計算の複雑さに関する構造的研究

藤澤 孝敏

北陸先端科学技術大学院大学 情報科学研究科

2000 年 2 月 15 日

キーワード： 構造的計算量理論, 計算の複雑さ, 多項式時間計算可能関数, function algebras, recursion schemes, Constable's class $\mathcal{K}$.

1 背景

現在のコンピュータが存在するはるか以前, 1930 年代に多くの数理論理学者たちにより「計算とは何か」, 「計算できるとはどういうことか」という問いに答えるための研究が行なわれた. 例えば, Turing のチューリング機械, Gödel と Herbrand の研究を基にした Kleene の帰納的関数, Church のλ-計算などである. これらの概念は一見すると全く別のものであるが, すべて本質的には同等なことが示されている. 同じ概念を別々の角度から見ていただけだったのである. こうして, 「原理的に計算可能な関数は帰納的関数であることにしよう」という, Church の提唱（もしくは Church-Turing の提唱）が誕生し, 計算可能性の概念が確立された.

しかしながら, 関数が原理的に計算可能だからと行って, 直ちに実際的な時間や資源で計算が可能とは限らない. いくつかの問題について, 原理的には計算可能であるが, 実際的には計算不可能であることが分かっている. そこで計算の複雑さの理論（計算量理論）は関数を計算するのに要する「コスト」や「難しさ」を理論的に明確にすることが目的である. 構造的計算量理論では, 計算量に応じてクラスを定義し, そのクラスの間の関係を明らかにする事が目的となる.

「実際的計算可能性」は現在のところ, 形式化された概念ではないが, 本研究では, 実際的計算可能な関数のクラスとは, どのようなクラスかを調べることを目的とする. その際, recursion schemes を用いて帰納的関数のように初期関数として, いくつかの関数を含み, ある演算に関して閉じている最小の関数の集合としてクラスを定義する方法を利用する. また, 実際的計算可能な関数のクラスとして, 多項式時間計算可能な関数のクラスと, Constable のクラス $\mathcal{K}$ について主に議論する.

2 多項式時間計算可能関数

多項式時間計算可能な関数のクラス (PTIME) は, 元々チューリング機械を用いて定義された. [5] によると, Cobham は, *bounded recursion on notation* (BRN) という原始帰納法の変形を用いて, 最初に機械モデルから独立な PTIME の特徴づけを行なった. 関数 fは, g, h_0, h_1, kから BRN により, 以下のように定義される. [1]

$$
\begin{aligned}
f(0,\vec{y}) &= g(\vec{y}), \\
f(s_0(x),\vec{y}) &= h_0(x,\vec{y},f(x,\vec{y})),\ x \neq 0 \text{ のとき} \\
f(s_1(x),\vec{y}) &= h_1(x,\vec{y},f(x,\vec{y}))
\end{aligned}
$$

さらに, すべての $x, \vec{y}$で $f(x,\vec{y}) \leq k(x,\vec{y})$ を満たしている. Cobham は, いくつかの初期関数を持ち合成と BRN に関して閉じている最小の関数のクラスを, PTIME と等価なクラスとして提案した. 原始帰納法では $f(x+1)$ を得るために $f(x)$ を使うので, $f(x)$ を計算するのに, およそ $2^{|x|}$ステップ必要となる. より小さいクラスを定義するために, Bennet が導入した, *recursion on notation* では, $f(x)$ を計算するの必要なスッテプ数は, $|x|$ となる. もちろん, BRN も recursion on notation の一種である.

Cobham の特徴づけは, 数多くの応用を生み出していった. しかし場合によっては不都合を引き起こしかねない要因も, いくつか持っていた. 最近, すべての $x, \vec{y}$で $f(x,\vec{y}) \leq k(x,\vec{y})$ を満たすという条件を持たない recursion scheme を用いて, PTIME を, 特徴付ける方法がいくつか知られてきている. 本論文では特に, 次の 2 種類を取り上げる. Bellantoni と Cook [1] は, $f(x_1,\ldots,x_n;y_1,\ldots,y_m)$ のように, 変数を位置に応じて区別する帰納法を考案した. セミコロンの左側に位置する変数 x_iを *normal* と呼び, 右側に位置する変数 y_iを *safe* と呼ぶ. 彼らが, 導入した *safe recursion on notation* (SRN)[2]により, fは, g, h_0, h_1から, 以下のように定義される.

$$
\begin{aligned}
f(0,\vec{y};\vec{a}) &= g(\vec{y};\vec{a}) \\
f(s_0(x),\vec{y};\vec{a}) &= h_0(x,\vec{y};\vec{a},f(x,\vec{y};\vec{a})),\ x \neq 0 \text{ のとき} \\
f(s_1(x),\vec{y};\vec{a}) &= h_1(x,\vec{y};\vec{a},f(x,\vec{y};\vec{a})).
\end{aligned}
$$

また, 石原 [6] は Clote [2] の *concatenation recursion on notation* から着想を得て *full concatenation recursion on notation* (FCRN) を考案した. $h_0(x,\vec{y},z), h_1(x,\vec{y},z) \leq 1$ として, fは FCRN より, g, h_0, h_1から以下のように定義される.

$$
\begin{aligned}
f(0,\vec{y}) &= g(\vec{y}) \\
f(s_0(x),\vec{y}) &= s_{h_0(x,\vec{y},f(x,\vec{y}))}(f(x,\vec{y})),\ x \neq 0 \text{ のとき} \\
f(s_1(x),\vec{y}) &= s_{h_1(x,\vec{y},f(x,\vec{y}))}(f(x,\vec{y})).
\end{aligned}
$$

[1] $s_0(x) = 2 \cdot x,\ s_1(x) = 2 \cdot x + 1$.

[2] [1] では, この recursion scheme は *predicative recursion on notation* と呼ばれてる.

多項式時間階層 (PH) も同じように, これらの recursion schemes を用いて特徴づけを行なうことができる.

3 Constable のクラス $\mathcal{K}$

Constable は, Kalmár の初等関数と多項式アナロジーを持つ関数のクラス $\mathcal{K}$ を定義した. ここで, 初等関数のクラス $\mathcal{E}$とは, 初期関数として 0, 射影関数, $x+1$, $+$, $\dot{-}$[3]を含み, 合成と *bounded summation* (BSUM) と *bounded product* (BPROD) に関して閉じている最小のクラスである. 関数 fは BSUM, BPROD により gから, それぞれ以下のようにして定義される.

$$f(x,\vec{y}) = \sum_{i=0}^{x} g(i,\vec{y}), \qquad f(x,\vec{y}) = \prod_{i=0}^{x} g(i,\vec{y}).$$

$\mathcal{E}$は指数関数を含むので実際的計算可能なクラスと, みなすことはできない. 一方, $\mathcal{K}$は, 初期関数として 0, 射影関数, $2\cdot x$, $2\cdot x+1$, $+$, $\dot{-}$, $\times$, $\lfloor x/y \rfloor$, を含み, 合成と *sharply bounded summation* (SBSUM) と *sharply bounded product* (SBPROD) に関して閉じている最小のクラスである. 関数 fは SBSUM, SBPROD により gから, それぞれ以下のようにして定義される.

$$f(x,\vec{y}) = \sum_{i=0}^{|x|} g(i,\vec{y}), \qquad f(x,\vec{y}) = \prod_{i=0}^{|x|} g(i,\vec{y}).$$

[3] で, Clote は以下のような $\mathcal{K}$と, 論理回路族による並列計算量クラスとの関係を示した.

$$\mathrm{TC}^0 \subseteq \mathcal{K} \subseteq \mathrm{NC}.$$

ここで, $k \geq 0$ で NC^kは LOGTIME-uniform で多項式 *size*, $O(\log^k n)$ *depth* で, さらにファンインが 2 に制限されたブール関数のゲート$\wedge$, $\vee$を持つ論理回路族によって受理される言語のクラスである. また, $\mathrm{NC} = \cup_k \mathrm{NC}^k$である. TC^0は, LOGTIME-uniform で, 多項式 *size*, *depth* は定数となり, threshold ゲートを持つ論理回路族によって受理される言語のクラスである. これらのクラスに対応する機械モデルから独立な特徴づけも TC^0は [4] で, NC は [2] で調べられている.

$\mathcal{K}$は, とても自然なクラスであるが, 直接対応する機械モデルにより定義されるクラスは, まだ分かっていない. そこで次のような問題が提起されている.

「$\mathcal{K}$に対応するのは, どのようなクラスか？」

Burtshick は, 多項式サイズで, ユニフォームな回路族と $\mathcal{K}$が関連を持っているのではないかと提案している. 石原は TC^0の初期関数に $x^{|y|}$を加えたクラスと $\mathcal{K}$との関連を示唆し, また TC^0は SBSUM に関して閉じていることを示した. そこで, 本研究では TC^0に $x^{|y|}$と $\lfloor x/y \rfloor$ を加えて新たなクラス $\mathcal{T}$を定義し, $\mathcal{K}$との同等性を調べる. 先ほどの議論より $\mathcal{T}$

[3] $x \dot{-} y$は, $x-y$が 0 以上のときは $x-y$になり, それ以外のときは 0 になる.

が, SBPROD に関して閉じていることが分かれば $\mathcal{T} = \mathcal{E}$となる. しかしながら, $\mathcal{T}$に x の２進数の長さを基にした２項係数$\binom{|x|}{y}$と階乗 $|x|!$が含まれていることは示すことができたが, 上述の問題を解決するには至らなかった. 今後の課題となろう.

参考文献

[1] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity,* 2:97-110, 1992.

[2] P. Clote. Sequential, machine-independent characterizations of the parallel complexity classes $ALOGTIME$,AC^k,NC^k and NC. In P. J. Scott, S. R. Buss, editor, *Feasible Mathematics,* pages 49-70, Birkhäuser, 1990.

[3] P. Clote. A Note on the Relation Between Polynomial Time Functionals and Constable's Class $\mathcal{K}$. In Kleine-Büning, editor, *Computer Science Logic.* Springer Lecture Notes in Computer Science, 1996. Result presented at CSL in Paderborn, 1995. To appear.

[4] P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In P. Clote and J. Remmel, editors, *Feasible Mathematics II,* pages 154-218, Birkhäuser, 1995.

[5] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel,editor, *Logic, Methodology and Philosophy of Science II,* pages 24-30, North-Holland, 1965.

[6] H. Ishihara. Function algebraic characterizations of the polytime functions. *Computational Complexity,* to appear, 1998.