示すデータに対して実行した場合、V21, V22の解は空集合となり、V23, V24の解は図15に示した通りになる。図15からわかるように、これらの解集合は、全くデータの重複を含まない。そして、これらの結果に対して、上述の取り出し方法を表す各項を実行すると、正しくQ1の解を取り出すことができる。

【0109】

以上のように、当初与えられた一つの検索式に対して、四つの検索式と、その解からの当初の検索式の解の取り出し方が生成され、これら四つの検索式の解集合は、互いに重複する要素を含まず、またある解集合中のある要素が、同じまたは別の解集合中のある要素の部分木として重複して出現することもない、一方で、これら四つの検索式の解集合は、当初の検索式の解集合の合成に必要な全てのデータを含んでおり、これら四つの検索式の解集合から、上で示された取り出し方法にしたがって、当初の検索式の解集合を合成することができる。

10

20

30

40

50

できる。

【0110】

任意の検索式集合に対して、前述の手順で作成された検索式が、上のような性質を満たす理由については、前述の「//」を含まない場合の手順に対する説明とほぼ同様にして説明できる。

【0111】

この明細書で示した実施形態では、木構造データから部分木の集合を取り出すシステムに本発明を適用する際の処理に関しては、パス式の形で検索式が与えられる場合を例にとって説明したが、本発明は、その他のさまざまな形で取り出し要求が与えられる場合においても、同様の考え方に基づいて適用可能である。

10

【0112】

(第六の実施の形態)

次に、本発明の第六の実施の形態について、説明する。本説明では、本発明を、インターネット上で公開されているXMLデータ連続問い合わせシステムのためのプロクシーサーバに適用する場合の一実施形態について説明する。

【0113】

図16に、本実施形態におけるXMLデータ連続問い合わせシステムとプロクシーサーバの全体構成図を示す。本実施形態においては、XMLデータ連続問い合わせシステム11がインターネット12の上に公開されているものとする。XMLデータ連続問い合わせシステムとは、XMLデータを格納し、ユーザからの検索式登録要求を受けるとその検索式を永続的に登録し、ユーザから登録されたこれらの検索式を、定期的に(例えば一定時間毎に)評価し、検索式が評価される度に、評価結果をその問い合わせを登録したユーザに送信するシステムであり、ニュース情報通知サービスや、株価情報通知システム等に利用される、さらに、インターネット12に接続された社内ネットワーク等のローカルネットワーク13上に、このXMLデータ連続問い合わせシステム11を利用するユーザ14が複数存在するものとする。

20

【0114】

このような環境において、当実施形態では、このローカルネットワーク13上に、プロクシーサーバ15を置き、このローカルネット13上の各ユーザ14は、直接、XMLデータ連続問い合わせシステム11に自分の検索式を登録する代わりに、このプロクシーサーバ15に対して検索式を登録する。そして、プロクシーサーバ15は、複数のユーザ14から登録された複数の検索式をもとに、第五の実施形態と同様の方法で新たな検索式を合成し、この合成した検索式をXMLデータ連続問い合わせシステム11に登録する。そして、XMLデータ連続問い合わせシステムから定期的に結果を受け取る度に、その結果を用いて、各ユーザ14が登録した検索式の解を合成し、これを各ユーザ14に送信する。

30

【0115】

このようにすると、複数のユーザが、解が一部重複するような多数の検索式を登録した場合には、これらの検索式はプロクシーサーバ15によって、解に重複を含まない問い合わせに置き換えられて連続問い合わせシステム11に登録され、その結果がインターネット12を介してプロクシーサーバ15に送信され、プロクシーサーバ15において一部重複を含む当初の検索式の解が合成されて、これがローカルネット13を介して各ユーザ14に送信されるので、各ユーザ14が直接、それらの検索式を連続問い合わせシステム11に登録する場合に比べ、インターネット12上での重複するデータの送信を排除し、インターネット12上の通信コストを軽減することができる。

40

【図面の簡単な説明】

【0116】

【図1】第五の実施形態におけるシステムの全体構成図

【図2】第一、第三の実施形態におけるシステムの全体構成図

【図3】三つの解集合A1、A2、A3の間の関係を表わすベン図

【図4】第二の実施形態におけるシステムの全体構成図

50

【図5】XMLデータの例（その1）

【図 6】 / a / b , / a / c / * , / a / * / d に対する解集合

【図7】図6の解集合から重複を取り除いた状態のデータ

【図 8】 / a / b // * に対する解集合

【図9】第四の実施形態におけるシステムの全体構成図

【図 10】XMLデータの例（その2）

【図 1.1】 // a, // b // * に対する解集合

【図12】V11, V15に対する解集

【図 1.3】 XML データの例（その 3）

【図14】// a /* / * に対する解集合

【図15】V23, V24に対する解集合

【図16】第六の実施形態におけるシステム環境全体の構成図

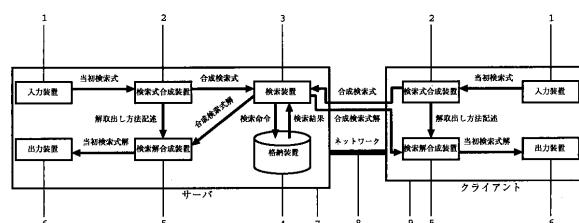
【符合の説明】

[0 1 1 7]

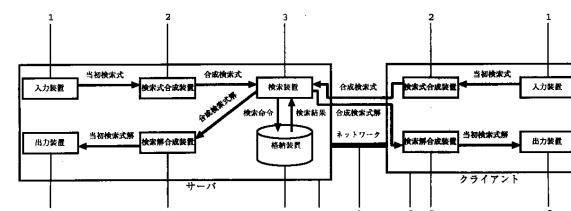
1 入力装置 , 2 検索式合成装置 , 3 検索装置 , 4 格納装置 ,
5 検索解合成装置 , 6 出力装置 , 7 サーバ , 8 ネットワーク , 9
クライアント , 10 重複除去装置 ,
11 XML データ連続問い合わせシステム , 12 インターネット ,
13 口 - カルネットワーク , 14 ユーザ , 15 プロクシ - サーバ

10

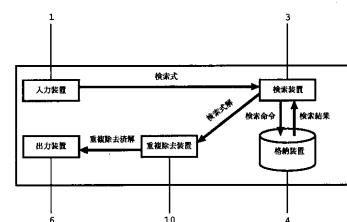
〔図 1 〕



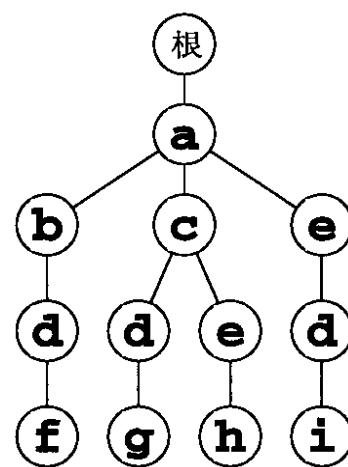
〔 図 4 〕



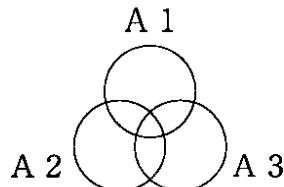
〔 四 2 〕



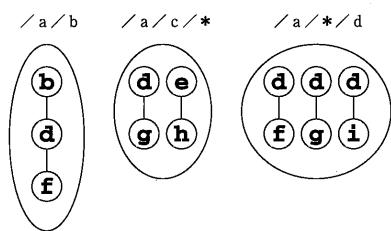
〔 5 〕



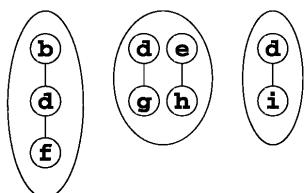
〔 四 3 〕



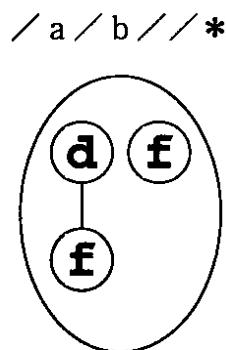
【図6】



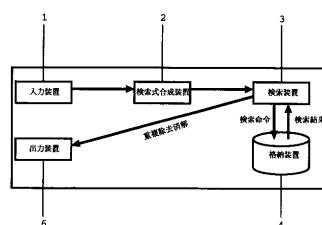
【図7】



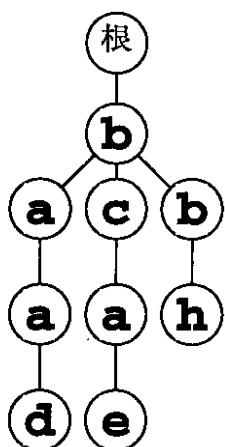
【図8】



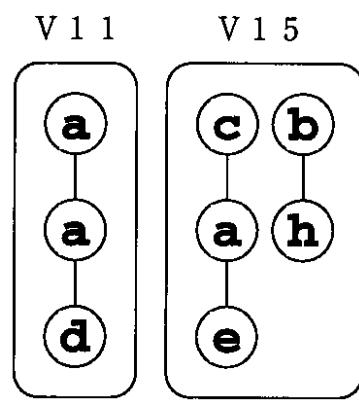
【図9】



【図10】

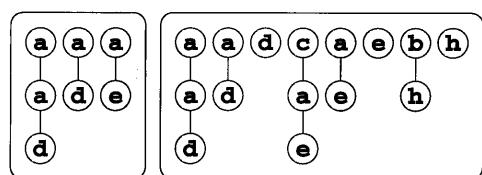


【図12】

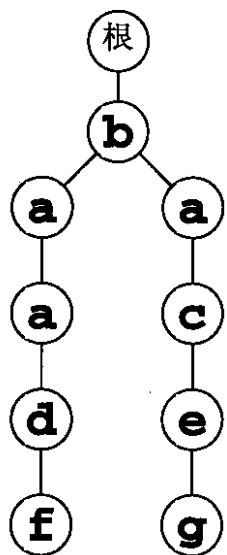


【図11】

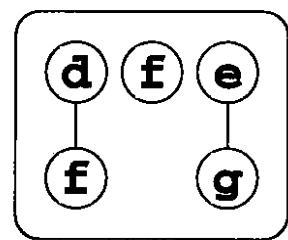
//a
/b///*



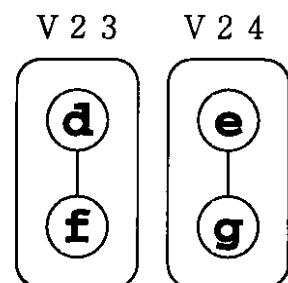
【図13】



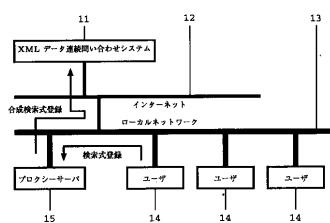
【図14】



【図15】



【図16】



フロントページの続き

(56)参考文献 特開2002-288214(JP,A)
特開平09-006791(JP,A)
特開平09-006805(JP,A)
特開平05-061919(JP,A)
特開平06-325093(JP,A)
特開2001-34618(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 17 / 30