数式処理 *J.JSSAC (2002)*
Vol. 9, No. 2, pp. 20 - 31

特集「2001 年度卒業論文・修士論文」

数学公式データベースとG関数

森永 昌義[*]　　村上 裕美[†]

愛媛大学大学院理工学研究科　　愛媛大学大学院理工学研究科

Abstract

数式処理のアルゴリズムの中で、記号積分に関する研究は重要な位置を占めている。しかし、積分可能な被積分関数の型は必ずしも多くはない。より広範な問題に数式処理を活用するためには、数学公式データベースを作成し、パターンマッチングを行うことにより記号積分を行う方法が必要になる。しかし、現在までの数式処理に関する研究では、 必ずしもデータベースとの結合は十分ではなく、たとえ結合されていても、特定の数式処理システムを強化するという目的で、ある種の数学公式データベースを作成していたもののみである。そこで、本研究では、数式処理システムの固有の数式表現や命令形態に依ることなく、数学公式データベースにより記号積分を行えるようなシステム作成を行うことを目標とする。具体的には、

- 数式処理システムに依存しない数学公式データベースの構築のために OpenMath 形式を採用
- 記号積分を実行するために、超幾何関数を一般化した Maijer の G 関数を実装し、公式の自動作成 · 検索

を実現した。多くの初等関数や特殊関数は Meijer の G 関数で表現することが可能であり、かつ、G 関数固有な演算によって、G 関数相互間の変換が可能である。そこで、数式処理システムを活用することによって、多くの公式を自動作成することが可能になる。ここでは、初等関数の記号積分に限定してシステムを作成する。

1 はじめに

数式処理のアルゴリズムの中で、記号積分に関する研究は極めて重要である。被積分関数の型を有理関数のみとしていた初期のアルゴリズムの拡張がはかられている。しかし、より広範な問題に数式処理を活用するためには、数学公式データベースを作成し、パターンマッチングを行うことにより記号積分を行う方法に頼らざるを得ない。現在までの数式処理に関する研究では、 必ずしもデータベースとの結合は十分には検討されてはいない。この種の研究として、数式処理システム GAL[8] を対象として、関係データベースの上で、数学公式の構造とインデキシング手法 [7] について述べたものが発表されているのみである。そこで、本研究

[*]morinaga@hpc.cs.ehime-u.ac.jp
[†]cumi@hpc.cs.ehime-u.ac.jp

では、数式処理システムの固有の数式表現や命令形態に依ることなく、数学公式データベースにより記号積分を行えるようなシステム作成を行うことを目標とする。このため、数式表現には、OpenMath 表現を採用した [3, 4, 5]。OpenMath による数式表現は XML(eXtensible Markup Language) 形式 [2] で表されており、XML の名前が、記号、演算子、関数名の数学オブジェクトである OpenMath オブジェクトに対応する。このような数学オブジェクトを木構造を用いて表現することによって、結果として公式の検索を容易にすることが出来る。記号積分を実行するために、超幾何関数を一般化した Maijer の G 関数 [1, 6] を実装し、公式の自動作成 · 検索を行う。

Meijer の G 関数は、

$$
\begin{aligned}
& G_{pq}^{mn}\left(z \,\middle|\, \begin{matrix} a_1, a_2, \ldots, a_p \\ b_1, b_2, \ldots, b_q \end{matrix}\right) \\
= \quad & \frac{1}{2\pi i} \oint_L \frac{\prod_{j=1}^{m} \Gamma(b_j + s) \prod_{j=1}^{n} \Gamma(1 - a_j - s)}{\prod_{j=n+1}^{p} \Gamma(a_j + s) \prod_{j=m+1}^{q} \Gamma(1 - b_j - s)} z^y dy
\end{aligned}
$$

で定義される。多くの初等関数や特殊関数は、G 関数で表現できる。また、G 関数固有な演算によって、G 関数相互間の変換が可能である。この技術を用い、任意の数式処理システムで計算することにより、G 関数を変形し、新しい数学公式を導出し、数学公式データベースに格納することが出来る。この操作によって大量の数学公式を生成することが可能になる。なお、本論では、数式処理システムとしては、Risa/Asir を用いた。本論で作成したシステムの公式の自動検索処理は

(a) XML 形式で書かれた入力式を解析し、定積分 · 不定積分などを判定

(b) 入力式を解析し、一般の表現の式から G 関数表現の式にするため検索キーを抽出

(c) 検索キーをもとにインデックス検索を行ない、公式番号を抽出

(d) 公式番号をもとに数学公式データベース検索を行ない、G 関数表現の式を検出

(e) 得られた G 関数表現の式に積分の処理を施し、新しい G 関数表現の式を決定

(f) 新しく得られた G 関数表現の式を解析し、検索キーを抽出

(g) 検索キーをもとにインデックス検索を行ない、公式番号を抽出

(h) 公式番号をもとに数学公式データベース検索を行ない、記号積分を出力

の順で処理を行う。

また、本システムの数学公式データベースの概念スキーマは、関係モデルとオブジェクトリレーショナルモデルを用いて設計されている。5 章で述べる数学公式を格納する関係には、関係モデルを用い、その数式の特性を格納する関係にはオブジェクトリレーショナルモデルを用いた。なお、この数式の特性化を行うには、XML の API(Application Program Interface) である DOM(Document Object Model) を用いて行っている。システムの記述は、数式処理

システムに依存しないこと等から Java を用いた。データベースシステムとしては、Java との親和性に優れた Oracle8i を採用した。

なお、以下、数学公式データベースと公式の分類についての概要を２章で与え、Meijer の G 関数と、それに対する Marichev の積分アルゴリズム及び G 関数に関するいくつかの公式を３章にまとめる。木構造を用いた数式の特性化を４章で述べる。実現したシステムの実際を５章で紹介する。

2 数学公式データベースと公式の分類

数式処理システムと関連づけて数学公式データベースを作成する研究としては佐々木等によるものがある [8, 7]。以下、これを基本として、数学公式を分類する。数学公式を、数式処理システムでの式の評価の際に、簡単化のために決定的な書き換え則として用いられる公式 (第１種の公式) と、非決定的な手順での式の変形則として用いられる公式 (第２種の公式) の二つに大別する。

- 第１種の公式は以下のような構造をしている。

$$O_p(F, \ldots) \to [\text{右辺}]$$

  ここで O_p とは $\int$ や $\sum$ 等の演算子を指す。"F" は"O_p" の主引数を指す。

- 第２種の公式は三角関数のように数式の簡単化に用いられるもので、以下のような構造をしている。

$$[\text{左辺}] = [\text{右辺}]$$

  このような公式は以下のような二つの異なった書き換え規則として用いられる。

$$[\text{左辺}] \to [\text{右辺}], \qquad [\text{左辺}] \leftarrow [\text{右辺}].$$

数学公式を特徴づけるために導入した、キーワードとキーワードのレベルという概念や、キーワードの次数、数式の全次数、数式の項数という数学公式の特徴量の導入に関しては、[7] の議論を踏襲する。すなわち、数学公式を格納するデータベースは以下のような項目間の関係によって設計される。

(a) 数学公式 (公式番号、左辺、右辺、条件)

(b) 構造 (公式番号、キーワード、キーワードのレベル)

(c) 性質 (公式番号、特徴記述子、状態)

(d) 名前 (公式番号、公式名)

ここで定義されているリレーションの中で、構造、性質、名前リレーションは実際の検索において中心的な役割を果たす。

2.1 公式のインデックス

論文 [8] の数学公式データベースでは、上述のリレーションの各属性値に基づき公式のインデックスを定義し導入している。公式のインデックスは、構造、性質、名前の各リレーションに対して 3 種類のものが定義される。ここでは第 1 種の公式についてのみ述べる。

1. 構造のインデックス
   (a) 公式を [左辺] → [右辺](条件) なる書き換え規則とみなす。公式が第 1 種の公式であるならば [左辺] $= O_p(F, \cdots)$ とする。
   (b) $\{f_1, f_2, \cdots, f_m\}$ を F 内の第 1 レベルのキーワードの集合とし、$\{g_1, g_2, \cdots, g_n\}$ を F 内の第 2 レベル集合とする。
2. 性質のインデックス
   (a) 与えられた数式 F において、F 内の第 1 レベルのキーワードの集合を $\{K_1, K_2, \cdots, K_n\}$ とする。
   (b) F を全次数 ($\#DEG$), 項数 ($\#TERM$) および F 内のキーワードの次数によって特徴づけられる。
   (c) 公式を [左辺] → [右辺](条件) なる書き換え規則とみなす。公式が第 1 種の公式であるならば [左辺] $= O_p(F, \cdots)$ とし、性質のインデックスを 2 で与えた特徴量によって定義する。
3. 名前のインデックス
   (a) 任意の公式 R において、R に特定の名前 $name_1, name_2, \cdots, name_k$ が付与されているならば R をこれらの名前によって特徴づける。

3 Meijer の G 関数

3.1 Meijer の G 関数と関数定理

Meijer の G 関数 [1, 6] は、超幾何関数 $F(\vec{a}; \vec{b}; z)$ を一般化したもので次式によって定義される。

$$
\begin{aligned}
& G^{mn}_{pq}\left(z \,\middle|\, \begin{array}{c} a_1, a_2, \ldots, a_p \\ b_1, b_2, \ldots, b_q \end{array} \right) \\
= \;& \frac{1}{2\pi i} \oint_L \frac{\prod_{j=1}^{m} \Gamma(b_j + s) \prod_{j=1}^{n} \Gamma(1 - a_j - s)}{\prod_{j=n+1}^{p} \Gamma(a_j + s) \prod_{j=m+1}^{q} \Gamma(1 - b_j - s)} z^y dy \qquad (1)
\end{aligned}
$$

積分路 L は $L_{\gamma+i\infty}$,L_∞,$L_{-\infty}$ の 3 つのいずれかになる。m, n, p, q は $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ の要素数を示している。通常の関数を G 関数で表現した例を以下に示す。

$$\begin{aligned}
\sin(x) &= \sqrt{\pi} G_{02}^{10}\left(\frac{x^2}{4}\,\middle|\,\begin{matrix}\cdot\ \cdot\\ \frac{1}{2},0\end{matrix}\right) && (2)\\
e^{-x} &= G_{01}^{10}\left(x\,\middle|\,\begin{matrix}\cdot\\ 0\end{matrix}\right) && (3)\\
J_\nu(x) &= G_{02}^{10}\left(\frac{x^2}{4}\,\middle|\,\begin{matrix}\cdot\ \ \cdot\\ \frac{\nu}{2},\frac{-\nu}{2}\end{matrix}\right) && (4)\\
{}_2F_1(a,b;c;-x) &= \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} G_{22}^{12}\left(x\,\middle|\,\begin{matrix}1-a,1-b\\ 0,1-c\end{matrix}\right) && (5)
\end{aligned}$$

Meijer の G 関数は次のような性質を持つ。これらを用いて数学公式データベースを作成する。

1. 定理 (Basic Properties)

$$\begin{aligned}
G(\mu,\vec{a};\vec{b};\vec{c};\mu,\vec{d};z) &= G(\vec{a};\vec{b};\vec{c};\vec{d};z) && (6)\\
G(\vec{a};\mu,\vec{b};\mu,\vec{c};\vec{d};z) &= G(\vec{a};\vec{b};\vec{c};\vec{d};z) && (7)\\
G(\vec{a};\vec{b};\vec{c};\vec{d};-z) &= G(1-\vec{a};1-\vec{b};1-\vec{c};1-\vec{d};z) && (8)\\
e^{tz}G(\vec{a};\vec{b};\vec{c};\vec{d};z) &= G(\vec{a}+t;\vec{b}+t;\vec{c}+t;\vec{d}+t;z) && (9)
\end{aligned}$$

2. 定理 (Duplication Formula)

$$\begin{aligned}
& (\vec{a};\vec{b};\vec{c};\vec{d};\frac{z}{k})\\
= \ & (2\pi)^{-(k-1)\beta/2}k^{1+\delta/2-\sigma}\\
\times \ & G \quad \Big(\triangle(\vec{a},k);\triangle(\vec{b},k);\triangle(\vec{c},k);\triangle(\vec{d},k);z+k\delta\log(k)\Big) && (10)
\end{aligned}$$

ここで、

$$\begin{aligned}
\triangle(\vec{a},k) &= \frac{\vec{a}}{k},\frac{\vec{a}+1}{k},\frac{\vec{a}+2}{k},\ldots,\frac{\vec{a}+k-1}{k}\\
\triangle(\vec{b},k) &= \frac{\vec{b}}{k},\frac{\vec{b}+1}{k},\frac{\vec{b}+2}{k},\ldots,\frac{\vec{b}+k-1}{k}\\
\triangle(\vec{c},k) &= \frac{\vec{c}}{k},\frac{\vec{c}+1}{k},\frac{\vec{c}+2}{k},\ldots,\frac{\vec{c}+k-1}{k}\\
\triangle(\vec{d},k) &= \frac{\vec{d}}{k},\frac{\vec{d}+1}{k},\frac{\vec{d}+2}{k},\ldots,\frac{\vec{d}+k-1}{k}
\end{aligned}$$

Marichev が以上の性質を用いて G 関数の定理を用いて G 関数の積分定理を論文 [1] で述べた。特殊関数の多くが G 関数として表現することができるので、これらの定理は一般的である。以下に、G 関数の積分定理を示す。

1. 不定積分

$$\int G(\vec{a};\vec{b};\vec{c};\vec{d};z)dz = G(1,\vec{a};\vec{b};\vec{c};0,\vec{d};z) \tag{11}$$

2. G 関数が一つ使われる場合

$$\int_0^\infty z^t G(\vec{a};\vec{b};\vec{c};\vec{d};\log(z)+v)dz = \frac{1}{u}e^{-\alpha v}\Gamma\left(\begin{array}{c} -\vec{a}+1-\alpha, \vec{c}+\alpha \\ \vec{b}+\alpha, -\vec{d}+1-\alpha \end{array}\right) \tag{12}$$

ここで、

$$\alpha = \frac{t+1}{u}$$

である。

$$\begin{aligned} & \int_0^y x^{\alpha-1} G_{pq}^{mn}\left(\omega x \left| \begin{array}{c} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{array}\right.\right) dx \\ = \ & y^\alpha G_{p+1,q+1}^{m,n+1}\left(\omega y \left| \begin{array}{c} a_1,\ldots,a_n,1-\alpha,a_{n+1},\ldots,a_p \\ b_1,\ldots,b_m,-\alpha,b_{m+1},\ldots,b_q \end{array}\right.\right) \end{aligned}$$

3. G 関数が二つ使われる場合

$$\int_0^\infty z^t G_1 G_2 dz = \frac{1}{u}e^{-\alpha v_2} G_3 \tag{13}$$

ここで、

$$\begin{aligned} G_1 &= G(\vec{a_1};\vec{b_1};\vec{c_1};\vec{d_1};u\log(z)+v_1) \\ G_2 &= G(\vec{a_2};\vec{b_2};\vec{c_2};\vec{d_2};u\log(z)+v_2) \\ G_3 &= G(\vec{a_1},-\vec{c_2}-\alpha+1;\vec{b_1},-\vec{d_2}-\alpha+1;\vec{c_1},-\vec{a_2}-\alpha+1;\vec{d_1},-\vec{b_2}-\alpha+1;v_1-v_2) \end{aligned}$$

$$\alpha = \frac{t+1}{u}$$

4. Cauchy の定積分

$$\int_0^\infty \frac{G(\vec{a};\vec{b};\vec{c};\vec{d};\log(z)+v)}{z-\mu}dz = -\pi\left(0,\vec{a};-\frac{1}{2},\vec{b};0,\vec{c};-\frac{1}{2},\vec{d};v+\log(\mu)\right) \tag{14}$$

3.2 公式の生成

Meijer の G 関数は次のような関係式によって新たに公式が生成できる。

3.2.1 Shift Operator

$D = \frac{\partial}{\partial z}$ とし、

$$\begin{aligned} A_i &= D + (-a_i + 1) \\ B_i &= -D + (b_i - 1) \\ C_i &= -D + c_i \\ D_i &= D - d_i \end{aligned}$$

とすると、

$$A_i G(\vec{a}; \vec{b}; \vec{c}; \vec{d}; z) = G(\vec{a} - \vec{e_i}; \vec{b}; \vec{c}; \vec{d}; z) \tag{15}$$

$$B_i G(\vec{a}; \vec{b}; \vec{c}; \vec{d}; z) = G(\vec{a}; \vec{b} - \vec{e_i}; \vec{c}; \vec{d}; z) \tag{16}$$

$$C_i G(\vec{a}; \vec{b}; \vec{c}; \vec{d}; z) = G(\vec{a}; \vec{b}; \vec{c} + \vec{e_i}; \vec{d}; z) \tag{17}$$

$$D_i G(\vec{a}; \vec{b}; \vec{c}; \vec{d}; z) = G(\vec{a}; \vec{b}; \vec{c}; \vec{d} + \vec{e_i}; z) \tag{18}$$

ここで、$\vec{e_i}$ は単位ベクトルである。shift Operator を用いた公式生成の例を以下に示す。

例 : $\sin(x) = G^{10}_{02}\left(\frac{x^2}{4} \middle| \begin{matrix} \cdot\ \cdot \\ \frac{1}{2}, 0 \end{matrix}\right)$ の C ベクトルについて shift Operator を用いることを考える。

$$\begin{aligned} G^{10}_{02}\left(\frac{x^2}{4} \middle| \begin{matrix} \cdot\ \cdot \\ \frac{3}{2}, 0 \end{matrix}\right) &= \left(\frac{x}{2}\frac{\partial}{\partial x} - \frac{1}{2} + 1\right) G^{10}_{02}\left(\frac{x^2}{4} \middle| \begin{matrix} \cdot\ \cdot \\ \frac{1}{2}, 0 \end{matrix}\right) \\ &= \left(\frac{x}{2}\frac{\partial}{\partial x} + \frac{1}{2}\right) \frac{\sin(x)}{\sqrt{\pi}} \\ &= \frac{1}{2\sqrt{\pi}}(-x\cos(x) + \sin(x)) \end{aligned}$$

これは、$\sin(x)$ に対応する G 関数表現の C ベクトルを 1 シフトした式を示している。

3.2.2 Contiguity Relation

$$\begin{aligned} & G(a_1 + 1; ; c_1, c_2; d_1; z) \\ = \; & -\frac{1}{(a_1 - d_1)(a_1 - c_2)(a_1 - c_1)} G(a_1 - 2; ; c_1, c_2; d_1; z) \\ & -\frac{3a_1 - c_1 - c_2 - d_1 - 3}{(a_1)(a_1 - c_2)(a_1 - c_1)} G(a_1 - 1; ; c_1, c_2; d_1; z) \\ & -(3a_1^2 - 2a_1c_1 - 2a_1c_2 - 2a_1d_1 + c_1c_2 + c_1d_1 + c_2d_1 - e^z - 3a_1 + c_1 + c_2 + d_1 + 1) \\ & \times (a_1 - d_1)^{-1}(a_1 - c_2)^{-1}(a_1 - c_1)^{-1} \times G(a_1; ; c_1, c_2; d_1; z) \end{aligned} \tag{19}$$

これらの関係式を用いると、ある G 関数 $G(\vec{a}; \vec{b}; \vec{c}; \vec{d}; z)$ の公式が与えられると新たに多くの公式を得ることができる。このような関係を利用して新たに公式を生成するアルゴリズムは、超幾何関数に対する公式生成アルゴリズムを G 関数に拡張することで得られている。

3.3 Marichev の積分アルゴリズム

これらの定理によって積分の求解を行う。G 関数を用いた積分の求解は、次の手順で行われる。

1. 被積分関数を G 関数表現にする
   式 (2)-(5), その他の式は論文 [1]、または式 (15)-(19) によって新たに得られた公式を参照し、G 関数表現の式に変形する。例えば $\int_0^y e^{-x}dx$ を求めたい場合、$\int_0^y G_{01}^{10}\left(x \,\middle|\, \begin{matrix} \cdot \\ 0 \end{matrix}\right) dx$ に変形する。

2. G 関数の積分の定理から求めたい積分の解を G 関数表現で求める
   これは上述した G 関数の定理を示した式 (11)-(14) を用いて行う。

3. 得られた G 関数表現の式を通常の関数表現に戻す
   ２で得られた式を式 (6)-(10) によって変形を行い、対応する通常の関数表現に変換する。

このアルゴリズムを数式処理システム REDUCE に実装する方法は論文 [1] で議論されている。１と３では、通常の関数表現と G 関数表現の関係を参照し、処理を行なっている。２については、積分公式の各引数に、実際の値を代入することで、解の G 関数表現を得ている。

4 数式の特性化

数学公式データベースを設計するにあたって、数学公式の特性化を行なわなければならない。本研究での数学公式データベースはプラットフォーム独立であることに重点をおいているため、数学公式の表現が、数式処理システム固有の表現に依存しないこと、効率的なインデックスを構成すること、数式の意味を保ったまま数学公式の通信が行なえること、という条件を満たさなければならない。このような条件を満たすために、数学的オブジェクトを表現するための標準である OpenMath[3, 4, 5] に従う。さらに、数学公式は OpenMath に従う XML 形式で表現されているので、XML の特性の中で、

- XML 文書がツリー構造で表現できること
- タグに囲まれたデータ情報がタグの名前として表現されていること

という特性を用いて、数学公式の特性化を行う。木構造から得られる特性は、どのような関数であるかという特性と、木構造自体から得られる特性として、深さ、葉の数、ノード数が得られる。また、G 関数表現に関する特性化であるが、G 関数の構造は

$$G\left(z \,\middle|\, \begin{matrix} \vec{a}, \vec{b} \\ \vec{c}, \vec{d} \end{matrix}\right) \tag{20}$$

であり、G 関数に関しては、式 (20) の G 関数のそれぞれの要素が特性となる。

- $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ の値そのもの

- $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ それぞれの要素数
- Z に対応する変数

これらをキーとしてデータベースを設計する。

5 システムの実際

4 章で述べた公式の特徴を基に、公式データベースのリレーションを以下のように設計した。

- 数学公式リレーション (公式番号, 一般の表現の式,G 関数表現の式)
- インデックスリレーション (公式番号, 深さ, 葉の数、OMS の数、シンボル名)
- G 関数リレーション (公式番号,A ベクトル,B ベクトル,C ベクトル,D ベクトル,Z)

数学公式リレーションでは公式番号を主キーとして検索を行なうことができるため、設計にはリレーショナルデータベースモデルに従った。インデックスリレーションおよび G 関数リレーションに関しては一つの公式に関して複数のキーワードが存在するため、オブジェクトリレーショナルモデルに従った。次に、公式運用部の流れを以下に示す。

図 1 で示されたそれぞれの動作について、以下で説明する。

(A) 入力式の解析
入力式は XML 形式で表されているので、DOM パーサを用いて木構造に変換して以下の解析を行なう。

- 不定積分であるか定積分を判定し、積分処理するためにどの定理が使えるかの判定を行なう。
- OMS ノードに対する変数名を抽出する。
- 木構造の特徴量を抽出する。

(B) インデックスリレーションによる検索
インデックスリレーションは XML 形式で格納する。これを木構造にしたものを用いて、検索を行なう。

```
<formura id="5001">
      <deep>4</deep>
      <leaf>7</leaf>
      <OMS>6</OMS>
      <Symbol>plus</Symbol>
      <Symbol>times</Symbol>
      <Symbol>sin</Symbol>
```

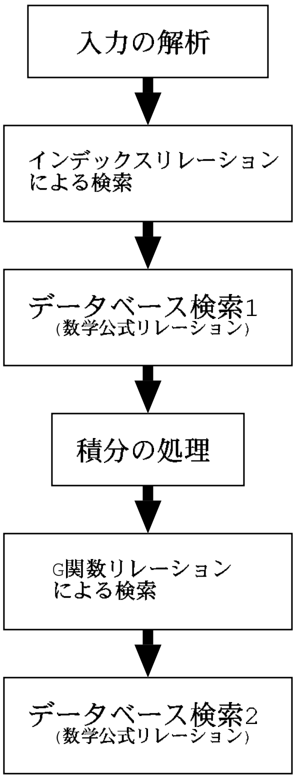

図 1: 公式運用部の流れ

```
    <Symbol>cos</Symbol>
</formula>
```

これは、

$$-\frac{1}{2}x\cos(x)+\frac{1}{2}\sin(x)=\sqrt{\pi}G^{10}_{02}\left(\frac{x^2}{4}\,\middle|\,\begin{array}{c}\cdot\ \cdot\\ \frac{3}{2},0\end{array}\right) \tag{21}$$

という公式の左辺に関する特徴量を示しているものである。このような公式に関するデータの集まりがインデックスリレーションには格納されている。このインデックスリレーションと入力式を解析することによって得られたキーをもとにパターンマッチングを行ない、それに対応する公式番号を抽出する。

(C) データベース検索 1

データベース検索は、Java 言語によるデータベース接続によって SQL を実行することで実現している。データアクセスインターフェースとして JDBC を利用している。(B)

で得られた公式番号をキーとしてデータベース検索を行ない XML 形式で書かれた G 関数表現の式を求める。

(D) 積分処理
(C) で得られた XML 形式で書かれた G 関数表現の式に積分の処理を行なうのだが、積分の処理は大別して不定積分と定積分がある。入力式を解析した際にどの積分であるか判定しているので、それによって、3 章で述べた対応する積分の処理を行なう。この処理によって、XML 形式で書かれた新たな G 関数表現の式を得る。

(E) G 関数リレーションによる検索
(B) と作業は同じであるが、G 関数表現の式から一般の表現の式に変換するために行なう。
　G 関数リレーションは XML 形式で格納する。これを木構造にしたものを用いて、検索を行なう。

```
<formula id="5001>
      <A name="vector_a"></A>
      <B name="vector_b"></B>
      <C name="vector_c"><OMI>3/2</OMI><OMI>0</OMI></C>
      <D name="vector_d"><OMI>0</OMI></D>
      <Meijer_Z>x^2/4</Meijer_Z>
</formula>
```

これは式 (21) の右辺の情報を記述したものである。このような公式に関するデータの集まりが G 関数リレーションには格納されている。この G 関数リレーションと (D) で得られた G 関数表現の式から得られる特徴量をもとにパターンマッチングを行ない、それに対応する公式番号を抽出する。

(F) データベース検索 2
(C) と作業は同様である。(E) で得られた公式番号をキーとして XML 形式で書かれた一般の表現の式を求める。

6 まとめ

　本研究では、数式処理システムに依存しない一部の数学公式データベースの構築と数学公式データベースを利用した積分の演算を行うシステムの構築を行った。
今後の課題としては

- 現在、対象とする関数が初等関数のみであるので、初等関数以外の数学公式データベースを増やすこと
- 各数式処理システムとつなぐシステムの構築

が挙げられる。

本論は愛媛大学工学部情報工学科での卒業論文 (平成１２年度 (村上)、平成１３年度 (森永)) を整理したものである。卒業論文作成時に御指導頂いた愛媛大学工学部 野田 松太郎教授、甲斐 博講師に感謝する。

参 考 文 献

[1] Adamchik,V.S. and Marichev,O.I., The Algorithm for calculating integrals of hypergeometric type functions and its realization in reduce system, proceedings of ISSAC '90, pp.212-221, 1990.

[2] Extensible Markup Language (XML)
http://www.w3.org/XML/

[3] The OpenMath Society
http://www.openmath.org

[4] The OpenMath Standard
http://www.nag.co.uk/projects/OpenMath/omstd

[5] OpenMath Core Content Dictionaries
http://www.nag.co.uk/projects/OpenMath/corecd

[6] Rozch,K., Meijer G Function Representations, proceedings of ISSAC '97, pp.205-211, 1997.

[7] 三枝義典、阿部昭博、佐々木建昭、増永良文、佐々木睦子：数式処理システム GAL における数学公式データベースのインデキシング手法、信学論 D-1, Vol.J74-D-I, pp.577-585, 1991.

[8] 佐々木建昭、増永良文、阿部昭博、元吉文男、三枝義典、佐々木睦子：数式処理システム GAL における数学公式データベース、第２９回プログラミングシンポジウム、1988.