(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3709305号
(P3709305)
(45)発行日　平成17年10月26日(2005.10.26)　(24)登録日　平成17年8月12日(2005.8.12)

(51)Int.Cl.[7]　F I
G06F 17/22　G06F 17/22 520N
G06F 17/21　G06F 17/22 520R
G06F 17/21 550K

請求項の数 5　(全 25 頁)

(21)出願番号　特願平11-187753
(22)出願日　平成11年7月1日(1999.7.1)
(65)公開番号　特開2001-14311(P2001-14311A)
(43)公開日　平成13年1月19日(2001.1.19)
審査請求日　平成13年10月15日(2001.10.15)

(73)特許権者　504373093
日立オムロンターミナルソリューションズ株式会社
東京都品川区大崎一丁目６番３号
(74)代理人　100093492
弁理士　鈴木　市郎
(72)発明者　古賀　昌史
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内
(72)発明者　古川　直広
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内
(72)発明者　池田　尚司
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内
最終頁に続く

(54)【発明の名称】地名文字列照合方法、地名文字列照合装置、地名文字列認識装置及び郵便物区分システム

(57)【特許請求の範囲】

【請求項１】

文字列の入力を受ける入力インタフェースと、複数の地名文字列を格納するメモリと、演算処理装置と、出力インタフェースとを有する地名認識装置における地名文字列照合方法において、

前記メモリには、地名文字列の一部または全部を構成する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより表された複数の地名文字列をもとに生成され、該複数の地名文字列における文字の連接関係を有向グラフを用いて表す地名表記ネットワークが予め格納されており、

前記演算処理装置は、前記入力インタフェースから入力される入力文字列の部分文字列が、前記メモリに地名表記ネットワークとして予め格納された複数の地名文字列のうちの１つと一致するか否かを判断することにより、入力文字列の中から地名を照合し、前記出力インタフェースからその照合結果を出力することを特徴とする地名文字列照合方法。

【請求項２】

地名文字列の一部または全部を構成する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより表された地名文字列をもとに生成され、該複数の地名文字列における文字の連接関係を有向グラフを用いて表す地名表記ネットワークを記憶する記憶手段と、文字列を入力する入力手段と、前記入力された文字列が前記記憶手段に地名表記ネットワークとして記憶した地名文字列であるか否かを照合する手段と、照合の結果を出力する手段と備えたことを特徴と

する地名文字列照合装置。
【請求項３】
文書の表面の濃淡を電気信号に変換して得られた画像を入力として、文書上に記載されていた文字を読み取る文字読取り手段と、請求項２記載の地名文字列照合装置とを備え、前記地名文字列照合装置内の入力手段が前記文字読取り手段からの文字列を入力することを特徴とする地名文字列認識装置。
【請求項４】
請求項３記載の地名文字列認識装置を使用し、郵便物の宛名の中の地名の文字列を認識し、郵便物を区分し、または、認識結果を郵便物に印刷することを特徴とする郵便物区分システム。
【請求項５】
文書の表面の濃淡を電気信号に変換して得られた画像を入力とし、文書上に記載されていた文字を読み取り、地名文字列を認識する地名文字列認識装置において、
地域を表す地名が、異なる文字列であるが同一の地域を意味する単語の配列により表現される複数の異表記を持つ地名文字列の集合を表現し、地名文字列の一部または全部を構成する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより地名文字列を表す地名表現方法を使用して地名文字列を記憶する手段と、
前記入力画像中の部分画像の配列であって、それぞれの部分画像が前記地名文字列表現により表わされた地名文字列の１つに含まれる各文字と類似するものを見い出す照合を行い、該照合結果を前記部分画像の配列の選択にフィードバックすることにより地名文字列を認識する手段とを備えたことを特徴とする地名文字列認識装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、地名文字列照合方法、文字列照合装置、文字列認識装置及び郵便物区分システムに係り、特に、文書上に記載された地名等の文字列を読取る装置に使用される地名文字列記憶手段及び照合手段に適用して好適な地名文字列照合方法、文字列照合装置、文字列認識装置及び郵便物区分システムに関する。
【０００２】
【従来の技術】
一般に、都道府県名、市町村名、字名等の地名単語の並びからなる文字列（以下、地名文字列という）を画像中から読み取る文字認識装置は、
（１）文字パターンを切り出す（文字切出し）、
（２）各々の文字パターンの字種（文字コード）を識別する（文字識別）、
（３）文字の識別結果を予め記憶した地名単語の列と照合する（文字列照合）、
の３つの機能を備えて構成されている。
【０００３】
文字列照合の方法に関する従来技術として、例えば、丸川等による方式（情報処理学会論文誌第３５巻第６号「手書き漢字住所認識のためのエラー修正アルゴリズム」）等が知られている。また、文字切出し、認識、照合を一体化した方式に関する従来技術として、隠れマルコフモデルに基づく方式（O. E. Agazzi、 et al., "Connected And Degraded Text Recognition using Planar Hidden Markov Models、" Proceedings of International Conference on Acoustics、 Speech、 and Signal Processing ）、探索的に文字列を認識する方法（Koga、 et al.,“Lexical Search Approach for Character-String Recognition” Third International Association for Pattern Recognition Workshop on Document Analysis Systems 1998）が知られている。
【０００４】
前述した従来技術は、文字列照合処理のために、予め出現し得る地名文字列を記憶する手段、地名文字列辞書が必要である。そして、地名文字列辞書としては、以下示すような３

種類のものがある。
（１）ファイルに格納された「辞書ソースファイル」
これは、後述する「地名表記規則ファイル」等であり、新規作成や修正のために、編集が可能でなくてはならない。
（２）メモリ上に格納された「辞書テーブル」
これは、後述する「地名表記ネットワーク」等であり、辞書ファイルの内容を、照合処理に適した形式でメモリ上に展開したものである。
（３）前述の（１）と（２）との中間段階の「辞書バイナリファイル」
これは、メモリ上への展開を容易にするため、予め展開処理の一部を施した結果をファイルに格納したものである。

【０００５】
従来技術に使用される辞書ソースファイルの形式は明らかにされていない場合が多い。しかし、従来技術は、いずれも、出現し得る地名文字列を予めもれなく辞書テーブルに記憶することを前提にしており、このため、出現し得る地名文字列を、予めもれなく列挙したテキストファイルが、辞書ソースファイルとして用いられていると考えられる。

【０００６】
【発明が解決しようとする課題】
前述した従来技術は、文字列照合処理のために辞書を用意する必要があるが、日本語には、同一の地域を異なる文字列で表現する異表記が多く、出現し得る地名文字列を辞書にもれなく登録することが困難であり、このための完全な辞書を人手で作成することが事実上不可能であるという問題点を有している。

【０００７】
日本語の地名の異表記には、使用する文字の違いによる異表記、単語の省略による異表記、付加的な文字列による異表記、通り名の表記による異表記等がある。以下、これらの異表記の例について説明する。

【０００８】
（１）　使用する文字の違いによる異表記
「小沢」と「小澤」、「市ヶ谷」と「市ケ谷」と「市が谷」等がある。

【０００９】
（２）　単語の省略による異表記
都道府県名を省略する異表記、「大字」、「字」を省略する異表記がある。
都道府県名を省略する異表記は、郵便物の宛名等の場合に多く見られ、例えば、「埼玉県川越市大字小ヶ谷」と「川越市大字小ヶ谷」等がある。また、「大字」、「字」を省略する例として、例えば、「埼玉県川越市大字小ヶ谷」と「埼玉県川越市小ヶ谷」等がある。

【００１０】
（３）付加的な文字列による異表記
小字名等の本来住所の特定には不要である文字列が付加され異表記であり、例えば、「埼玉県川越市大字小ヶ谷」と「埼玉県川越市大字小ヶ谷字東関」等がある。

【００１１】
（４）通り名の表記による異表記
京都などで多く見られるもので、例えば、「京都市下京区大政所町」と「烏丸通仏光寺下る」等がある。

【００１２】
前述したように、地名の異表記には各種のものがあるが、例えば、「埼玉県川越市小ヶ谷」という地名を例にして、これに対応する異表記を調べて見ると、次に列挙するように、極めて多数の異表記があることが判る。
「埼玉県川越市小ヶ谷」
「埼玉県川越市小ケ谷」
「埼玉県川越市小が谷」
「埼玉県川越市大字小ヶ谷」

「埼玉県川越市大字小ケ谷」
「埼玉県川越市大字小が谷」
「川越市小ヶ谷」
「川越市小ケ谷」
「川越市小が谷」
「川越市大字小ヶ谷」
「川越市大字小ケ谷」
「川越市大字小が谷」

【００１３】

前述の例では、さらに「埼玉県川越市小ヶ谷東田」、「埼玉県川越市小ヶ谷東関」、「埼玉県川越市小ヶ谷西関」等、小字名が併せて用いられる場合があり、前述で列記した１２の異表記との組み合わせを考慮すると、合計８４通りの異表記が存在することになる。

【００１４】

従来技術の場合、前述したような多様な異表記の全ての組み合わせを、網羅的に人手で辞書ファイルに登録する必要があり、このため、辞書ファイル作成には多くの人手がかかるという問題があった。しかも、異表記が特に多い京都市等の場合、市内の町名と通り名との呼び方の合計が数十万通りにものぼり、完全な辞書を人手で作成することは事実上不可能であった。

【００１５】

本発明の第１の目的は、前述した問題点を解決し、文字列照合用辞書に多様な異表記をもれなく登録することを容易にすることのできる地名表記方法を提供することにある。

【００１６】

前述したように、地名の表記に異表記が多い場合、仮に異表記をもれなく辞書に記載することができたとしても、従来技術のものでは、辞書の記憶容量が大きくなり、処理時間も異表記の数に応じてに大きくなってしまうという問題点を生じることになる。

【００１７】

前述の問題点を解決することができる技術として、トライ（Trie）と呼ばれるデータ形式により、辞書の記憶容量を小さくし、さらに照合処理を高速にすることができるようにした技術が、（Koga、 et al.,“Lexical Search Approach for Character-String Recognition” Third International Association for Pattern Recognition Workshop on Document Analysis Systems 1998）等に記載されて知られている。この技術は、表記に多様さがある部分のみ分岐するような、木形式のデータとして地名を表記するものであり、文字列の集合からTrieを自動的に生成することを容易にしたものである。

【００１８】

前述の技術は、例えば、「埼玉県川越市小ヶ谷東田」、「埼玉県川越市小ヶ谷東関」、「埼玉県川越市小ヶ谷西関」の３つの表記から、以下のようなTrieを容易に生成することができる。

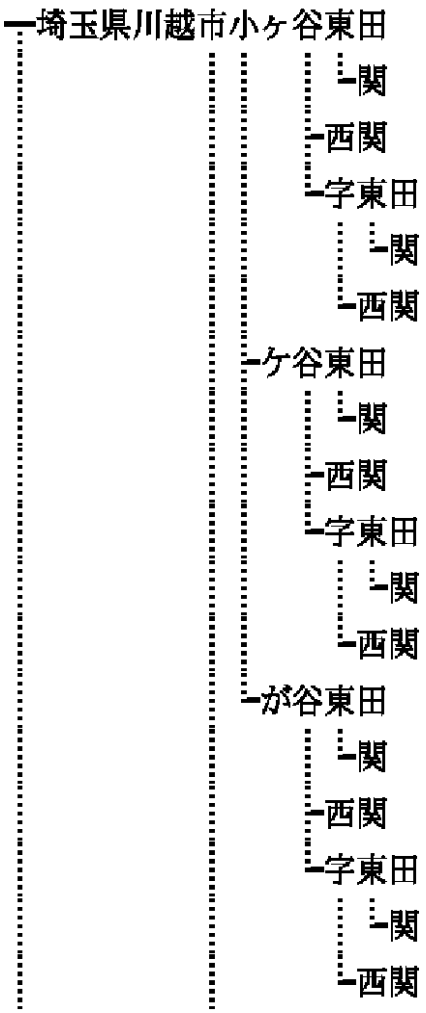
埼玉県川越市小ヶ谷東田
　　┣関
　　┗西関

以下、このように、地名文字列における文字の連接関係をネットワークで表したものを地名表記ネットワークと呼ぶこととする。

【００１９】

しかし、このようなTrie型の地名表記ネットワークは、文字列の一部分に相違がある場合、これらを全く別の文字列として扱い、別の枝を生成せざるを得ないことになり、このため、例えば、「埼玉県川越市小ヶ谷」の異表記群に対応するTrieは、次に示すような大きなものになってしまう。

【００２０】



（以下略）

【００２１】

前述したように、Trie型の地名表記ネットワークを用いる手法で異表記を表現しようとしても、辞書容量、処理時間ともに大幅に増大してしまうという問題点を生じてしまう。

【００２２】

従って、本発明の目的は、多様な異表記を認識するのための地名辞書に使用するための、記憶容量が小さくかつ高速に照合処理ができるような記憶形式を持つ地名文字列照合方法、文字列照合装置、文字列認識装置及び郵便物区分システムを提供することにある。

【００２３】

【課題を解決するための手段】

本発明によれば前記目的は、文字列の入力を受ける入力インタフェースと、複数の地名文字列を格納するメモリと、演算処理装置と、出力インタフェースとを有する地名認識装置における地名文字列照合方法において、前記メモリには、地名文字列の一部または全部を構成する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより表された複数の地名文字列をもとに生成され、該複数の地名文字列における文字の連接関係を有向グラフを用いて表す地名表記ネットワークが予め格納されており、前記演算処理装置は、前記入力インタフェースから入力される入力文字列の部分文字列が、前記メモリに地名表記ネットワークとして予め格納された複数の地名文字列のうちの１つと一致するか否かを判断することにより、入力文字列の中から地名を照合し、前記出力インタフェースからその照合結果を出力することにより達成される。

【００２４】

また、前記目的は、地名文字列照合装置において、地名文字列の一部または全部を構成

する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより表された地名文字列をもとに生成され、該複数の地名文字列における文字の連接関係を有向グラフを用いて表す地名表記ネットワークを記憶する記憶手段と、文字列を入力する入力手段と、前記入力された文字列が前記記憶手段に地名表記ネットワークとして記憶した地名文字列であるか否かを照合する手段と、照合の結果を出力する手段と備えたことにより達成される。

【００２５】

また、前記目的は、地名文字列認識装置において、文書の表面の濃淡を電気信号に変換して得られた画像を入力として、文書上に記載されていた文字を読み取る文字読取り手段と、前述の地名文字列照合装置とを備え、前記地名文字列照合手段内の入力手段が前記文字読取り手段からの文字列を入力することにより達成される。

【００２６】

また、前記目的は、郵便物区分システムにおいて、前述の地名文字列認識装置を使用し、郵便物の宛名の中の地名の文字列を認識し、郵便物を区分し、または、認識結果を郵便物に印刷することにより達成される。

【００２７】

また、前記目的は、文書の表面の濃淡を電気信号に変換して得られた画像を入力とし、文書上に記載されていた文字を読み取り、地名文字列を認識する地名文字列認識装置において、地域を表す地名が、異なる文字列であるが同一の地域を意味する単語の配列により表現される複数の異表記を持つ地名文字列の集合を表現し、地名文字列の一部または全部を構成する部分文字列毎に、文字または構文カテゴリの配列を定義し、文字または定義済みの構文カテゴリの配列からなる構文カテゴリにより地名文字列を表す地名表現方法を使用して地名文字列を記憶する手段と、前記入力画像中の部分画像の配列であって、それぞれの部分画像が前記地名文字列表現により表わされた地名文字列の１つに含まれる各文字と類似するものを見い出す照合を行い、該照合結果を前記部分画像の配列の選択にフィードバックすることにより地名文字列を認識する手段とを備えたことにより達成される。

【００２８】

具体的に言えば、本発明は、前記目的を達成するため、地名の異表記を文脈自由文法の生成規則を用いて表現する。文脈自由文法は、ある文の要素（構文カテゴリ）がどのような他の構文カテゴリの列に置換されるかを、生成規則により表わす(「自然言語処理入門」近代科学社、ISBN4-7649-0143-9）。本発明は、生成規則の表現法の１つとして知られるＢＮＦ記法（Backus-Naur-Form）（中田「コンパイラ」ISBN4-7828-5057-3)を、地名の表現に適するよう拡張した拡張ＢＮＦ記法を用いる。

【００２９】

前述の生成規則により、典型的な異表記のパタン、例えば、「ヶ」、「ケ」、「が」を１つの構文カテゴリとして定義することができ、地名の異表記の集合を簡潔に表現できる。さらに、ＢＮＦ記法で採用されている選択記号を用いることにより、地名の異表記をより簡潔に表現することが可能となる。このため、本発明によれば、多様な異表記の集合をもれなく記載した辞書を容易に作成することができる。

【００３０】

ＢＮＦ記法は、文脈自由文法の生成規則を、置換、オプション、選択等の記号を用いて表現する記法であり、以下のような記号を用いる。

::＝　　置換。左辺の構文カテゴリを右辺の構文カテゴリまたは文字の配列で置換できることを意味する。

［］　　オプション。［］内の記述があってもなくてもよいことを意味する。

｜　　選択。右辺、左辺のいづれかを意味する。

【００３１】

一例として、前述の「埼玉県川越市小ヶ谷」の異表記の生成規則をＢＮＦ記法で表現した例を以下に示す。

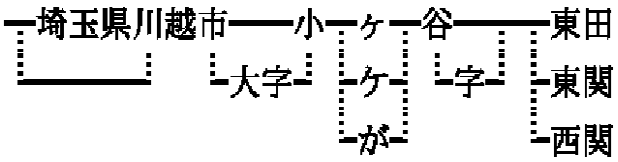
＜ｗヶ＞　::＝ヶ｜ケ｜が

＜埼玉県川越市小ヶ谷＞::＝［埼玉県］川越市［大字］小＜ｗヶ＞谷［［字］東田｜東関｜西関］

【００３２】

また、前述のような表記形式を用いることにより、地名表記ネットワークを小形化することが可能となる。前述の表記形式では、部分文字列の相違は記号「［］」 や「｜」を用いて陽に表現されている。このため、部分文字列の相違が異表記にありえる場合、その部分をバイパスするような経路をネットワーク上に容易に設定することができる。例えば、前述のＢＮＦ記法の表記は、下に示すようなコンパクトなネットワークに置き換えることができる。従来のような文字列の羅列からこのようなコンパクトなネットワークを生成することは、困難であった。

【００３３】



【００３４】

【発明の実施の形態】

以下、本発明による地名表記方法及び地名文字列認識方法の実施形態を図面により詳細に説明する。

【００３５】

図１は本発明の実施形態による地名文字列認識の処理例を説明するフローチャートであり、まず、このフローについて説明する。なお、以下の説明において使用するフローチャートは、ゲーン・サーソン記法に従って表現した。この記法に関しては、「Ｊ。マーチン「ソフトウエア構造化技法」近代科学社」、「ISBN4 - 7649 - 0124 - 2 C3050 P5562E」に記載されている。

【００３６】

（１）まず、地名の認識に先立ち、地名文字列生成規則編集処理（ステップ１０１）が、地名の異表記の事例に基づき地名文字列の生成規則を作成し、この生成規則を地名文字列生成規則ファイル１０２に格納する。ステップ１０１の地名文字列生成規則編集処理は、計算機を介した人間の編集作業により実現することができる。

【００３７】

（２）次に、地名表記ネットワーク生成処理（ステップ１０３）が、地名文字列生成規則ファイル１０２を読み込み、地名認識１０４のための辞書である地名表記ネットワークを生成する。ステップ１０３の地名表記ネットワーク生成処理は、計算機上のプログラムとして実現することができる。

【００３８】

（３）次に、地名認識処理（ステップ１０４）が、地名表記ネットワークを参照し、入力画像中から地名文字列を読み取る。ステップ１０４の地名認識処理１０４は、計算機上のプログラムとして実現することができる。

【００３９】

地名文字列生成規則ファイル１０２は、本発明による「拡張ＢＮＦ記法」を用い、地名の異表記群を文脈自由文法の生成規則により表現する。拡張ＢＮＦ記法は、ＢＮＦ記法に結合等の記号を拡張したものであり、以下に説明するような記号を用いられる。

::＝　　置換。左辺の構文カテゴリを右辺の構文カテゴリ、または、文字の配列で置換できることを意味する。

［］　　オプション。［］内の記述を省略してもよい事を意味する。

｜　　選択。この記号の右、左のいづれかを意味する。

（）　　結合。前後の変数より先に括弧内を評価する。

＜W文字列＞　構文カテゴリ。

＜Ｎ数字＞　　特定の地域を示す地名文字列の異表記群を表す構文カテゴリ。数字は、地名の識別子。０より大きい整数を用いる。

【００４０】

そして、前述した記号は、以下の優先順位により評価される。

（１）＜Ｗ文字列＞及び＜Ｎ数字＞の変数名の定義

（２）［ ］及び（ ）のかっこ類。２重以上の入れ子でかっこ類を用いる場合、内側のかっこを優先して評価

（３）　|

（４）::＝

【００４１】

図２は前述のステップ１０１による編集処理で編集された地名文字列生成規則により表現された地名の表記例と生成規則を用いずに異表記を羅列した例とを示す図である。

【００４２】

図２（Ａ）に示す地名文字列生成規則により表現された地名の表記例は、その例として、「埼玉県川越市大字小ヶ谷」（「東田」「東関」「西関」が小字）、「埼玉県川越市大字笠幡」（「久保」「河南」が小字）、「埼玉県川越市下広谷」の異表記を本発明による拡張ＢＮＦ記法で表記した例である。このように、多数の異表記を含む地名を、本発明により導入した記号を用いることにより、極めて簡単に表現することができる。これに対して、図２（Ｂ）に示す生成規則を用いずに異表記を羅列した例は、多数の異表記を羅列するだけであるので、図２（Ａ）に示す４行の表記から生成される異表記の数は１０６通りにもなる。図２（Ｂ）に示しているのはその一部である。

【００４３】

地名文字列生成規則ファイル１０２は、通常のテキストファイルであり、地名文字列生成規則編集処理のステップ１０１の実現手段としては一般的なテキストエディタを適用することが可能である。

【００４４】

図３は図２（Ａ）の生成規則の例から作られる地名表記ネットワークを模式的に示す図であり、以下、これについて説明する。

【００４５】

地名表記ネットワークは、各辺が部分文字列に、各頂点が部分文字列の境界に対応する有向グラフである。各辺の方向は、文字列中の文字の順に一致する。図３において、NULLと記された辺は、NULL遷移、すなわち、その箇所に何も文字列がなくてよいことを示している。図３の中の右下に線の入った円３０１は、地名文字列の開始位置を示す。また、中央に斜線が入った円３０２～３０４は、地名文字列の終わりの位置を示す。

【００４６】

図４は地名表記ネットワークを計算機上に実装する際のデータ形式を説明する図であり、以下、これについて説明する。そして、計算機上に地名表記ネットワークを実装する際、地名表記ネットワークは、図４に示すようなデータ形式（left-child. right-sibling representation、Ｔ．コルメン他「アルゴリズムイントロダクション」近代科学社、pp. 201-202)を用いて表現される。このデータ形式は、文字の連接関係を子ポインタで、地名表記ネットワークの分岐を兄弟ポインタで表現するものである。

【００４７】

図４（Ａ）は、各データレコードの構成要素を示しており、各データレコードは、データ項目ｃ４０１、ｂ４０２、ｄ４０３の３つのデータ項目からなる。データ項目ｃは文字コードであり、データ項目ｂは兄弟ポインタである。また、データ項目ｄは子ポインタである。そして、あるデータレコードからの分岐は、兄弟ポインタにより、また、文字列は、子ポインタによって接続されたリスト形式で表現される。例えば、図３に示す地名表記ネットワークを前述したデータレコードによりリスト形式で表現すると、図４（Ｂ）に示すようなものとなる。

【００４８】

図４（Ｂ）に示すリスト形式で表現した地名表記ネットワークにおいて、データレコード４０４’（文字コード「小」に対応）からは、データレコード４０４～４０６に分岐するが、データレコード４０４’から４０４には子ポインタにより連結され、データレコード４０４、４０５、４０６は兄弟ポインタにより連結されている。また、文字列「埼玉県」は、子ポインタで接続されたデータレコード４０７、４０８、４０９で表されている。また、データレコードがNULL遷移に対応する場合、そのデータレコードの文字コードｃ４０１にはNULL文字が格納され、このNULL文字が格納されたデータコードから分岐するデータレコードは、省略されてもよいことが意味される。さらに、地名文字列の最後の文字に対応するデータレコードの次に、データレコード４１０として示すように、１つ余分のデータレコードが設けられ、このデータレコード４１０の子ポインタｄには、NULLポインタが格納されて、ネットワークの終端であることを表すと共に、兄弟ポインタｂに地名の識別子が格納される。

【００４９】

前述のような形式表現される図４（Ｂ）に示すリスト形式の地名表記ネットワークは、各データレコードがノードに対応するグラフとみなすことができ、図３に模式的に表した地名表記ネットワーク中の各辺が、ここでは文字数分のノードで表わされたことになる。

【００５０】

図５は図１のステップ１０３における地名文字列生成規則から地名表記ネットワークを生成する処理を説明するフローチャート、図６は生成される構文木の例を説明する図であり、以下、これらについて説明する。

【００５１】

まず、地名表記生成規則ファイル１０２の中の各地名文字列の生成規則、例えば、図２（Ａ）の上から２行目以降の＜Ｎ数字＞で始まる各行について、制御ループ５０１により１つづつ処理する。各行に対し、まず、ステップ５０２で、行内の文字列の構文解析を行い、図６に示すような構文木を作成する。次に、ステップ５０３で、その地名異表記群に対応する地名表記ネットワークの終端ノードt_iを生成する。以下、特にことわりがない場合、「地名表記ネットワーク上のノード」とは図４（Ａ）の形式のデータレコードを示すものとする。t_iの中の文字コードｃにはNULLを、子ポインタｄにはNULLを、兄弟ポインタｂにはその地名異表記群の地名識別子を格納する。次に、ステップ５０４で、後述する関数procを使用してその地名異表記群に対応する地名表記ネットワークを生成する。全ての地名文字列生成規則を処理し終わった後、ステップ５０５で、生成された地名表記ネットワークの冗長な部分を統合する。

【００５２】

地名文字列生成規則から構文木を生成する処理は、例えば、「自然言語処理入門」（近代科学社、ISBN4-7649-0143-9、pp. 19-30）に記載されているような生成規則に応じた遷移ネットワークを生成する手法等を使用することができる。図６に示すステップ５０２の処理で生成された構文木の例は、図２（Ａ）の２行目から生成される構文木の例である。この図６において、「＋」が記された円は文字列の連接を、「［］」が記された円はオプションを、「｜」が記された円は選択を表わし、四角は文字列を表わしている。拡張ＢＮＦ記法は、括弧「（」、「）」も使用されるが、本発明の実施形態に使用する構文木は、括弧に対応するノードは設けず、括弧により定まる演算の順序を、構文木の構造自身に反映させたものとする。

【００５３】

関数procは、構文木から地名表記ネットワークを生成するために使用する関数であり、ｐとａとの２つの引き数をとる。引き数ｐは、生成する地名表記ネットワークの終端のノードの子ポインタｄがとる値を指定する。また、引き数ａは、処理対象の構文木の最上位ノードを示す。あるノードに引き数ａが指定されると、引き数ａ以下の全てのノードが再帰的に処理される。

【００５４】

図７は関数procによる処理動作を説明するフローチャート、図８、図９は関数procによっ

て地名表記ネットワークが生成される過程を説明する図、図１０は地名表記生成規則から生成された地名表記ネットワーク群を示す図であり、以下、これらについて説明する。なお、図７において、図に示されているｐ、ｑ、ｒは、図４に示す形式のデータレコードのアドレスを表す変数であり、記号「 －＞ 」は、データレコードの中のデータ項目を表わしている。また、図７に示すフローの処理は、構文木のノードａの種類に応じて４つに場合分けして実行される。

【００５５】

（１）構文木のノードａの種類を判別し、種類が、「＋」、「｜」、「［ ］」、「文字列」の何れてあるかを判定する（ステップ７０１）。

【００５６】

（２）ステップ７０１での判定で、構文木のノードａの種類が「＋」すなわち結合であった場合、まず、変数ｑに引き数ｐをコピーする。すなわち、この処理で生成する部分ネットワークの終端ノードのアドレスをコピーする（ステップ７０２）。

【００５７】

（３）次に、構文木の子ノードn_i （１≦ｉ≦子ノードの数）を右から順に関数proc（ ）により処理して地名表記ネットワークの部分ネットワークを生成する。その際、関数proc（ ）で生成する部分ネットワークの終点がｑとなるよう引き数を渡す。この結果生成された部分ネットワークの始点のポインタをｑへ代入し直し、次に生成する部分ネットワークの終点とする。このようにして関数proc( )を繰り返し呼び出すことにより、構文木の「＋」の子ノードから生成される地名表記ネットワークの部分ネットワークが次々と連結される（ステップ７０３、７０４）。

【００５８】

（４）全ての子ノードを処理し終わったなら、その時点でのｑすなわち部分ネットワークの先頭を戻り値として返す（ステップ７０５）。

【００５９】

（５）ステップ７０１での判定で、構文木のノードａの種類が「｜」すなわち選択であった場合、まず、子ノードの１つn_1 から部分ネットワークを生成し、得られた部分ネットワークの先頭アドレスを変数ｑに代入する（ステップ７０６）。

【００６０】

（６）次に、変数ｒにｑの値を代入し、他の子ノードn_i （２≦ｉ≦子ノードの数）から生成する部分ネットワークを順に生成する。生成した部分ネットワークの先頭アドレスは、ｒの兄弟ポインタｂに格納する。さらに、生成した部分ネットワークの先頭アドレスをｒに代入し、以下同様の処理を繰り返す（ステップ７０７～７１０）。

【００６１】

（７）全ての子ノードを処理し終わったなら、ｑすなわち一番始めに生成した部分ネットワークの先頭のアドレスｑを戻り値として返す（ステップ７１１）。

【００６２】

（８）ステップ７０１での判定で、構文木のノードａの種類が「［ ］」すなわちオプションであった場合、まず、構文木の子ノードに対応する部分ネットワークを生成し、その先頭アドレスを変数ｑに格納する。その際、生成した部分ネットワークの末端はｐとなるようパラメータを指定する（ステップ７１２）。

【００６３】

（９）次に、NULL遷移に対応するノードを関数newNd（ ）を用いて生成し、そのアドレスをｑの兄弟ポインタｂに格納する。なお、newNd（ ）は、図４に示す形式のデータレコードの記憶領域を新たに１つ確保する関数であり、確保されたデータレコードのデータ項目ｂにはNULLポインタがセットされる（ステップ７１３）。

【００６４】

（10）次に、NULL遷移に対応するノードの文字コードｃにNULLを代入し、さらに、NULL遷移に対応するノードの子ノードポインタｄにｐをセットする（ステップ７１４、７１５）。

【００６５】
（11）最後に生成した部分ネットワークの先頭のアドレスｑを戻り値として返す（ステップ７１６）。
【００６６】
（12）ステップ７０１での判定で、構文木のノードａの種類が文字列であった場合、まず、変数ｑにｐの値を代入する（ステップ７１７）。
【００６７】
（13）次に、下記の処理を文字列中の各文字C_i（1≦i≦文字列長）に対して、文字列の終わりから順に繰り返し、各文字に対応するノードを１つづつ生成する。ここでは、まず、関数newNd（ ）でノードの記憶領域を１つ分確保する。次に、新たに生成したノードの文字コードｃに、C_iを代入する。次に、新たに生成したノードの子ノードｄにｑの値を代入する。次に、ｑの値を新たに生成したノードのアドレスで置き換える（ステップ７１８～７２２）。
【００６８】
（14）前述の処理を各文字C_iに対して実行した後、新たに生成した部分ネットワークのアドレスｑを戻り値として返す（ステップ７２３）。
【００６９】
関数procによって地名表記ネットワークが生成される過程を示す図８、図９において、８０１は、図５に示すフローのステップ５０３の処理で終端ノードを生成し、識別子「3501104」を格納したところである。その後、図７に示すフローの各ステップの処理により、図８、図９に示す図の上から下に順に示すように地名表記ネットワークが生成されていく。そして、図６に示す構文木のノード６０３を関数procで処理した場合、まず、ノード６０２に対応する部分ネットワークが関数procで生成され、８０２に示すようにノード６０２に対応する部分ネットワークが生成される。次に、ノード６０４に対応する部分ネットワークを関数procで生成する。この場合、ｐはノード８０４のアドレスを格納しており、生成された部分ネットワークは、８０３に示すようにノード８０４に接続される。
【００７０】
図５に示すフローの制御ループ５０１により、各地名文字列の生成規則毎に別個の地名表記ネットワークが生成される。この結果、図２の地名表記生成規則から生成された地名表記ネットワーク群は、図１０に示すようなものとして生成され、さらに、ステップ５０５の処理により、これらのネットワーク群の冗長な部分、例えば、埼玉県川越市の部分を統合し、図３により説明したような地名表記ネットワークが生成される。
【００７１】
図１１は従来技術を用いて地名表記ネットワークを生成する処理手順を説明するフローチャート、図１２は従来技術により生成される地名表記ネットワークの生成過程の例を説明する図、図１３は従来技術により生成された地名表記ネットワークの例を示す図であり、以下、これらの図を参照して、生成規則を用いない場合の地名表記ネットワーク生成方法について説明する。
【００７２】
ここで従来技術を説明する理由は、従来の地名文字列の表記方法からはTrieと呼ばれる木構造の地名表記ネットワークしか生成することができず、本発明の表記方法から生成される地名表記ネットワークが記憶容量、照合に要する処理時間共に優れていることを示すためである。従来技術による地名の表記を表現する手法は、図２（Ｂ）に示すような地名文字列の羅列であり、図１１により説明するフローは、このような単語の羅列から地名表記ネットワークを生成する手順である。ここで、図２（Ｂ）の中のｋ番目の文字列をS_k、その長さをL_k、各文字列のｉ番目の文字をC_iとする。また、各文字列に対応する識別子が別途記憶されているものとする。そして、生成する地名表記ネットワークは、図４に示すデータ形式で実現する。
【００７３】
（１）まず、地名表記ネットワークの仮の根となるノードｒｒを生成する。このノードの

子ノードポインタｄにはNULLをセットする（ステップ１１０１、１１０２）。
【００７４】
（２）ループ１１０３により全ての文字列S_kを１つづつ処理する。
【００７５】
（３）まず、変数ｐに根のアドレスを代入する。次に文字列中の文字の１つ毎に、サブルーチンSrchNxt を呼び出す。サブルーチンSrchNxt は、各文字に対応するノードがすでに生成されているか否かを判断し、生成されていない場合、新たなノードを追加する処理手順であり、この手順については後述する（ステップ１１０４～１１０６）。
【００７６】
（４）サブルーチンSrchNxtで文字列中の文字を処理し終わったならば、新たな子ノードを関数newNd( )で生成し、その文字列の識別子をポインタｂの領域に格納し、さらにこの新たな子ノードのアドレスをｐの子ノードポインタｄに代入する。ループ１１０３が終了した時点でのｒｒの子ノードが地名表記ネットワークの根となる（ステップ１１０７～１１１０）。
【００７７】
次に、サブルーチンSrchNxtの処理について説明する。
【００７８】
（１）まず、変数ｑにｐの子ノードｄの値を代入し、次に、ループ処理を行って全てのｐの子ノードを変数ｑにより走査し、対応する文字コードすなわちデータ項目ｃがC_iと等しいか否か調べる。もし等しければ、すでにC_iに対応するノードが生成されているとみなして、ポインタｐをそのノードｑに進めて終了する（ステップ１１１１、１１１３～１１１５、ループ１１１２）。
【００７９】
（２）ステップ１１１３でのチェックで、データ項目ｃがC_iと等しくなければ、ｑにｑの兄弟ポインタの値を代入し、ｑがNULLとなるまでループ処理を繰り返す。（ステップ１１１６）。
【００８０】
（３）ループ処理が終了してもC_iに対応するノードが見いだせない場合、新たなノードを関数newNd( )で生成し、新たなノードの文字コードｃにC_iを、子ノードポインタｄにNULLを、兄弟ポインタｂにｐの子ノードポインタｄの値をそれぞれ代入し、この新たなノードのアドレスをｐの子ノードポインタｄへ代入し、ポインタｐに新たな子ノードのアドレスを代入して、このサブルーチンの処理を終了する（ステップ１１１７～１１２２）。
【００８１】
前述した図１１の処理手順に従って地名表記ネットワークが生成される過程を図１２に示している。ここで挙げた例は、図２（Ｂ）の上から３行を処理する過程である。まず、始めに、「川越市小ヶ谷」に対応する地名表記ネットワークが生成される（１２０１）。次に、「川越市笠幡」を処理するが、「川越市」の部分は、１２０１ですでに生成されているので、新たなノードは生成されない。しかし、ポインタｐが１２０２に示す位置に達し、「笠」の文字を処理するときには、「笠」に該当するノードは「市」の子ノードにはない。そこで、「小」の兄弟ノードとして新たに「笠」に該当するノードを生成する。以下、残りの文字「幡」に対応するノードを、新たに生成したノードの子ノードとして連結する（１２０３）。「川越市下広谷」の場合も同様に処理され、「下」に対するノードを「小」、「笠」の兄弟として新たに生成し（１２０４）、以降の文字に対応するノードが連結される（１２０５）。
【００８２】
図１３は図２（Ｂ）に記した異表記群から生成した地名表記ネットワークの一部を模式的に示すものであるが、この例は、図３に示す場合と異なり、従来の表記方法から生成される地名表記ネットワークは、木の形式、すなわち一度分岐したら再び合流することがない形式になっている。これは、Trieとして知られているデータの表現形式である。図３と比較すると、冗長な部分が多いことが判る。例えば、「東田」、「東関」、「西関」に対応

する部分ネットワークが、６回繰り返されている。このことは、必要とする記憶容量の増大につながることを意味し、階層的なメモリ構成をとる計算機の場合、アクセスするメモリ空間が大きくなるとキャッシュのミスヒットなどによりアクセスが遅くなり、ひいては後述する文字列照合処理自身が遅くなることになる。

【００８３】

本発明により、図３に示すような冗長さが少ない地名表記ネットワークを生成することができることは、生成規則による地名単語表記の本質的な利点である。この生成規則を用いると、冗長な箇所を明確に表現することができる。例えば、図２（Ａ）に示す例の場合、「小ヶ谷」の「ヶ」には３通りの異表記があるが、「ヶ」以降の文字列は同じであることが拡張ＢＮＦ記法により示されている。このため、図３に示すように、「小」と「谷」との間のみ、３つの経路があるようなネットワークが生成される。

【００８４】

これに対して、図２（Ｂ）に示すような従来の地名文字列の表記方法は、「ヶ」以降の異表記群が等価かどうかを検知することができず、図１３に示すようなネットワークしか生成することができない。

【００８５】

図１４は図１に示す地名認識処理１０４での処理動作を説明するフローチャートであり、以下、これについて説明する。

【００８６】

（１）まず、入力画像から文字行切出し処理により、文字行の部分の画像を切出す（ステップ１４０１）。

【００８７】

（２）次に、文字切出し処理により、文字行画像中から文字と思われるパタン、すなわち、候補パタンを切出す。この段階で一意に文字の境界を決定できない場合、複数の境界の仮説に基づいて、文字パタンの切出しを試み、それぞれの仮説に対応した候補パタンを出力する（ステップ１４０２）。

【００８８】

（３）次に、文字認識処理により、切出されたそれぞれの候補パタンがどんな文字であるかを認識し、候補文字列として出力する。文字の切出し方が複数の仮説に基づいている場合、また、文字認識の結果、１つのパタンに対し複数の候補文字が出力され場合、文字認識処理は、それぞれの切出し方及び候補文字に組み合わせに対応して複数の候補文字列を出力する（ステップ１４０３）。

【００８９】

（４）最後に、文字列照合処理により、それぞれの候補文字列が正しい地名文字列になっているか否かを、地名表記ネットワークを参照して照合する。照合で受理された候補文字列を地名認識結果とする（ステップ１４０４）。

【００９０】

図１５は前述の文字列照合処理１４０４での処理動作を説明するフローチャートであり、以下、これについて説明する。この処理は、１つの文字列を入力とし、入力文字列の少なくとも一部が地名文字列として受理し得るか否か判定し、受理し得るなら該当するその地名表記の識別子を求める処理である。ここで、入力文字列の長さをＬ、文字列のｉ番目の文字をC_iとする。

【００９１】

（１）まず、ループ１５０１により、照合の起点ｓを１からＬまで変えながら、ステップ１５０２、１５０３を繰り返す。

【００９２】

（２）ノードを指し示す変数ｐに、地名表記ネットワークの根のアドレスをセットする。次に、引き数ｐおよびｓを与えて関数srchを呼び出す。関数srchは、地名表記ネットワーク中から入力文字列に一致する経路を見い出し、その終端のノードのアドレスを返す関数である。srchの戻り値が、NULLポインタでなければ照合に成功したものとみなして、関数

srchの戻り値が示すノードに格納された識別子を出力する（ステップ１５０２～１５０４）。

【００９３】

（３）もしｓがＬに達しても照合が成功しなければ、文字列照合処理は失敗したものとして、処理を終了する（ステップ１５０５）。

【００９４】

前述の処理において、関数srchは、再帰的に自分自身からも呼び出され、地名表記ネットワーク中から入力文字列に一致する経路を深さ優先で探索する。関数srchは、引き数ｐ及びｉの２つの引き数をとる。引き数ｐは、探索を開始するノードを指し示す。また、引き数ｉは、整数であり、現在の処理で注目しているのが入力文字列中の何番目の文字かを表す。受理される文字列が見つかった場合、関数srchは、その文字列の終端のノードのアドレスを返し、受理される文字列が見つからなかった場合、NULLポインタを返す。

【００９５】

図１６は前述の処理での関数srchの処理動作を説明フローチャートであり、以下、これについて説明する。

【００９６】

（１）まず、引き数ｐが文字列終了ノードを指しているか否かを調べる。もし、文字列終了ノードを指している場合には、入力文字列が受理されたとみなし、ｐを戻り値として返して処理を終了する（ステップ１６０１）。

【００９７】

（２）次に、すでに全ての文字を処理し終わったか否か判定する。ｉがＬより大きく、全ての文字を処理し終わっているにもかかわらず、地名表記ネットワークの終端にｐが達していない場合、NULLを返す（ステップ１６０２）。

【００９８】

（３）次に、ｐのデータ項目ｃが文字列のｉ番目の文字C_iと一致するか否かを調べる。もし一致すれば、ｐの子ノードｐ－＞ｄを探索の起点とし、ｉ＋１番目から文字列を処理するように、関数srchを再帰的に呼び出す。この戻り値ｒがNULLでなければ、文字列が受理されたとみなし、ｒを戻り値として処理を終了する（ステップ１６０３）。

【００９９】

（４）次に、pがNULL遷移に対応するノードか調べる。もしそうであれば、ｐの子ノードｐ－＞ｄを探索の起点とし、ｉ番目から文字列を処理するように、関数srchを再帰的に呼び出す。この戻り値ｒがNULLでなければ、文字列が受理されたとみなし、ｒを戻り値として終了する（ステップ１６０４）。

【０１００】

（５）次に、ｐに兄弟ノードｐ－＞ｂが連結されているかどうか調べる。もし連結されていれば、ｐの兄弟ノードｐ－＞ｂを探索の起点とし、ｉ番目から文字列を処理するように、関数srchを再帰的に呼び出し、この戻り値を上位に返す（ステップ１６０５）。

【０１０１】

（６）もし、前述のいずれの処理でも入力文字列が受理されなければ、これ以上の探索はできないため、NULLを戻り値として処理を終了する（ステップ１６０６）。

【０１０２】

前述までで説明した本発明の実施形態は、文字切出し、文字認識、文字列照合を順次行うとして説明したが、本発明は、古賀他「宛名読取り装置および郵便物等区分機および文字列認識方法」（特願平１０－２８０７７号〔特許第３２４６４３２号公報参照〕）のように文字列照合結果を文字切出しにフィードバックする方式に容易に拡張することもできる。

【０１０３】

図１７は本発明の実施形態による地名文字列認識の処理を応用したシステムの構成例を示すブロック図、図１８は地名文字列生成規則編集装置の構成を示すブロック図である。このシステム例は、郵便区分システムに本発明を適用した例である。図１７、図１８におい

て、１７０１は郵便区分機、１７０２はスキャナ、１７０３はディレイライン、１７０４はソーター、１７０５は地名認識装置、１７０６は入力用インタフェース、１７０７は演算処理装置、１７０８は出力用処理装置、１７１０はメモリ、１７１１はネットワークインタフェース、１７１２はハードディスク、１７１３はメディア着脱可能記憶装置、１７１４は地名文字列生成規則編集装置、１７１８はネットワーク、１８０１はマウス、１８０２はキーボード、１８０３はディスプレイ、１８０４は地名文字列生成規則編集プログラム、１８０５は文字列照合プログラム、１８０６は地名表記ネットワーク表示プログラム、１８０７は地名文字列生成規則ファイル、１８０８は地名表記ネットワーク生成プログラム、１８０９は地名表記ネットワークデータ、１８１０は通信装置、１８１１はメディア着脱可能記憶装置、１８１２はコンピュータである。

【０１０４】

図１７に示すシステムは、１台または複数台の郵便区分機１７０１と、１台または複数台の地名文字列生成規則編集装置１７１４とがネットワーク１７１８で接続されて構成される。郵便区分機１７０１は、スキャナ１７０２、ディレイライン１７０３、ソータ１７０４、地名認識装置１７０５から構成される。また、地名認識装置１７０５は、入力用インタフェース１７０６、演算処理装置１７０７、出力用処理装置１７０８、メモリ１７１０、ネットワークインタフェース１７１１、ハードディスク１７１２、メディア着脱可能記憶装置１７１３から構成される。なお、図における太線は、郵便物の流れを示す。

【０１０５】

図１７に示すシステムにおいて、スキャナ１７０２から入力された郵便物に記載されている地名の画像情報は、地名認識装置１７０５へ転送される。そして、郵便物がディレイライン１７０３を搬送される間に、地名認識装置１７０５は、郵便物に記載されている地名を認識し、認識結果をソータ１７０４へ転送する。ソータ１７０４は、認識結果に応じて郵便物を区分する。

【０１０６】

郵便物の区分の準備段階として、地名認識装置１７０５は、ハードディスク１７１２から地名表記ネットワーク生成プログラムファイルをメモリ１７１０に読み込んで演算装置１７０７で起動する。地名表記ネットワーク生成プログラムの制御の下に、地名認識装置１７０５は、地名文字列生成規則を地名文字列生成規則編集装置１７１４からネットワークインタフェース１７１１を介して入力し、地名表記ネットワークファイルを作成してハードディスク１７１２に格納する。

【０１０７】

なお、地名文字列生成規則は、ネットワークを介して地名文字列生成規則編集装置１７１４から入力する代わりに、フロッピーディスクドライブ等のメディア着脱可能記憶装置１７１３より読み込んでもよい。

【０１０８】

地名認識装置１７０５は、郵便物の区分時に、ハードディスク１７１２から認識プログラムファイル及び地名表記ネットワークファイルをメモリ１７１０に読み込んで演算装置１８０７により実行する。そして、地名認識装置１７０５は、認識プログラムの制御の下に、入力インタフェース１７０６から画像を入力し、郵便物に記載された地名を認識し、認識結果を出力インタフェース１７０８を介して出力する。

【０１０９】

地名文字列生成規則編集装置１７１４は、図１８に示すように、コンピュータ１８１２に、マウス１８０１、キーボード１８０２、ディスプレイ１８０３、地名文字列生成規則ファイル１８０７を格納するディスク装置、通信装置１８１０、メディア着脱可能記憶装置１８１１を接続して構成される。編集作業は、コンピュータ１８１２上で動作する地名文字列生成規則編集プログラム１８０４を介し地名文字列生成規則ファイル１８０７を編集することにより実行される。地名文字列生成規則ファイル１８０７は、テキストファイルであり、編集には通常のテキストエディタを用いることができる。また、コンピュータ１８１２上で地名表記ネットワーク生成プログラム１８０８を実行し、地名文字列生成規則

ファイル１８０７から地名表記ネットワーク１８０９を生成することができる。
【０１１０】
地名文字列生成規則編集装置１７１４は、前述の機能により、編集中の地名単語生成規則が文法的に正しいか否かを確認することができ、さらに、認識処理での文字列照合１４０４と等価なプログラム１８０５を実行し、キーボード１８０３から入力された試験用の文字列が受理されるか否かを確認することができる。
【０１１１】
また、コンピュータ１８１２は、地名表記ネットワーク１８０９を、例えば、図３に示すような形式で表示するための地名表記ネットワーク表示プログラム１８０６を実行するので、作業者は、編集結果を視覚的に確認することができる。編集した結果の地名文字列生成規則ファイル１８０７は、通信装置１８１０を介して地名認識装置１７０５へ転送され、あるいは、メディア着脱可能記憶装置１８１１によりフロッピーディスクなどの着脱可能な記憶メディアに複写され、記憶メディアにより郵便区分機１７０１へ輸送されてもよい。
【０１１２】
図１９は本発明の他の実施形態の構成を示すブロック図、図２０はディスプレイに表示される画面例を説明する図である。この例は、本発明による地名文字列の表記方法及び地名照合方式を利用し、地名を表す文字列から地名に関する情報を検索するための地名録装置である。図１９において、１９０１はマウス、１９０２はキーボード、１９０３はディスプレイ、１９０４はプリンタ、１９０５は入力ファイル、１９０６は出力ファイル、１９０７は地名録プログラム、１９０８は地名付加情報ファイル、１９０９は地名文字列生成規則ファイル、１９１０は通信モジュール、１９１１はインタフェースモジュール、１９１２は地名リストデータ、１９１３は地名リストソートモジュール、１９１４は地名情報検索モジュール、１９１５は地名リスト生成モジュール、１９１６は文字列照合モジュール、１９１７は地名表記展開モジュール、１９１８は地名表記ネットワーク生成プログラム、１９１９は地名表記ネットワークデータである。
【０１１３】
図１９に示す装置は、以下のようなサービスを提供するものである。
（１）キーボードから入力された地名文字列の標準形を表示または印刷する。
（２）キーボードから入力された地名文字列の異表記を表示または印刷する。
（３）キーボードから入力された地名文字列に対応する地域の情報（郵便番号など）を表示または印刷する。
（４）ファイルから入力した地名文字列を標準形または郵便番号等該当する地域に固有の情報に変換してファイルへ出力する。
（５）ネットワークから入力した地名文字列を標準形または郵便番号等該当する地域に固有の情報に変換してネットワークへ出力する。
前述において、標準形とは、例えば、行政区分で定められているある地域を表す正式な文字列のことである。
【０１１４】
図１９に示す実施形態は、計算機上で実行される地名録プログラム１９０７に、マウス１９０１、キーボード１９０２、ディスプレイ１９０３、プリンタ１９０４、入力ファイル１９０５、出力ファイル１９０６、地名付加情報ファイル１９０８、地名文字列生成規則ファイル１９０９が接続されて構成される。表示、入出力は、インタフェースモジュール１９１１を介して行われる。検索対象の文字列が入力されると、地名情報検索モジュール１９１４は、文字列照合モジュール１９１６を呼び出す。文字列照合モジュール１９１６は、図１４における文字列照合処理１４０４と等価な処理を司るモジュールであり、地名表記生成規則ファイル１９０９から地名表記ネットワーク生成プログラム１９１８によって生成された地名表記ネットワークデータ１９１９を参照し、入力文字列がいかなる識別子の地名表記に該当するかを調べる。
【０１１５】

地名情報検索モジュール１９１４は、得られた識別子を手がかりに、地名付加情報ファイル１９０８から、標準形と、郵便番号等の付加的な情報とを検索する。また、地名表記展開モジュール１９１７は、地名表記ネットワークデータ１９１９からあり得る異表記を全て列挙する。得られた異表記群は、地名リストデータ１９１２に格納し、必要に応じインタフェースモジュール１９１１を介して出力する。また、地名リストソートモジュール１９１３は、操作者の指示に従い、異表記群の順序を並べ替えて出力する。このような処理のための入力は、キーボード１９０１、入力ファイル１９０５、通信モジュール１９１０のいずれを介して行われてもよい。また、出力は、ディスプレイ１９０４、出力ファイル１９０６、通信モジュール１９１０のいずれを介して行ってもよい。

【０１１６】

図２０に示す図１９の実施形態のディスプレイ１９０３に表示される画面例において、図２０（Ａ）に示す例は、操作者が、「川越市小ヶ谷」という文字列を入力して、検索を実行した際にディスプレイ１９０３に表示される画面例である。入力文字列は、フィールド２００５へ入力され、ボタン２００６をマウスでクリックすることにより、検索が実行される。検索の結果、入力文字列に該当することが判った文字列は、ウインドウ２００７に表示される。各行の「標準」の項目には、その文字列が標準形か否かが表わされる。項目「地名」は、その文字列を表示する。項目「郵便番号」には、その文字列に対応する郵便番号を表示されるが、その他のその地域の付加情報を表示してもよい。

【０１１７】

領域２００４に並べられた「標準」、「地名」、「郵便番号」の枠はボタンとなっており、各ボタンをマウスによりクリックすることにより、それぞれの項目に基づいた行の並べ換えを指示する。ウインドウ２００８は、検索のオプションを指定するためのものである。ここで、標準形のみを表示するか、字、大字等に基づく異表記群を表示するか、通称名（「＊＊団地」等）に基づく異表記群を表示するかを指定する。ボタン２００２は、表示内容の印刷を指示するためのボタンであり、ボタン２００１は、キーボードとディスプレイとに代わり、ファイルを入出力するモードへの切り替えのためのボタンである。また、ボタン２００３は、プログラムの終了を指示するためのボタンである。

【０１１８】

図２０（Ｂ）に開かれたウィンドウ２００９は、照合の結果得られた地名の読み方、小字、郵便番号等の詳細な情報を表示するウインドウである。このウインドウ２００９は、ウインドウ２００７上に表示された検索結果をマウスでクリックすることにより起動される。

【０１１９】

なお、本発明の実施形態による表記方法により表記された地名文字列は、ＦＤ、ＭＯ、ＤＶＤ等の記憶媒体に地名辞書として格納し提供することができる。

【０１２０】

【発明の効果】

以上説明したように本発明によれば、地名の表記に多くの異表記がある場合でも、あり得る全ての地名文字列を網羅する地名辞書を少ない人手で作成することができる。また、高速な照合処理が可能なネットワーク形式の地名辞書を容易に作成することができる。

【図面の簡単な説明】

【図１】本発明の実施形態による地名文字列認識の処理例を説明するフローチャートである。

【図２】編集された地名文字列生成規則により表現された地名の表記例と生成規則を用いずに異表記を羅列した例とを示す図である。

【図３】生成規則の例から作られる地名表記ネットワークを模式的に示す図である。

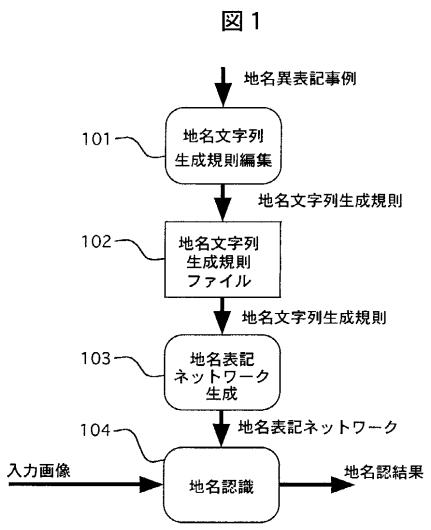
【図４】地名表記ネットワークを計算機上に実装する際のデータ形式を説明する図である。

【図５】地名文字列生成規則から地名表記ネットワークを生成する処理を説明するフローチャートである。

【図６】生成される構文木の例を説明する図である。
【図７】地名表記生成規則から地名表記ネットワーク生成する関数procによる処理動作を説明するフローチャートである。
【図８】関数procによって地名表記ネットワークが生成される過程を説明する図（その１）である。
【図９】関数procによって地名表記ネットワークが生成される過程を説明する図（その２）である。
【図１０】地名表記生成規則から生成された地名表記ネットワーク群を示す図である。
【図１１】従来技術を用いて地名表記ネットワークを生成する処理手順を説明するフローチャートである。
【図１２】従来技術により生成される地名表記ネットワークの生成過程の例を説明する図である。
【図１３】従来技術により生成された地名表記ネットワークの例を示す図である。
【図１４】図１に示す地名認識処理での処理動作を説明するフローチャートである。
【図１５】図１４に示す文字列照合処理での処理動作を説明するフローチャートである。
【図１６】関数srchの処理動作を説明フローチャートである。
【図１７】本発明の実施形態による地名文字列認識の処理を応用したシステムの構成例を示すブロック図である。
【図１８】地名文字列生成規則編集装置の構成を示すブロック図である。
【図１９】本発明の他の実施形態の構成を示すブロック図である。
【図２０】ディスプレイに表示される画面例を説明する図である。
【符号の説明】
１０１　地名文字列生成規則編集処理
１０２　地名文字列生成規則ファイル
１０３　地名表記ネットワーク生成処理
１０４　地名認識処理
１４０４　文字列照合処理
１７０１　郵便区分機
１７０２　スキャナ
１７０３　ディレイライン
１７０４　ソーター
１７０５　地名認識装置
１７０６　入力用インタフェース
１７０７　演算処理装置
１７０８　出力用処理装置
１７１０　メモリ
１７１１　ネットワークインタフェース
１７１２　ハードディスク
１７１３　メディア着脱可能記憶装置
１７１４　地名文字列生成規則編集装置
１７１８　ネットワーク
１８０１　マウス
１８０２　キーボード
１８０３　ディスプレイ
１８０４　地名文字列生成規則編集プログラム
１８０５　文字列照合プログラム
１８０６　地名表記ネットワーク表示プログラム
１８０７　地名文字列生成規則ファイル
１８０８　地名表記ネットワーク生成プログラム
１８０９　地名表記ネットワークデータ

１８１０　通信装置
１８１１　メディア着脱可能記憶装置
１８１２　コンピュータ
１９０１　マウス
１９０２　キーボード
１９０３　ディスプレイ
１９０４　プリンタ
１９０５　入力ファイル
１９０６　出力ファイル
１９０７　地名録プログラム
１９０８　地名付加情報ファイル
１９０９　地名文字列生成規則ファイル
１９１０　通信モジュール
１９１１　インタフェースモジュール
１９１２　地名リストデータ
１９１３　地名リストソートモジュール
１９１４　地名情報検索モジュール
１９１５　地名リスト生成モジュール
１９１６　文字列照合モジュール
１９１７　地名表記展開モジュール
１９１８　地名表記ネットワーク生成プログラム
１９１９　地名表記ネットワークデータ

【図１】



【図２】

図2

<Wヶ> ::= ヶ|ケ|が
<N3501104> :: = [埼玉県]川越市[大字]小<Wヶ>谷[[字]東(田|関)|西関]
<N3501175> :: = [埼玉県]川越市[大字]笠幡[[字]久保|河南]
<N3500804> :: = [埼玉県]川越市下広谷

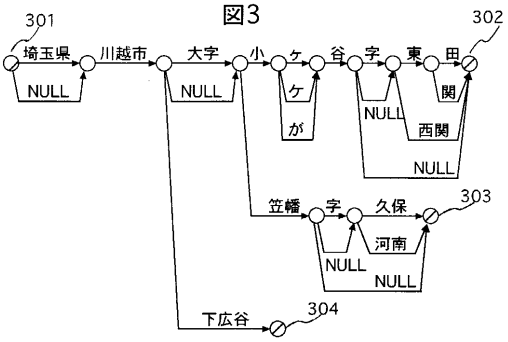
(A)

川越市小ヶ谷
川越市笠幡
川越市下広谷
埼玉県川越市小ヶ谷
埼玉県川越市笠幡
埼玉県川越市下広谷
川越市大字小ヶ谷
川越市大字笠幡
川越市大字下広谷
埼玉県川越市大字小ヶ谷
埼玉県川越市大字笠幡
埼玉県川越市大字下広谷
....

(B)

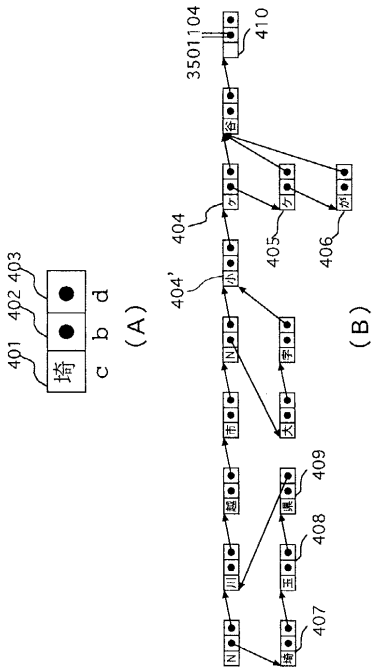
【図３】

図3



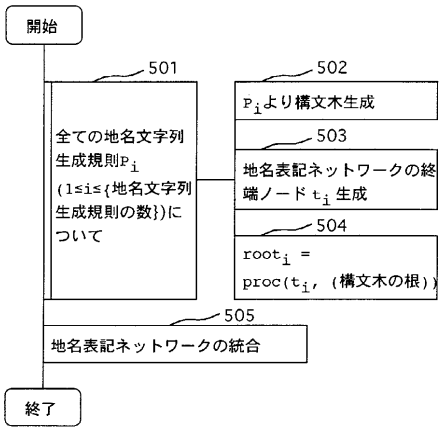
【図４】

図4



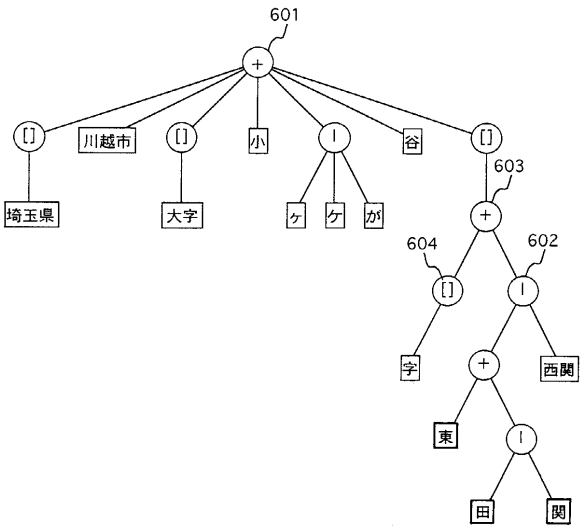
【図５】

図5



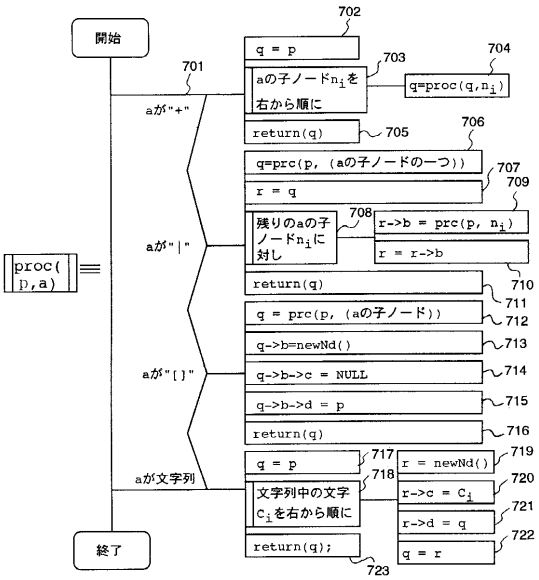
【図６】

図6



【図７】

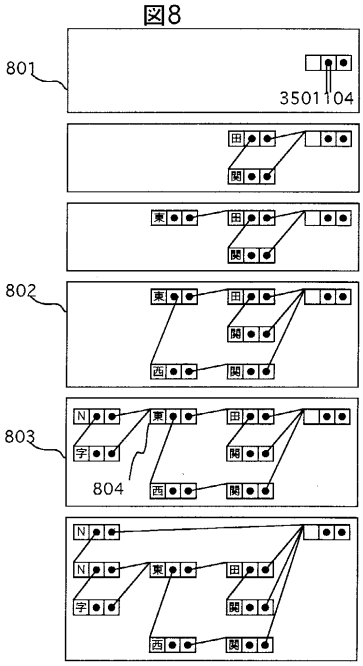
図7



p：地名表記ネットワーク上のノード
（proc(p,a)で生成する部分の終点）
a：処理対象の構文木のノード
ni：aのi番目の子ノード

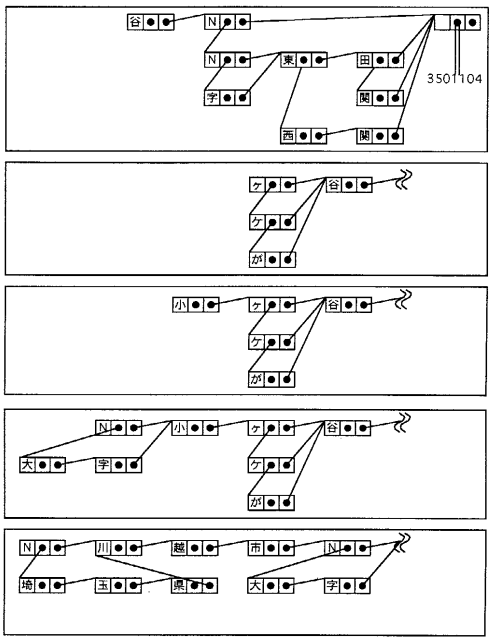
【図８】

図8



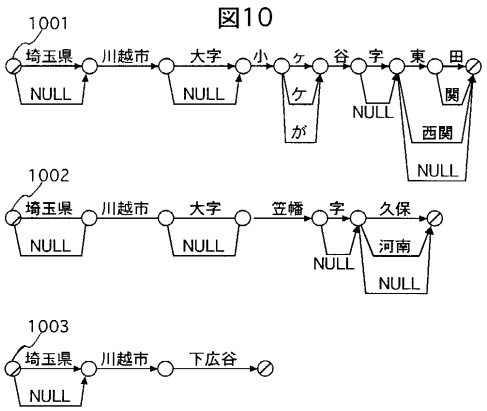
【図９】

図9



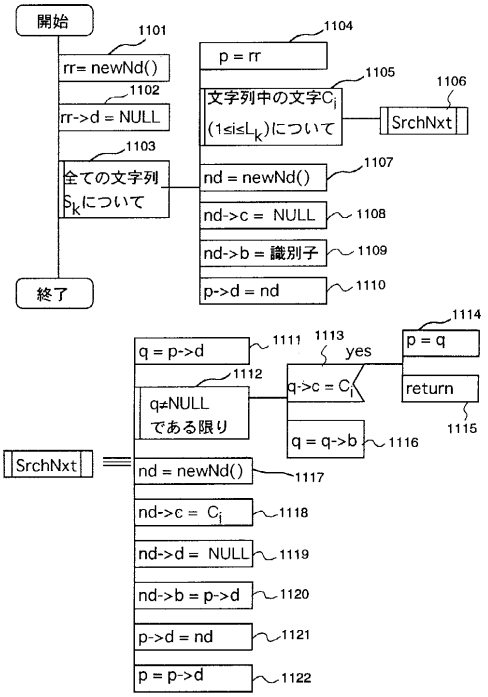
【図１０】

図10



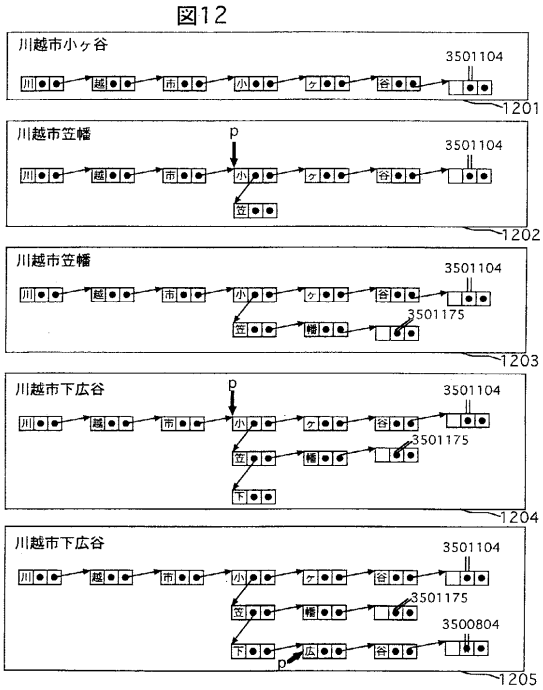
【図１１】

図11



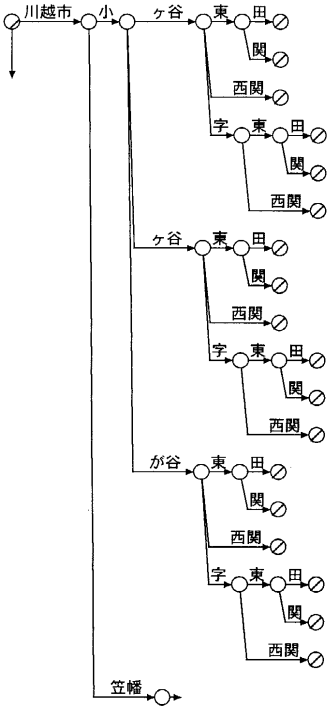
【図１２】

図12



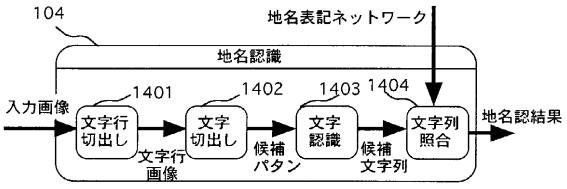
【図１３】

図13



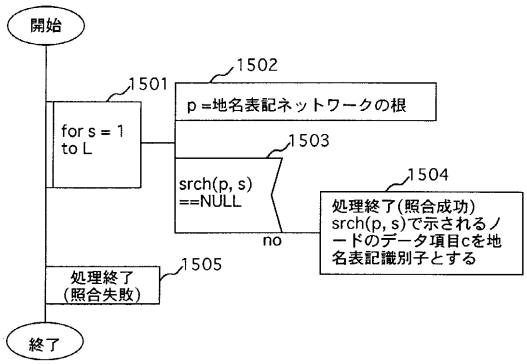
【図１４】

図14



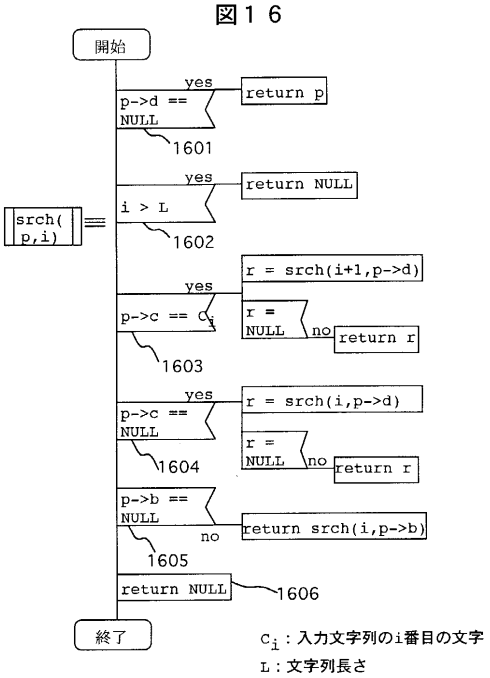
【図１５】

図15



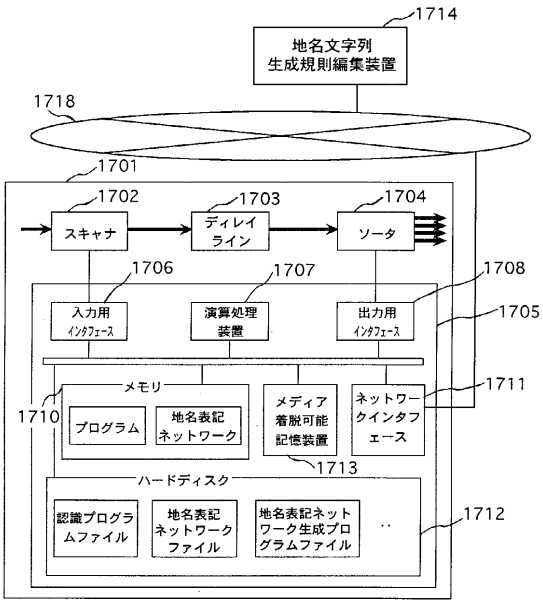
【図１６】

図１６



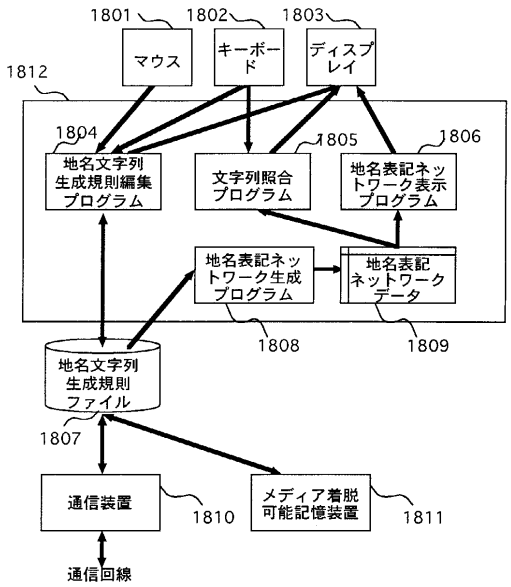
【図１７】

図17



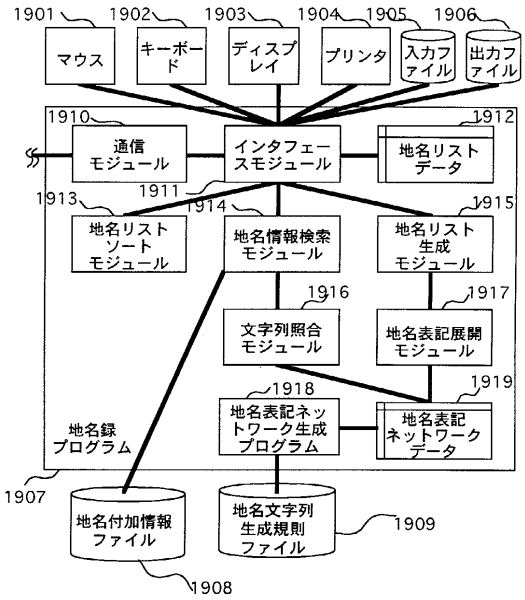
【図１８】

図18



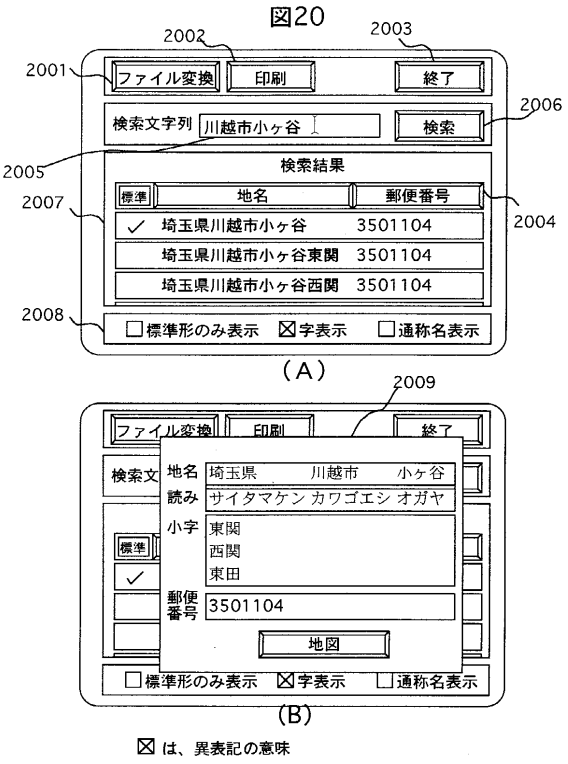
【図１９】

図19



【図２０】

図20



☒ は、異表記の意味

フロントページの続き

(72)発明者　緒方　日佐男
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内
(72)発明者　酒匂　裕
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内
(72)発明者　藤澤　浩道
東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社　日立製作所　中央研究所内

審査官　水野　恵雄

(56)参考文献　特開２０００－１６３４１８（ＪＰ，Ａ）
特開平０４－１５６６６２（ＪＰ，Ａ）
特開２０００－１６３４１８（ＪＰ，Ａ）
特開平０４－１５６６６２（ＪＰ，Ａ）
特開平１１－１２０２９３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 17/22　520
G06F 17/22　520
G06F 17/21　550