(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4965090号
(P4965090)
(45)発行日　平成24年7月4日(2012.7.4)　(24)登録日　平成24年4月6日(2012.4.6)

(51)Int.Cl.　F I
G06F 17/21 (2006.01)　G06F 17/21 530P

請求項の数 13　外国語出願　(全 17 頁)

(21)出願番号　特願2005-187819(P2005-187819)
(22)出願日　平成17年6月28日(2005.6.28)
(65)公開番号　特開2006-85673(P2006-85673A)
(43)公開日　平成18年3月30日(2006.3.30)
審査請求日　平成20年6月27日(2008.6.27)
(31)優先権主張番号　10/943,095
(32)優先日　平成16年9月15日(2004.9.15)
(33)優先権主張国　米国(US)

前置審査

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　110001243
特許業務法人　谷・阿部特許事務所
(74)復代理人　100115624
弁理士　濱中　淳宏
(74)復代理人　100156971
弁理士　稲　綾子

最終頁に続く

(54)【発明の名称】数式の構築を自動化するためのシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
２次元形式の数式を自動的に構築するコンピューティングシステムであって、
プロセッサユニットと、
コンピュータ可読命令を格納するシステムメモリであって、
線形文字列形式の前記数式の入力を受諾する入力モジュールであって、前記数式の入力は、１つまたは複数の英数字の文字と、少なくとも２つの英数字でない演算子とを含む複数の文字で構成される文字列である、入力モジュールと、
前記数式の前記文字列を解釈し、前記数式の前記文字列に２次元形式の数式の構築を行うべき構築点が含まれるかどうかを、前記少なくとも２つの英数字でない演算子に割り当てられた優先順位の値に少なくとも部分的に基づいて判定する解釈モジュールであって、前記優先順位の値は、
キャリッジリターン演算子に対する優先順位の値、
左丸括弧、左角括弧、および右山括弧のうち１つの演算子に対する優先順位の値、
右丸括弧、右角括弧、および左山括弧のうち１つの演算子に対する優先順位の値、
垂直タブ演算子に対する優先順位の値、
分数演算子、ルート演算子、ユニコード積分、合計、積、ｎ項演算子、および階乗以外の他のユニコード数学演算子に対する優先順位の値、
分数演算子に対する優先順位の値、
ルート演算子に対する優先順位の値、

ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値、
下付きおよび上付きのうち１つの演算子に対する優先順位の値、
発音区別符号および階乗のうち１つの演算子に対する優先順位の値
の順に増加する値であり、
前記文字列の前記複数の文字を順次検査していき、
（ｉ）前記文字列の第１の文字に続く第２の文字が英数字でない第２の演算子であって、かつ該第２の演算子が、右丸括弧、右角括弧、および右山括弧のいずれかの演算子であるか、左丸括弧、左角括弧、および左山括弧のいずれでもない演算子であることと、
（ｉｉ）前記第２の演算子の優先順位の値が、前記文字列において前記第２の演算子の前に現れる英数字でない第１の演算子の優先順位よりも小さいこと、あるいは前記第１および第２の演算子の双方の優先順位の値が、前記他のユニコード数学演算子に対する優先順位の値であるか、前記分数演算子に対する優先順位の値であるか、前記ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値であること
の双方の条件が満たされたとき、前記第２の演算子を前記構築点であると判定する、解釈モジュールと、
前記文字列に前記構築点が見つかったとき、前記数式の一部分であって、前記文字列内の前記検査を開始した文字から前記構築点までの文字を含む部分を、前記２次元形式の数式にフォーマットする自動構築モジュールと
を含む、システムメモリと
を備えることを特徴とするシステム。

【請求項２】
前記２次元形式にフォーマットされた前記数式の一部分を編集することを可能にする修正モジュールをさらに備えることを特徴とする請求項１に記載のシステム。

【請求項３】
前記線形文字列形式の前記数式を編集することを可能にする修正モジュールをさらに備えることを特徴とする請求項１に記載のシステム。

【請求項４】
前記数式を表示する表示モジュールをさらに備えることを特徴とする請求項１に記載のシステム。

【請求項５】
コンピュータシステムに入力された線形文字列形式の数式から２次元形式の数式を自動的に構築する方法であって、
線形文字列形式の数式の入力を受諾するステップであって、前記数式の入力は、１つまたは複数の英数字の文字と、少なくとも２つの英数字でない演算子とを含む複数の文字で構成される文字列である、ステップと、
前記数式の前記文字列に、２次元形式の数式の構築を行うべき構築点が含まれるかどうかを、前記少なくとも２つの英数字でない演算子に割り当てられた優先順位の値に少なくとも部分的に基づいて判定するステップ、前記優先順位の値は、
キャリッジリターン演算子に対する優先順位の値、
左丸括弧、左角括弧、および右山括弧のうち１つの演算子に対する優先順位の値、
右丸括弧、右角括弧、および左山括弧のうち１つの演算子に対する優先順位の値、
垂直タブ演算子に対する優先順位の値、
分数演算子、ルート演算子、ユニコード積分、合計、積、ｎ項演算子、および階乗以外の他のユニコード数学演算子に対する優先順位の値、
分数演算子に対する優先順位の値、
ルート演算子に対する優先順位の値、
ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値、
下付きおよび上付きのうち１つの演算子に対する優先順位の値、
発音区別符号および階乗のうち１つの演算子に対する優先順位の値
の順に増加する値であり、

前記文字列の前記複数の文字を順次検査していき、
（ｉ）前記文字列の第１の文字に続く第２の文字が英数字でない第２の演算子であって、かつ該第２の演算子が、右丸括弧、右角括弧、および右山括弧のいずれかの演算子であるか、左丸括弧、左角括弧、および左山括弧のいずれでもない演算子であることと、
（ｉｉ）前記第２の演算子の優先順位の値が、前記文字列において前記第２の演算子の前に現れる英数字でない第１の演算子の優先順位よりも小さいこと、あるいは前記第１および第２の演算子の双方の優先順位の値が、前記他のユニコード数学演算子に対する優先順位の値であるか、前記分数演算子に対する優先順位の値であるか、前記ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値であること
の双方の条件が満たされたとき、前記第２の演算子を前記構築点であると判定するステップと、
前記文字列に前記構築点が見つかったとき、前記数式の一部分であって、前記文字列内の前記検査を開始した文字から前記構築点までの文字を含む部分を、前記２次元形式の数式にフォーマットするステップと
を含むことを特徴とする方法。
【請求項６】
前記２次元形式としてフォーマットされた前記数式の一部分を編集するステップをさらに含むことを特徴とする請求項５に記載の方法。
【請求項７】
前記数式を編集するステップをさらに含むことを特徴とする請求項５に記載の方法。
【請求項８】
前記線形文字列形式の前記数式の追加入力を受諾するステップと、
前記追加入力が、少なくとも２つの英数字でない演算子を含む複数の文字で構成される第２の文字列であるとき、前記第２の文字列に２次元形式の数式の構築を行うべき第２の構築点が含まれるかどうかを、前記第２の文字列内の前記少なくとも２つの英数字でない演算子に割り当てられた優先順位の値に少なくとも部分的に基づいて判定するステップであって、
前記第２の文字列の前記複数の文字を順次検査していき、
（ｉｉｉ）前記第２の文字列の第１の文字に続く第２の文字が英数字でない第４の演算子であって、かつ該第４の演算子が、右丸括弧、右角括弧、および右山括弧のいずれかの演算子であるか、左丸括弧、左角括弧、および左山括弧のいずれでもない演算子であることと、
（ｉｖ）前記第４の演算子の優先順位の値が、前記第２の文字列において前記第４の演算子の前に現れる英数字でない第３の演算子の優先順位よりも小さいこと、あるいは前記第３および第４の演算子の双方の優先順位の値が、前記他のユニコード数学演算子に対する優先順位の値であるか、前記分数演算子に対する優先順位の値であるか、前記ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値であること
の双方の条件が満たされたとき、前記第４の演算子を前記第２の構築点であると判定する、ステップと、
前記第２の文字列に前記第２の構築点が見つかったとき、前記数式の前記第２の文字列の一部分であって、前記第２の文字列内の前記検査を開始した文字から前記第２の構築点までの文字を含む第２の部分を、前記線形文字列形式から前記２次元形式に変換するステップと
をさらに含むことを特徴とする請求項５に記載の方法。
【請求項９】
前記数式を表示するステップをさらに含むことを特徴とする請求項５に記載の方法。
【請求項１０】
コンピュータ可読であり、請求項５に記載の方法を実行させるための命令をエンコードすることを特徴とするコンピュータプログラム。
【請求項１１】

コンピュータシステムに入力された線形文字列形式の数式の一部を２次元形式の数式として自動的に構築する方法であって、

（ａ）線形文字列形式の数式の入力を受諾するステップであって、前記数式の入力は、１つまたは複数の英数字の文字と、少なくとも２つの英数字でない演算子とを含む複数の文字で構成される第１の文字列である、ステップと、

（ｂ）前記数式の前記第１の文字列の前記複数の文字を順次検査して、前記数式の前記第１の文字列に２次元形式の数式の構築を行うべき第１の構築点が含まれるかどうかを、前記第１の文字列内の前記少なくとも２つの英数字でない演算子に割り当てられた優先順位の値に少なくとも部分的に基づいて判定するステップであって、前記優先順位の値は、

キャリッジリターン演算子に対する優先順位の値、

左丸括弧、左角括弧、および右山括弧のうち１つの演算子に対する優先順位の値、

右丸括弧、右角括弧、および左山括弧のうち１つの演算子に対する優先順位の値、

垂直タブ演算子に対する優先順位の値、

分数演算子、ルート演算子、ユニコード積分、合計、積、ｎ項演算子、および階乗以外の他のユニコード数学演算子に対する優先順位の値、

分数演算子に対する優先順位の値、

ルート演算子に対する優先順位の値、

ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値、

下付きおよび上付きのうち１つの演算子に対する優先順位の値、

発音区別符号および階乗のうち１つの演算子に対する優先順位の値

の順に増加する値であり、

前記第１の文字列の前記複数の文字を順次検査していき、

（ｉ）前記第１の文字列の第１の文字に続く第２の文字が英数字でない第２の演算子であって、かつ該第２の演算子が、右丸括弧、右角括弧、および右山括弧のいずれかの演算子であるか、左丸括弧、左角括弧、および左山括弧のいずれでもない演算子であることと、

（ｉｉ）前記第２の演算子の優先順位の値が、前記第１の文字列において前記第２の演算子の前に現れる英数字でない第１の演算子の優先順位よりも小さいこと、あるいは前記第１および第２の演算子の双方の優先順位の値が、前記他のユニコード数学演算子に対する優先順位の値であるか、前記分数演算子に対する優先順位の値であるか、前記ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値であること

の双方の条件が満たされたとき、前記第２の演算子を前記第１の構築点であると判定するステップと、

（ｃ）前記第１の文字列に前記第１の構築点が見つからなかった場合、前記第１の構築点が見つかるまで、前記線形文字列形式の前記数式の前記第１の文字列に追加される文字の入力を受諾し、前記追加された文字を含む前記第１の文字列について前記（ｂ）のステップを実行するステップと、

（ｄ）前記第１の文字列に前記第１の構築点が見つかった場合、

前記数式の一部分であって、前記第１の文字列の前記検査を開始した文字から前記第１の構築点までの文字を含む第１の部分を、前記線形文字列形式から２次元形式に変換するステップと、

前記数式の前記第１の部分を前記２次元形式で表示するステップと、

前記数式の前記第１の部分を前記２次元形式で表示した後で、前記線形文字列形式の前記数式の第２の文字列の入力を受諾するステップであって、前記第２の文字列は、少なくとも２つの英数字でない演算子を含む複数の文字で構成されるステップと、

（ｅ）前記第２の文字列の前記複数の文字を順次検査して、前記第２の文字列に２次元形式の数式の構築を行うべき第２の構築点が含まれるかどうかを、前記第２の文字列の前記少なくとも２つの英数字でない演算子に割り当てられた前記優先順位の値に少なくとも部分的に基づいて判定するステップであって、

前記第２の文字列の前記複数の文字を順次検査していき、

（ｉｉｉ）前記第２の文字列の第１の文字に続く第２の文字が英数字でない第４の演算子であって、かつ該第４の演算子が、右丸括弧、右角括弧、および右山括弧のいずれかの演算子であるか、左丸括弧、左角括弧、および左山括弧のいずれでもない演算子であることと、

（ｉｖ）前記第４の演算子の優先順位の値が、前記第２の文字列において前記第４の演算子の前に現れる英数字でない第３の演算子の優先順位よりも小さいこと、あるいは前記第３および第４の演算子の双方の優先順位の値が、前記他のユニコード数学演算子に対する優先順位の値であるか、前記分数演算子に対する優先順位の値であるか、前記ユニコード積分、合計、積、またはｎ項演算子に対する優先順位の値であること

の双方の条件が満たされたとき、前記第４の演算子を前記第２の構築点であると判定する、ステップと、

（ｆ）前記第２の文字列に前記第２の構築点が見つからなかった場合、前記第２の構築点が見つかるまで、前記線形文字列形式の前記数式の前記第２の文字列に追加される文字の入力を受諾して、前記追加された文字を含む前記第２の文字列について前記（ｅ）のステップを実行するステップと、

（ｇ）前記第２の文字列に前記第２の構築点が見つかった場合、

前記数式の前記第２の文字列の一部分であって、前記第２の文字列の前記検査を開始した文字から前記第２の構築点までの文字を含む第２の部分を、前記線形文字列形式から２次元形式に変換するステップと、

前記数式の前記第２の部分を前記２次元形式で表示するステップと、

前記数式の前記第２の部分を前記２次元形式で表示した後で、前記線形文字列形式の前記数式の第３の文字列の入力を受諾するステップと

を含むことを特徴とする方法。

【請求項１２】

前記２次元形式で表示された前記数式の前記第１の部分を編集するステップをさらに含むことを特徴とする請求項１１に記載の方法。

【請求項１３】

コンピュータ可読であり、請求項１１に記載の方法を実行させるための命令をエンコードすることを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、コンピュータシステムに記入される数式の解釈および構築のためのシステムおよび方法に関する。

【背景技術】

【０００２】

文書処理アプリケーションおよびｈｔｍｌエディタに数式を効率的に入力できる機能は、より多くの技術情報が文書処理およびウェブページ形式で配信されるようになるのに従って、ますます重要になりつつある。ＴｅＸおよびＬａＴｅＸなどのプログラムにより、ユーザは、様々な計算機環境に渡って移植可能な形式で数式を活字に組み、印刷することができるようになる。しかし、このようなプログラムは複雑であり、ユーザは、式を入力し、活字に組み、印刷することができるようになる前に、プログラムがどのように作用するかに関して、特別な知識をもつことが必要となる。

【０００３】

ワードプロセッサプログラムは通常、ユーザが文書処理環境において数式を作成し編集することを可能にするエディタと一括販売される。このようなエディタの一例が、ワシントン州レドモンドのマイクロソフトコーポレーションによって販売されている、Ｍｉｃｒｏｓｏｆｔ（登録商標）Ｅｑｕａｔｉｏｎ　Ｅｄｉｔｏｒ３．０である。こうしたタイプの数式エディタは一般に、ユーザが、２次元の式を作るために様々なツールバーアイコンの中から選択を行うことを要求するＷＹＳＩＷＹＧエディタである。しかし、ツールバー

アイコンの選択は、複雑で長い式を頻繁に記入する経験豊富なユーザにとってわずらわしい場合がある。

【０００４】

【非特許文献１】M. Sargent III, "Unicode Nearly Plain-Text Encoding of Mathematics," 26th Internationalization & Unicode Conference, San Jose, California (September 2004)

【発明の開示】

【発明が解決しようとする課題】

【０００５】

したがって、このような数式を２次元形式で表示しながら、式の記入を容易にし得るシステムおよび方法を提供することが望ましい。

【課題を解決するための手段】

【０００６】

本発明は、コンピュータシステムに記入される数式の解釈および構築のためのシステムおよび方法に関する。

【０００７】

本発明の一態様は、式を自動的に構築するコンピューティングシステムに関する。本システムは、線形文字列形式での式の入力を受諾する入力モジュールと、入力を解釈し、構築点に到達したときを自動的に判定する解釈モジュールとを含み得る。本システムは、構築点に達すると、式の少なくとも一部分を２次元形式に自動的にフォーマットするフォーマット化モジュールも含み得る。

【０００８】

本発明の別の態様は、コンピュータシステムに記入された数式を自動的に構築する方法に関する。本方法は、線形文字列形式での式の入力を受諾すること、構築点に到達しているかどうか自動的に判定すること、および構築点に到達すると、式の少なくとも一部分を線形文字列形式から２次元形式に自動的に変換することを含み得る。

【０００９】

本発明のさらに別の態様は、コンピュータシステムに記入された数式を自動的に構築する方法に関する。本方法は、線形文字列形式での式の第１の文字の入力を受諾すること、第１の文字を検査して、第１の構築点に到達しているかどうか判定することを含み、第１の構築点に到達している場合、式の第１の部分を線形文字列形式から２次元形式に自動的に変換すること、式の第１の部分を２次元形式で表示すること、およびその後、線形文字列形式での式の第２の文字の入力を受諾し続けることを含み、第１の構築点に到達していない場合、線形文字列形式での式の第２の文字の入力を受諾することを含み得る。

【発明を実施するための最良の形態】

【００１０】

添付の図面に対して参照が行われるが、こうした図面は、必ずしも実物大で描かれているわけではない。

【００１１】

本発明は、これ以降、本発明の実施形態が示されている添付の図面を参照してより詳しく記述される。ただし、本発明は、多くの様々な形で実施されることができ、本明細書において説明される実施形態に限定されると解釈されるべきではない。そうではなく、こうした実施形態は、本開示が詳細かつ完全なものとなるように与えられ、当業者に本発明の範囲を十分に伝える。同じ番号は、全体を通して同じ要素を指す。

【００１２】

本発明は、コンピュータシステムに記入される数式の解釈および構築のためのシステムおよび方法に関する。

【００１３】

本明細書で使用する、「線形文字列形式（ｌｉｎｅａｒ　ｓｔｒｉｎｇ　ｆｏｒｍａｔ）」という言葉は、例えば、ＴｅＸやＬａＴｅＸなど、線形表記法を用いた、数式の、線

形テキストに基づく表現を指す。線形文字列形式の式の例は、「ｘ＝１／２」（「ｘは、２分の１に等しい」）である。

【００１４】

「２次元形式」という言葉は、数式が、例えば、ポーランド式プレフィックス形式など、非線形表記法を用いて表される形式を指す。ポーランド式プレフィックス形式は、関数開始文字を含む形式であり、関数開始文字の後に、例えば、分子、分離文字、分母、および関数終了区切り文字が続く。２次元形式の式の例は、

【００１５】

【数１】

$$x = \frac{1}{2}$$

【００１６】

である（「ｘは、２分の１に等しい」）。

【００１７】

式は、線形文字列形式から２次元形式に、かつその逆に変換され得る。例えば、本明細書で開示される、式のいくつかの部分を線形文字列形式から２次元形式に自動的に変換するシステムおよび方法を含む実施形態などである。

【００１８】

ここで図１を参照すると、コンピュータシステム例１００が示されている。図１に示されるコンピュータシステム１００は、例えば、デスクトップコンピュータ、ラップトップコンピュータ、およびハンドヘルドコンピュータなど、様々な形をとり得る。さらに、コンピュータシステム１００が示されているが、本明細書で開示されるシステムおよび方法は、様々な代替コンピュータシステムでも実装され得る。

【００１９】

システム１００は、プロセッサユニット１０２、システムメモリ１０４、およびシステムメモリなど様々なシステムコンポーネントをプロセッサユニット１０２に結合するシステムバス１０６を含む。システムバス１０６は、様々なバスアーキテクチャのいずれかを使用するメモリバス、周辺バス、およびローカルバスなどいくつかのタイプのバス構造のいずれでもよい。システムメモリは、ＲＯＭ（読出し専用メモリ）１０８およびＲＡＭ（ランダムアクセスメモリ）１１０を含む。ＢＩＯＳ１１２（基本入出力システム）は、コンピュータシステム１００内部の要素間の情報の転送を助ける基本ルーチンを含み、ＲＯＭ１０８に格納される。

【００２０】

コンピュータシステム１００は、ハードディスクからの読出しおよびそこへの書込みを行うハードディスクドライブ１１２、取り外し可能磁気ディスク１１６からの読出しまたはそこへの書込みを行う磁気ディスクドライブ１１４、および、例えばＣＤ　ＲＯＭ、ＤＶＤ、または他の光学媒体などの取り外し可能光ディスク１１９からの読出しまたはそこへの書込みを行う光ディスクドライブ１１８をさらに含み得る。ハードディスクドライブ１１２、磁気ディスクドライブ１１４、および光ディスクドライブ１１８は、それぞれハードディスクドライブインターフェース１２０、磁気ディスクドライブインターフェース１２２、および光ドライブインターフェース１２４によって、システムバス１０６に接続される。こうしたドライブおよびそれに関連するコンピュータ可読媒体は、コンピュータシステム１００用に、コンピュータ可読命令、データ構造、プログラム、および他のデータの不揮発性記憶をもたらす。

【００２１】

本明細書で説明される環境例は、ハードディスク１１２、取り外し可能磁気ディスク１１６、および取り外し可能光ディスク１１９を利用し得るが、データを格納することができる、コンピュータ可読な他のタイプの媒体も、システム例１００において使われ得る。例示的な動作環境において使われ得るこうしたコンピュータ可読な他のタイプの媒体の例

は、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイカートリッジ、ＲＡＭ（ランダムアクセスメモリ）、ＲＯＭ（読出し専用メモリ）を含む。

【００２２】

オペレーティングシステム１２６、１つまたは複数のアプリケーションプログラム１２８、他のプログラムモジュール１３０、およびプログラムデータ１３２など、いくつかのプログラムモジュールが、ハードディスク１１２、磁気ディスク１１６、光ディスク１１９、ＲＯＭ１０８、またはＲＡＭ１１０に格納され得る。

【００２３】

ユーザは、例えばキーボード１３４、マウス１３６、または他のポインティングデバイスなどの入力デバイスを介して、コマンドおよび情報をコンピュータシステム１００に入力することができる。他の入力デバイスの例は、ツールバー、メニュー、タッチスクリーン、マイクロホン、ジョイスティック、ゲーム用パッド、ペン、衛星パラボラアンテナ、およびスキャナを含む。こうしたおよび他の入力デバイスはしばしば、システムバス１０６に結合されるシリアルポートインターフェース１４０を介して処理ユニット１０２に接続される。それにもかかわらず、こうした入力デバイスは、パラレルポート、ゲームポート、ＵＳＢ（ユニバーサルシリアルバス）など、他のインターフェースによっても接続され得る。ＬＣＤディスプレイ１４２または他のタイプの表示デバイスも、ビデオアダプタ１４４などのインターフェースを介してシステムバス１０６に接続される。ディスプレイ１４２に加え、コンピュータシステムは通常、スピーカおよびプリンタなど、他の周辺出力デバイス（図示せず）を含み得る。

【００２４】

コンピュータシステム１００は、リモートコンピュータ１４６など、１つまたは複数のリモートコンピュータへの論理接続を使用してネットワーク接続された環境において動作し得る。リモートコンピュータ１４６は、コンピュータシステム、サーバ、ルータ、ネットワークＰＣ、ピアデバイスまたは他の共通ネットワークノードでよく、通常、コンピュータシステム１００に関連して上述された要素の多くまたはすべてを含む。ネットワーク接続は、ＬＡＮ（ローカルエリアネットワーク）１４８およびＷＡＮ（ワイドエリアネットワーク）１５０を含む。このようなネットワーク環境は、会社、企業規模のコンピュータネットワーク、イントラネットおよびインターネットにおいてよく見られる。

【００２５】

ＬＡＮネットワーク環境において使われる場合、コンピュータシステム１００は、ネットワークインターフェースまたはアダプタ１５２を介してローカルネットワーク１４８に接続される。ＷＡＮネットワーク環境において使われる場合、コンピュータシステム１００は通常、モデム１５４、または、インターネットなどのワイドエリアネットワーク１５０を介した通信を確立する他の手段を含む。モデム１５４は、内部にあっても外部にあってもよく、シリアルポートインターフェース１４０を介してシステムバス１０６に接続される。ネットワーク接続された環境では、コンピュータシステム１００に関連して図示されるプログラムモジュールまたはその一部は、リモートメモリ記憶装置に格納され得る。図示したネットワーク接続は例であり、コンピュータ間の通信リンクを確立する他の手段も使われ得ることが理解されよう。

【００２６】

本明細書で説明される実施形態は、コンピューティングシステムにおける論理演算として実装され得る。論理演算は、（１）コンピュータによって実装される一連のステップまたはコンピュータシステム上で実行されるプログラムモジュールとして、かつ（２）コンピューティングシステム内で実行される、相互に関連する論理またはハードウェアモジュールとして実装され得る。この実装は、特定のコンピューティングシステムの性能要件に依存する、選択の問題である。したがって、本明細書で説明される実施形態をなす論理演算は、動作、ステップ、またはモジュールと呼ばれる。こうした動作、ステップ、およびモジュールは、本明細書に添付された特許請求の範囲に記載される本発明の精神および範囲から逸脱することなく、ソフトウェア、ファームウェア、特殊目的デジタル論理、およ

びこれらのどの組合せとしても実装され得ることが当業者には理解されよう。このソフトウェア、ファームウェア、または同様のコンピュータ命令シーケンスは、エンコードされ、コンピュータ可読記憶媒体に格納されることができ、コンピューティングデバイスの間での伝送用に、搬送波信号にエンコードされることもできる。

【００２７】

ここで図２を参照すると、式の一部分を自動的に構築するシステム例２００が示されている。システム２００は、入力モジュール２１０、解釈モジュール２２０、自動構築モジュール２３０、表示モジュール２４０、および修正モジュール２５０を含む。システム２００は、例えば、上述したコンピュータシステム１００上で実行されるアプリケーションでよい。

【００２８】

入力モジュール２１０は、ユーザが、システム２００に式を入力することを可能にする。例えば、ユーザは、上述されたキーボード１３４および／またはマウス１３６などの入力デバイスを使って、式を入力することができる。一実施形態では、入力モジュール２１０は、ユーザが、線形文字列形式表記法を用いて式を入力することを可能にする。線形文字列形式表記法の例は、ＴｅＸおよびＬａＴｅＸを含む。他の線形文字列形式表記法も用いられ得る。

【００２９】

例えば、一実施形態では、入力モジュール２１０は、Ｓｃｒｏｌｌ　Ｓｙｓｔｅｍｓ，Ｉｎｃ．によって提供されるＰＳ　Ｔｅｃｈｎｉｃａｌ　Ｗｏｒｄ　Ｐｒｏｃｅｓｓｏｒというアプリケーションに組み込まれており、かつ、例えば、非特許文献１に記載されているのと同様の線形形式表記法を用いた式の入力を受諾する。線形形式表記法は、ＴｅＸおよびＬａＴｅＸの表記法と似ているが、コンピュータ言語コンパイラによって翻訳されるのと同様の、簡略化された表記法である。

【００３０】

例えば、入力モジュール２１０は、線形文字列形式での、以下の式Ａの入力を受諾することができる。

ｙ＝ａ／（ｂ＋ｃ）＋ｄ　　（Ａ）

【００３１】

ユーザが、線形文字列形式で式を記入し始めると、解釈モジュール２２０は、入力を解釈して、式の一部分が、２次元形式に自動的に構築され得るときを識別する。

【００３２】

概して、ある文字に先行するテキストを、有効な線形形式文法（例えば、ＴｅＸ、ＬａＴｅＸ、Ｌｉｎｅａｒ　Ｆｏｒｍａｔ）によって定義される構築表現用に、線形にフォーマットされた明白なテキストとさせるような文字をユーザが記入すると、構築が起こる。式の一部分が構築され得るかどうかに関する決定は、文字に先行するオペランドと比較した、直前に記入された文字およびその優先順位の分析に基づく。概して、各文字は、その文字を検討して、どのような構築が起こり得るかを判定するために調べられる。例えば、ユーザが、オペランドをタイプ入力し、次いで、オペランドとは分離した文字をタイプ入力すると、その文字の優先順位が、先行文字の優先順位以下の場合、自動構築が起こり得る。例えば、プラス記号「＋」などの演算子は、先行演算子が除算記号「／」である場合、自動構築を引き起こし得る。

【００３３】

以下の例は、どのようにして、解釈モジュール２２０が、式の一部分の構築が起こり得るときを識別するように構成され得るかを示す。線形形式が例として使われる。ただし、他の線形文字列形式文法（例えば、ＴｅＸ、ＬａＴｅＸ）も同様にして使われ得る。

【００３４】

構築点を識別するために、解釈モジュール２２０は、ユーザによってタイプ入力された各文字を検査する。自動構築が起こるべきときを判定するのに、優先順位指向技術が用いられる。例えば、一実施形態では、線形形式文法の演算子は、以下の表１で与えられる優

先順位をもつ。

【００３５】

【表１】

表１：演算子優先順位

演算子	優先順位
キャリッジリターン	0
「(」または「[」または「{」または「<」	1
「)」または「]」または「}」または「>」	2
垂直タブ	3
他のユニコード数学演算子（すなわち、表1の他の優先順位レベルに列挙されていない演算子）	4
分数（「/」およびatop）	5
オペランドの、配列への変換を示す矩形または演算子での、オペランドの囲い込みを示す$\sqrt{}$または$\sqrt[3]{}$または$\sqrt[4]{}$または演算子	6
ユニコード積分、合計、積、および他のn項op	7
下付きまたは上付き	8
発音区別符号または階乗	9

【００３６】

表１で与えられている演算子優先順位は例示であり、優先順位に対する変更も行われ得る。

【００３７】

自動構築をトリガするかどうか判定するために入力を検査するとき、解釈モジュール２２０は、数式イタリックへの／からの変換を最初に調べる。タイプ入力された文字が、数式イタリック対応物を有するアスキーまたはギリシャ文字である場合、その文字は、数式イタリックバージョンに翻訳される。文字が、下付き「_」、上付き「^」、またはスペースであり、先行文字が数式イタリックであり、式の中でさらに多くの文字がその文字に先行する場合、数式イタリックの範囲は、関数名の辞書と比較される。見つかった場合、名称は、通常のテキストに翻訳し戻される（例えば、「ｓｉｎ」は、「ｓｉｎ」に翻訳し戻される）。

【００３８】

このような翻訳が行われない場合、構築される関数中に挿入点（ＩＰ）があれば、解釈モジュール２２０は、現在の引数の先頭で始まり、あるいは、どちらがＩＰの最も近くにあろうとも、式全体の始まりまたはＩＰに先行するキャリッジリターンで始まる。

【００３９】

次に、起こり得る自動構築の選択範囲は、第１の主構築演算子、（すなわち、「）」または「］」または「｝」または「＞」の１つではない演算子）に進むことによって狭められる。ＩＰに戻る前にこのような演算子が発見された場合、範囲は、分数の分子または下付き「_」もしくは上付き「^」記号の基字（ｓｃｒｉｐｔｂａｓｅ）を含むように再度拡大される。次いで、この範囲内で、テキストに対して構築が試みられる。

【００４０】

具体的には、解釈モジュール２２０は、この範囲を走査し、単なるオペランドをリッチテキスト列スタックにプッシュし、演算子を演算子スタックにプッシュしていく。表現のすぐ後に続く演算子が「）」または「］」または「｝」または「＞」のとき、あるいは演算子が「（」または「［」または「｛」または「＜」でなく、かつ以下の条件の１つが真のとき、表現の自動構築がトリガされる。（ｉ）演算子の優先順位が、その前の演算子の優先順位より小さい、あるいは（ｉｉ）演算子およびその前の演算子の優先順位が両方と

も、４、５、または７と等しい。

【００４１】

いくつかの実施形態では、構築される関数の引数の中で変更が行われる（すなわち、ＩＰが、既に構築されている式の一部分の中にある）場合、自動構築をいつトリガするか判定する、解釈モジュール２２０による分析は、編集されている引数に制限され得る。この制限は、解釈モジュール２２０によって実施される分析を容易にし、そうすることによって処理効率を向上させ得る。他の実施形態では、解釈モジュール２２０は、構築される引数が編集されるかどうかにかかわらず、式全体を分析するように構成され得る。

【００４２】

自動構築を示す一例では、解釈モジュール２２０は、上の式Ａを解釈し、式Ａの一部分が構築され得るときを識別する。式Ａに対して、解釈モジュール２２０は、ユーザによって、式Ａの入力中に、右括弧「）」の後にスペースが記入されると、自動構築をトリガする。式Ａのこの部分が、以下の式Ａ’として示されている。

ｙ＝ａ／（ｂ＋ｃ）　　（Ａ’）

【００４３】

以下の擬似コードは、解釈モジュール２２０が、自動構築がトリガされ得るときを識別するために、ユーザによる式Ａの式Ａ’部分の入力をどのようにして解釈し得るかの一例を示す。

【００４４】

【表２】

ユーザによる 入力モジュールへの入力	解釈モジュールによるアクション（群）
a	「a」は外部記憶装置に入る 「a」を数式イタリック「*a*」に翻訳する 自動的な構築をトリガしようと試みるが失敗する
/	「/」は外部記憶装置に入る 自動的な構築をトリガしようと試みるが失敗する
(	「(」は外部記憶装置に入る 自動的な構築をトリガしようと試みるが失敗する
b	「b」は外部記憶装置に入る 「b」を数式イタリック「*b*」に翻訳する 自動的な構築をトリガしようと試みるが失敗する
+	「+」は外部記憶装置に入る 自動的な構築をトリガしようと試みるが失敗する
c	「c」は外部記憶装置に入る 「c」を数式イタリック「*c*」に翻訳する 自動的な構築をトリガしようと試みるが失敗する
)	「)」は外部記憶装置に入る 自動的な構築をトリガしようと試みるが失敗する
スペース	自動的な構築を首尾よくトリガする

【００４５】

上で与えられた擬似コードに示されるように、自動構築は、自動構築点を表す、所定の文字の１つを識別する解釈モジュール２２０において、常に成功するわけではない。例えば、式Ａ’に対して、解釈モジュール２２０は、プラス記号「＋」が記入されたとき、自動構築をトリガしない。というのは、式Ａの記入におけるその時点では、より大きい優先順位をもつ先行演算子がないので、式のどの部分も２次元形式に構築されることができないからである。さらに、解釈モジュール２２０は、左括弧「（」が記入されたとき、自動構築をトリガしない。というのは、式Ａの一部分の構築が起こり得る前に必要とされる付

加情報が記入される可能性があるからである。しかし、左括弧「（」の後にスペースが記入されると、スペースのすぐ後に続く文字が「）」または「］」または「｝」または「＞」の１つなので、解釈モジュール２２０は、式Ａの式Ａ’部分の自動構築をトリガし得る（上の表１の優先順位値２を参照）。

【００４６】

上の擬似コードに記述されるように、記入された各文字は、外部記憶装置に置かれる。外部記憶装置は、タイプ入力されたデータが格納され取得され得る、永続的な記憶空間である。さらに、上の擬似コードによって示されるように、解釈モジュール２２０は、例えば、数式イタリック形式への変数の変換など、式のフォーマット化のいくつかの側面も実施することができる。

【００４７】

解釈モジュール２２０が、自動構築をトリガすると、自動構築モジュール２３０は、式の少なくとも一部分を２次元形式に変換しようと試みる。例えば、解釈モジュール２２０が、式Ａの式Ａ’部分に対して自動構築をトリガすると、自動構築モジュール２３０は、以下の式Ｂ’によって示されるように、式Ａの式Ａ’部分を２次元形式に変換する。

【００４８】

【数２】

$$y = \frac{a}{b+c} \qquad (B')$$

【００４９】

以下の擬似コードは、自動構築モジュール２３０が、どのようにして式Ａ’から式Ｂ’を構築し得るかの一例である実施形態を示す。

【００５０】

【表３】

Push math italic "*a*" onto (rich-text) string stack
Push "/" onto operator stack
Push "(" onto operator stack
Push math italic "*b*" onto string stack
Push "+" onto operator stack
Push math italic "*c*" onto string stack
")" causes "b+c" to be concatenated on string stack
"Space" causes parenthesis "(" and ")" to be removed and fraction to be built up

【００５１】

一実施形態では、自動構築モジュール２３０は、コンピュータ言語コンパイラによって使われる分析と同様の、優先順位に依存する表現分析を用いて構築を実施する。その上、自動構築モジュール２３０は、式を所望通りにさらにフォーマットすることができる。例えば、式Ｂ’によって示されるように、自動構築モジュール２３０は、自動構築が起こると、括弧「（」および「）」ならびに左括弧「）」の後のスペースを削除することによって、式をフォーマットする。

【００５２】

表示モジュール２４０は、２次元形式で構築されている式の部分を、上述したＬＣＤディスプレイ１４２などの表示デバイスを使って、ユーザに対して表示する。表示デバイス２４０は、記入されているが、まだ線形文字列形式で構築されていない、式のどの部分も表示する。

【００５３】

式の一部分の自動的な構築の後、入力モジュール２１０は、線形文字列形式での式の入力を許可し続け、解釈モジュール２２０は、入力を解釈し続ける。例えば、式Ａに対して、式Ａ’が自動的に構築されると、ユーザは、式Ａの残りの部分を記入し続けることができる。式Ａの残りの部分（「＋ｄ」）に対してはそれ以上の構築が必要ないので、表示モ

ジュール２４０は、式Ａの残りの部分がシステム１００に入力された後、以下の式Ｂのように、式Ａ全体を２次元形式で表示する。

【００５４】

【数３】

$$y = \frac{a}{b + c} + d \qquad (B)$$

【００５５】

修正モジュール２５０は、ユーザが式を編集することを可能にする。ユーザは、式が線形文字列形式である間も２次元形式である間も、式を編集することができる。ユーザが式を編集するとき、解釈モジュール２２０は、式の追加部分が２次元形式に自動的に構築され得るかどうか判定するために、入力を監視し続ける。

【００５６】

自動構築を示す別の例が、以下の２次元形式で示される式Ｃによって与えられる。

【００５７】

【数４】

$$y = \frac{a}{b^2 + c^2} \qquad (C)$$

【００５８】

ユーザからの、線形文字列形式での式Ｃの入力が、入力モジュール２１０によって受諾されると、解釈モジュール２２０は、各文字を検査して、いつ自動構築をトリガするか判定する。ユーザが、式Ｃの分母にプラス符号「＋」を記入すると（すなわち、ユーザが「ｙ＝ａ／（ｂ＾２＋」を記入したとき）、解釈モジュール２２０は、自動構築モジュール２３０による、以下の式Ｃ’として示される、式Ｃの一部分の構築をトリガする。

ｙ＝ａ／（ｂ^2＋　　（Ｃ’）

【００５９】

具体的には、プラス符号「＋」の優先順位値が、上付き「＾」符号の優先順位値より小さいので、変数「ｂ」に対する上付きが構築される。上の表１を参照されたい。しかし、分母が明白に定義されていないので、分母全体はまだ構築されていない。

【００６０】

次に、ユーザが、右括弧「）」を記入する（すなわち、ユーザが、「ｙ＝ａ／（ｂ^2＋ｃ＾２」を記入する）と、変数「ｃ」に対する上付きが、以下の式Ｃ”に示されるように構築される。

ｙ＝ａ／（ｂ^2＋ｃ^2　　（Ｃ”）

【００６１】

最後に、ユーザが、右括弧「）」の後にスペースまたはキャリッジリターンを記入すると、式Ｃの分母が、上の式Ｃとして示されるように構築される。

【００６２】

ここで図３を参照すると、ユーザがコンピュータシステムに式を入力すると、式を自動的に構築する方法例３００が示されている。動作３１０で、ユーザは、線形文字列形式での式の入力を開始することを可能にされる。式の各文字が記入されると、制御は動作３２０に渡され、ここで、例えば、上述した優先順位構築論理を用いて記入された式の一部分に対して、自動構築が起こり得るかどうかに関して判定が行われる。自動構築が起こり得る場合、制御は動作３３０に渡される。自動構築が起こることができない場合、制御は、動作３１０に戻され、式の入力が続く。

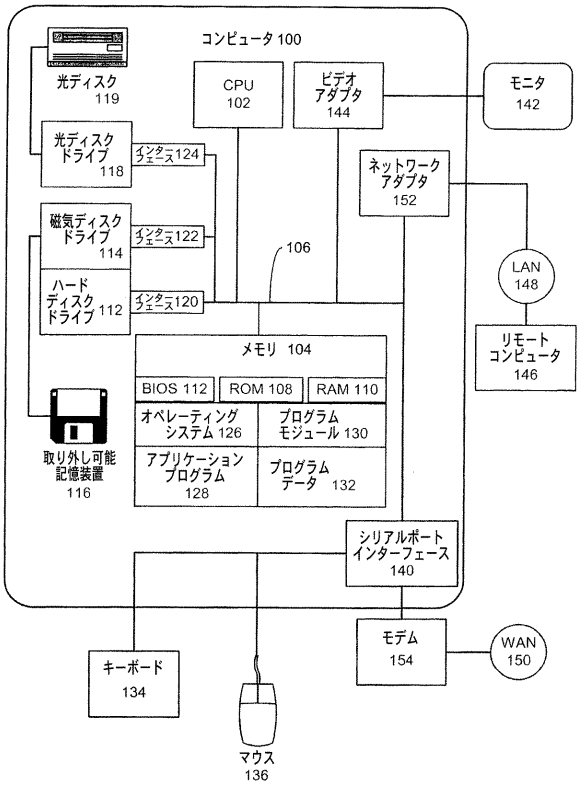
【００６３】

動作３３０で、式の一部分が、２次元形式で構築される。次に、動作３４０で、式の構築部分が、２次元形式で表示される。最後に、制御は、動作３１０に戻され、線形文字列形式での式の入力が続く。

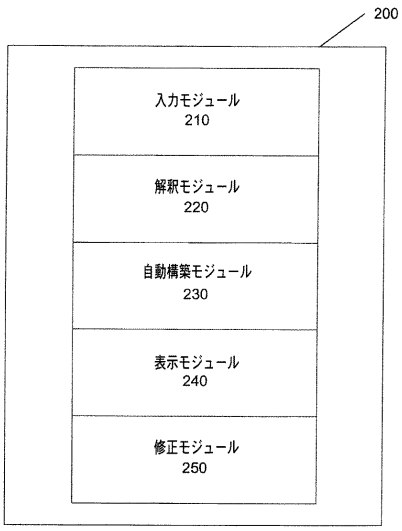
【００６４】
ここで図４を参照すると、式を自動的に構築する別の方法例４００が示されている。方法４００は、方法３００と類似しており、動作３１０、３２０、３３０、および３４０を含む。ただし、方法４００は、記入動作３１０と構築判定動作３２０の間に置かれた動作４１５も含む。動作４１５は、入力を検査して、構築が起こり得ることを合図する所定の文字を探す。このような文字の例は、右括弧「）」、バイナリ演算子、あらゆる句読点、タブ、エンター、およびスペースを含む。こうした文字のいずれが記入された場合でも、動作４１５は、自動構築が起こり得るかどうかに関する判定を可能にするために、動作３２０に制御を渡す。記入された文字が、上で列挙された文字の１つでない場合、動作４１５は、制御を動作３１０に戻し、式の一部分が構築され得るかどうかに関して判定せずに、線形文字列形式での式の入力が続く。
【００６５】
例えば、構築は、下付き文字「＿」が記入された後には起こることができない。というのは、ユーザは、式の下付き部分の構築が遂行され得る前に、下付き内容を記入する必要があるからである。したがって、下付き文字が動作３１０で記入された後で制御が動作４１５に渡されると、動作４１５は、構築が起こり得るかどうかに関して判定するために、制御を動作３２０に渡すのではなく、単に、制御を動作３１０に戻して、式の次の文字の記入を待機する。このようにして、自動構築の判定は、所定の文字が記入されたときのみ起こるので、式の記入および解釈が最適化され得る。
【００６６】
いくつかの実施形態では、本明細書で開示されるシステムおよび方法は、よく使われる式の部分が自動的に記入され構築されることを可能にすることによって、さらに改良される。例えば、ワシントン州レドモンドのマイクロソフトコーポレーションによって販売されている、Ｍｉｃｒｏｓｏｆｔ（登録商標）Ｗｏｒｄというアプリケーションは、ユーザが、いくつかの識別用文字を記入することによって、頻繁に使われるテキスト列を自動的に展開することを可能にする「オートコレクト」機能を含む。ユーザは、こうした機能を使って、ユーザによる数回のキーストロークに基づいて、よく使われる式部分を展開することができる。本明細書で開示されるシステムおよび方法は、「オートコレクト」機能によって展開される文字列を自動的に分析し、可能な場合は、文字列を自動的に構築するのに用いられ得る。このようにして、よく使われる式部分が、迅速に記入され構築され得る。
【００６７】
上述された様々な実施形態は、例示の目的でのみ与えられ、本発明を限定するものと解釈されるべきではない。本明細書において示され、記述された実施形態およびアプリケーション例に従うことなく、かつ添付の特許請求の範囲において述べられる本発明の真の精神および範囲から逸脱することなく、本発明に対して行われ得る様々な修正および変更が、当業者には容易に理解されよう。
【図面の簡単な説明】
【００６８】
【図１】本発明の一実施形態による汎用コンピューティングシステム例を示す図である。
【図２】本発明の一実施形態による、式の一部分を自動的に構築するシステム例を示す図である。
【図３】本発明の一実施形態による、式の一部分を自動的に構築する方法例を示すフローチャートである。
【図４】本発明の一実施形態による、式の一部分を自動的に構築する別の方法例を示すフローチャートである。
【符号の説明】
【００６９】
１００　コンピュータ
１０４　メモリ

１１２　ハードディスクドライブ
１１４　磁気ディスクドライブ
１１６　取り外し可能記憶装置
１１８　光ディスクドライブ
１１９　光ディスク
１２０　インターフェース
１２２　インターフェース
１２４　インターフェース
１２６　オペレーティングシステム
１２８　アプリケーションプログラム
１３０　プログラムモジュール
１３２　プログラムデータ
１３４　キーボード
１３６　マウス
１４０　シリアルポートインターフェース
１４２　モニタ
１４４　ビデオアダプタ
１４６　リモートコンピュータ
１５２　ネットワークアダプタ
１５４　モデム
２１０　入力モジュール
２２０　解釈モジュール
２３０　自動構築モジュール
２４０　表示モジュール
２５０　修正モジュール

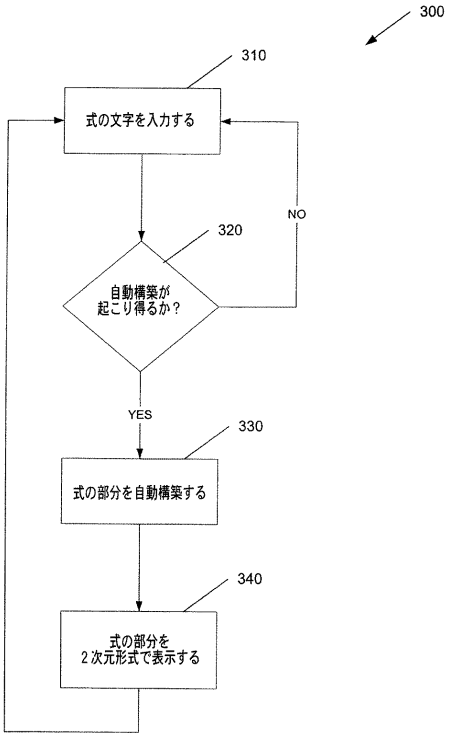
【図１】



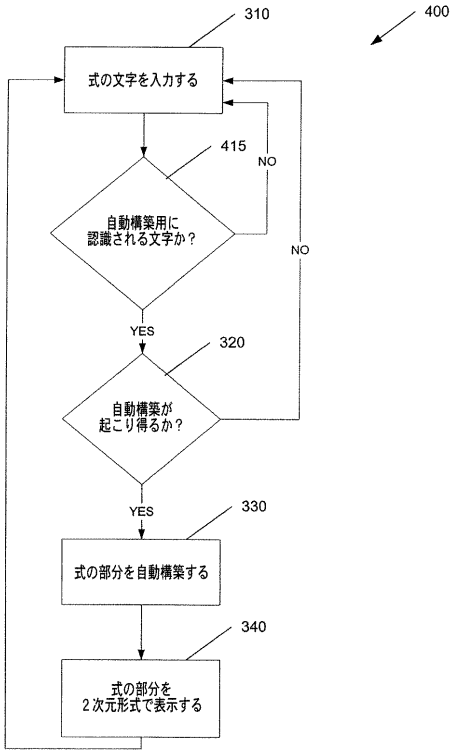
【図２】



【図３】


300
310
式の文字を入力する
320
自動構築が
起こり得るか？
NO
YES
330
式の部分を自動構築する
340
式の部分を
２次元形式で表示する

【図４】


400
310
式の文字を入力する
415
自動構築用に
認識される文字か？
NO
NO
YES
320
自動構築が
起こり得るか？
YES
330
式の部分を自動構築する
340
式の部分を
２次元形式で表示する

フロントページの続き

(72)発明者 イーサン ジョセフ バーンステイン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ジェニファー ピー. ミチェルステイン
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 マレー サージェント ザ サード
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 セイド アボウーハラワ
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

審査官 成瀬 博之

(56)参考文献 特開平１１－２５９４５６（ＪＰ，Ａ）
特開２０００－２９３５１７（ＪＰ，Ａ）
特開平０６－２３１０８１（ＪＰ，Ａ）
杉村 貴士，ＦｒｅｅＢＳＤを使いこなそう 使えるアプリケーションガイド ＯｐｅｎＯｆｆｉｃｅ．ｏｒｇ １．０．２ ＳｔａｒＳｕｉｔｅ ６．０，ＦｒｅｅＢＳＤ ＰＲＥＳＳ，日本，株式会社毎日コミュニケーションズ，２００３年 3月１８日，第１６号，ｐｐ．８９－９２

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １７／２０－１７／２６