数理解析研究所講究録
第 353 巻 1979 年 126-135

論理関数の複雑さと形式言語の関連について

京都大学工学部　安浦寛人
矢島脩三

1. まえがき　論理関数の複雑さを表わす尺度として，その関数を論理回路として実現する際の回路の段数や素子数，あるいはブール表現でその関数を表わした時のブール表現中の演算子数等が使われる．論理関数 f の計算量（Combinational Complexity）および計算時間（Deley Complexity）とは，それぞれ，与えられた数種類の基本的な論理素子を組み合せて，組合せ回路として f を実現する際に最低限必要となる素子数および回路の段数である．一方，論理関数 f の式量（Formula Size）とは，与えられた数種類の演算子と変数および定数を組み合せて，f を式として表現する時に最低限必要となる演算子数である．さまざまな論理関数に対し，入力変数の数と計算量，計算時間，式量との関係を一般的に議論するため，我々は，各 n に対してちょうど1個の n 変数関数を含む集合，"関数列" を導入した[1] 関数列は，$\{0,1\}$ 上の形式言語と1対1に

対応する．本稿では，この対応に基いて，関数列の計算量，計算時間，式量により，$\{0,1\}$ 上の言語のクラス分けを行い，それら各クラスの性質を調べる．さらに，言語理論の諸クラスとの関連について報告する．

2．論理関数の複雑さ　基底とは，論理関数の有限集合である．基底B上の組合せ回路とは，それぞれBに属する関数を実現する論理素子によって構成されるフィードバックループを含まない1出力の回路である．組合せ回路の段数とは，回路の入力から出力に至る経路のうちで，最も長い経路上の素子数である．任意の論理関数を基底B上の組合せ回路で実現できるとき，Bは完全であるという．

論理関数 f の基底B上における計算量 $C_B(f)$ とは，f を実現する基底B上の組合せ回路の中で，素子数が最も小さな回路の素子数である．論理関数 f の基底B上での計算時間 $D_B(f)$ とは，f を実現する基底B上の組合せ回路の中で，段数が最も小さな回路の段数である．論理関数 f の基底B上での式量 $F_B(f)$ とは，f を実現する基底B上の組合せ回路の中で，各素子のファンアウトを1に制限した場合に，素子数の最も小さな回路の素子数である．式量 $F_B(f)$ は，Bに属する関数を表わす演算子を使って f を式として表現する際に，最も演算

子数の小さな式の演算子数と定義することもできる．

論理関数 f に対する計算量，計算時間，式量は，基底の選び方によりその値が異る．しかし，完全基底の間では，計算量と計算時間は基底の違いにより高々定数倍しか変わらない[2]．一方，式量については，このような関係は成立しない[3]．以下，本稿では，完全基底 $B=\{xy, x+y, \bar{x}\}$ 上で考える．よって，計算量，計算時間に関するオーダの議論は他の基底においてもそのまま成立するが，式量については他の基底では使えないことに注意しておく．

3．関数列と形式言語　AND や OR，パリティ関数，加算，乗算等，多くの実用的な関数は，任意の入力数に対して定義されるが，それらは入力変数の数とは独立なある規則に従っていると考えられる．このような関数の集合を一般的に取扱うために，我々は各 n に対してちょうど一個の n 変数関数を含む関数列を導入した[1]．

関数列 $\{f_n\}$ を $\{f_n \mid f_n$ は n 変数関数，$n=1,2,\cdots\}$ で定義する．一般には，ある自然数 m に対して，m 変数関数を含まないような関数列も考えられるが，ここではこのような m に対しては，$f_m(x_1, x_2, \cdots, x_m) \equiv 0$ なる定数関数が列中に入っていると考える．

関数列 $\{f_n\}$ に対して，$\{0,1\}$ 上の形式言語 $L_{\{f_n\}}$ を

$$L_{\{f_n\}} = \{x_1 x_2 \cdots x_n \mid f_n(x_1, x_2, \cdots, x_n) = 1,\ f_n \in \{f_n\},\ n = 1, 2, \cdots\}$$

と定義する．逆に，$\{0,1\}$ 上の言語 L に対して，

$$f_n(x_1, x_2, \cdots, x_n) = \begin{cases} 0 & x_1 x_2 \cdots x_n \notin L \text{ のとき} \\ 1 & x_1 x_2 \cdots x_n \in L \text{ のとき} \end{cases}$$

によって関数列 $\{f_n\}_L$ を定義できる．これより，関数列 $\{f_n\}$ と $\{0,1\}$ 上の言語が 1 対 1 に対応することがわかる．

言語の上での各種操作と関数列上での論理関数的な操作との対応についてまとめておく．

〔性質 1〕 L_1, L_2 をそれぞれ $\{0,1\}$ 上の言語とする．L_1, L_2 に対応する関数列を $\{f_n\}$, $\{g_n\}$ と表わす．このとき，

$L_1 \cup L_2 = L_{\{f_n + g_n\}}$，　　$L_1 \cap L_2 = L_{\{f_n g_n\}}$，

$\overline{L_1} = L_{\{\overline{f_n}\}}$，　　$L_1^R = L_{\{f_n(x_n, x_{n-1}, \cdots, x_2, x_1)\}}$（反転），

$L_1 L_2 = L_{\{\sum_{i=0}^{n} f_i(x_1, x_2, \cdots, x_i) g_{n-i}(x_{i+1}, x_{i+2}, \cdots, x_n)\}}$（連接）但し Σ は論理和，

が成立する．

4．関数列の複雑さによる言語のクラス

$\{0,1\}$ 上の言語のクラス $\mathcal{L}_{D,\varphi(n)}$, $\mathcal{L}_{C,\varphi(n)}$, $\mathcal{L}_{F,\varphi(n)}$ をそれぞれ次のように定義する．

$$\mathcal{L}_{D,\varphi(n)} = \{L_{\{f_n\}} \mid {}^\forall f_n \in \{f_n\} \text{ に対し } D_B(f_n) = O(\varphi(n))\}$$

$$\mathcal{L}_{C,\varphi(n)} = \{L_{\{f_n\}} \mid {}^\forall f_n \in \{f_n\} \text{ に対し } C_B(f_n) = O(\varphi(n))\}$$

$\mathcal{L}_{F,\varphi(n)} = \{ L_{\{f_n\}} \mid {}^{\forall} f_n \in \{f_n\}$ に対し $F_B(f_n) = O(\varphi(n)) \}$.

ここに，基底 $B=\{xy, x+y, \bar{x}\}$ とする．特に，

$\mathcal{L}_{C,pol} = \bigcup_{p(n):多項式} L_{C,p(n)}$ ，　$\mathcal{L}_{F,pol} = \bigcup_{p(n):多項式} L_{F,p(n)}$，

で$\mathcal{L}_{C,pol}$と$\mathcal{L}_{F,pol}$を定義する．

〔性質2〕　$\psi(n) = O(\varphi(n))$ならば，

$\mathcal{L}_{D,\psi(n)} \subseteqq \mathcal{L}_{D,\varphi(n)}$，$\mathcal{L}_{C,\psi(n)} \subseteqq \mathcal{L}_{C,\varphi(n)}$，$\mathcal{L}_{F,\psi(n)} \subseteqq \mathcal{L}_{F,\varphi(n)}$

が成立する．

〔定理1〕[4] $\mathcal{L}_{D,n} = \mathcal{L}_{C,\frac{2^n}{n}} = \mathcal{L}_{F,\frac{2^n}{\log n}} = \{\{0,1\}$上のすべての言語$\}$

〔定理2〕[4] $\mathcal{L}_{D,\log n} = \mathcal{L}_{F,pol} \subseteqq \mathcal{L}_{C,pol}$.

真にn変数に依存する関数からなる関数列の計算時間，計算量，式量は，それぞれ，$\Omega(\log n)$，$\Omega(n)$，$\Omega(n)$となることが知られている[4]．このことから，意味のある最小のクラスとして，$\mathcal{L}_{D,\log n}$，$\mathcal{L}_{C,n}$，$\mathcal{L}_{F,n}$が重要である．また，定理2に示したように，$\mathcal{L}_{D,\log n}$との関連から，$\mathcal{L}_{C,pol}$，$\mathcal{L}_{F,pol}$も重要である．次にこれら5つのクラスのブール演算等の下での閉包性について述べる．

〔性質3〕　$\mathcal{L}_{D,\log n} = \mathcal{L}_{F,pol}$は，和集合，共通集合，補集合，連接，反転，Commutative Closure[注]，正規集合との共通部分をとる各演算について閉じている．

注．言語L中の各語wの記号の置換によってできる語のすべてからなる集合を，LのCommutative Closureという．

〔性質4〕 $\mathcal{L}_{C,n}$ は和集合，共通集合，補集合，反転，Commutative Closure，正規集合との共通部分をとる各演算について閉じている．

〔性質5〕 $\mathcal{L}_{F,n}$ は和集合，共通集合，補集合，反転については閉じており，Commutative Closure および正規集合との共通部分をとる演算については閉じていない．

〔性質6〕 $\mathcal{L}_{C,pol}$ は和集合，共通集合，補集合，連接，反転，Commutative Closure，正規集合との共通部分をとる各演算について閉じている．

$\mathcal{L}_{D,\log n}$，$\mathcal{L}_{C,n}$，$\mathcal{L}_{F,n}$ はすべて無限集合であるが，その濃度は，連続体濃度である．

〔性質7〕 $\mathcal{L}_{D,\log n}$，$\mathcal{L}_{C,n}$，$\mathcal{L}_{F,n}$ はすべて連続体濃度をもつ無限集合である．

（証明） $L_{\{f_n\}} \in \mathcal{L}_{D,\log n}$ とすると，$L_{\{\bar{f}_n\}} \in \mathcal{L}_{D,\log n}$ である．今，関数列 $\{g_n \mid g_n = f_n$ または $g_n = \bar{f}_n\}$ を考える．$L_{\{g_n\}} \in \mathcal{L}_{D,\log n}$ である． 各自然数 n に対し，$g_n = f_n$ ならば $p_n = 1$，$g_n = \bar{f}_n$ ならば $p_n = 0$ なる無限系列 $p_1 p_2 \cdots$ を $\{g_n\}$ に対応させる． $p_1 p_2 \cdots$ を2進小数とみなせば，区間 $(0, 1)$ の実数と $\{g_n\}$ が1対1に対応する．よって $\mathcal{L}_{D,\log n}$ は連続体濃度をもつ．$\mathcal{L}_{C,n}$ と $\mathcal{L}_{F,n}$ についても同様に証明できる．（証明終）

<u>5. 各種言語との関連</u> 我々は，有限記述のできる言語に対応する関数列に興味がある．本節では，言語理論でよく知られているいくつかの言語のクラスと，ここで定義したクラスとの関連について述べる．言語Lに対応する関数列$\{f_n\}_L$を計算する回路を構成するのに，Lを受理する受理機械が，$L\cap\{0,1\}^n$を受理する際の動作を論理回路でシミュレートする方法が考えられる．

Ungerは，決定性有限オートマトンを一次元に展開した組合せ回路を，木構造に再構成することにより，回路の段数を$O(\log n)$段，素子数を$O(n)$個とすることができることを示した[5]．

〔定理3〕[1] $\{0,1\}$上の正規集合Rに対応する関数列$\{f_n\}_R$に対して，段数$O(\log n)$段，素子数$O(n)$個の回路実現が存在する．

〔系1〕 正規集合のクラスは，$\mathcal{L}_{D,\log n}$，$\mathcal{L}_{C,n}$，$\mathcal{L}_{F,pol}$に含まれる．

FischerとPippenger，Schnorrらは，時間限定の決定性チューリング機械の動作をSimulateする組合せ回路の素子数について議論している[2]．

〔定理4〕[6] $\{0,1\}$上の言語Lが，決定性$T(n)$時間限定$S(n)$テープ限定チューリング機械で受理されるとき，$L\in\mathcal{L}_{C,T(n)\log S(n)}$

が成立する．ただし，nは入力語の長さであり，$T(n) \geqq n$である．

〔系2〕 決定性チューリング機械で，多項式時間で受理される$\{0,1\}$上の言語のクラスは$\mathcal{L}_{C,pol}$に含まれる．

一方，Borodinは，非決定性テープ限定チューリング機械の動作をシミュレートする組合せ回路の段数を議論している．

〔定理5〕[7] オフライン非決定性$S(n)$テープ限定チューリング機械で認識できる言語Lは$\mathcal{L}_{D,S(n)}$に入る．ただし，$S(n) \geqq \log n$．

文脈自由言語，決定性文脈自由言語のクラスは，定理4と定理5より次のクラスに含まれることがわかる．

〔定理6〕 文脈自由言語のクラスは，$\mathcal{L}_{D,(\log n)^4}$，$\mathcal{L}_{C,n^3 \log n}$ に含まれる．

〔定理7〕 決定性文脈自由言語のクラスは，$\mathcal{L}_{C,n\log n}$に含まれる．

Commutative Closureが自分自身に一致するような言語をCommutative言語とよぶ．Commutative言語のクラスは，連続体濃度をもち，帰納的可算でない言語をもふくむ．しかし，このクラスは対称関数との関係から重要である．すなわち，Commutative言語のクラスに対応する関数列のクラスは，対称関数の部分集合であるようなすべての関数列の集合と一致す

る．対称関数に対しては，段数 $O(\log n)$ 段，素子数 $O(n)$ 個となる回路構成法がある．[4] 式量についても $O(n^6)$ 以下となることが知られている．

〔定理8〕 Commutative 言語のクラスは，$\mathcal{L}_{D,\log n}$，$\mathcal{L}_{C,n}$ $\mathcal{L}_{F,n^6}$ に含まれる．

6．あとがき　最後に，重要と思われる未解決問題について述べておく．定理2に示したように，$\mathcal{L}_{C,pol}$ と $\mathcal{L}_{D,\log n}$ の間の包含関係について，$\mathcal{L}_{C,pol}$ に属して，$\mathcal{L}_{D,\log n}$ に属さないような言語あるいはそれに対応する関数列はまだ見つかっていない．n ビットの2進数の除算の問題は，素子数 $O(n \log n \log\log n)$，段数 $O((\log n)^2)$ で実現できることがわかっている．この問題は，$\mathcal{L}_{C,pol} \neq \mathcal{L}_{D,\log n}$ を示す例に使える可能性がある．逆に，$\mathcal{L}_{C,pol} = \mathcal{L}_{D,\log n}$ であれば，除算が $O(\log n)$ 段で実現できることになり，除算回路の高速化が可能となる．

このように関数列の問題は現実の回路構成の問題と密接に関係しており，回路の高速化，回路規模の最小化に対し，定理3で述べた構成法等はきわめて強力である．関数列の記述に形式言語を用いることにより，言語理論，オートマトン理論の諸結果を論理関数の回路実現に利用することができる．

謝辞　御討論頂いた本学上林彌彦助教授，稲垣耕作博士はじめ矢島研究室の諸氏に感謝します．

文献

(1)　安浦，矢島，"論理関数を実現するのに必要な論理回路の段数について"，信学技報 EC78-39，1978，11月．

(2)　Savage,J.E., "The Complexity of Compu ting," Wiley-Intersciences, 1976.

(3)　Pratt,V.R., "The Effect of Basis on Size of Boolean Expressions," 16th Annual Symposium on Foundations of Computer Sciences, Oct. 1975.

(4)　安浦，矢島，"論理関数の複雑さについて"，信学技報 AL78-73，1979 1月．

(5)　Unger,S.H. "Tree Realizations of Iterative Circuits," IEEE Trans. on Comput., vol.C-26,no.4,pp.365-383, April 1977.

(6)　Schnorr,C.P. "The Network Complexity and the Turing Machine Complexity of Finite Functions," Acta Informatika vol.7, pp.95-107, 1976.

(7)　Borodin,A. "On Relating Time and Space to Size and Depth," SIAM J.Comput. vol.6 no.4,pp.733-744, Dec.1977.