数理解析研究所講究録
1054 巻 1998 年 66-70

正則言語による論理関数の計算量解析

—— 群の上で動作するモノイドプログラムについて ——

戸田誠之助
日本大学文理学部応用数学科
〒156 東京都世田谷区桜上水 3–25–40
03-3329-1151 / toda@math.chs.nihon-u.ac.jp

あらまし 文献 [Bar89] において，Barrington は，段数 d の任意の論理回路が 5 次の交代群の上で動作する長さ 4^d のモノイドプログラムによって模倣できることを示した．さらに，この結果の拡張として，任意の非可解群 G に対しても同様の結果が成り立つことを示している．ただし，このときのモノイドプログラムの長さは 4^d ではなく，$(4|G|)^d$ になっている．本稿では，任意の非可解群についても 5 次の交代群の場合と全く同じ結果が成り立つことを述べる．さらに，群の「非ベキ零性」がモノイドプログラムの計算能力に関するある種の境界を示していることを述べる．

キーワード 計算量理論，オートマトン理論，正則言語，論理関数，群，モノイド

Complexity Analysis of Boolean Functions via Regular Languages

—— Some observations on M-Programs over Groups ——

Seinosuke TODA
Department of Applied Mathematics,
College of Humanities and Science, NIHON University
3–25–40 Sakurajyosui, Setagaya-ku, Tokyo 156
03-3329-1151 / toda@math.chs.nihon-u.ac.jp

Abstract In a seminal paper, Barrington [Bar89] showed a lovely result that a Boolean circuit of depth d can be simulated by an M-program of length at most 4^d working over the alternating group of degree five. He further showed that, for all nonsolvable groups G, a Boolean circuit of depth d can be simulated by an M-program of length at most $(4|G|)^d$ working over G. In this note, we improve the upper bound on the length from $(4|G|)^d$ to 4^d. We further observe that the "nonnilpotent" notion of groups precisely exhibits a boundary on whether M-programs can compute any Boolean functions.

keywords computational complexity theory, automaton theory, Boolean function, group, monoid

1. Preliminaries

We assume that the readers are familiar with Boolean circuits. We only note that our circuits consist of NOT-gates, AND-gates with fan-in two, OR-gates with fan-in two, and input gates with each of which a Boolean variable is associated. In this section, we first give the definition of M-programs over groups.

Definition 1.1. Let G be a group and n a positive integer. We define a *monoid-instruction*(an *M-instruction* for short) γ over G to be a three-tuple (i, a, b) where i is a positive integer, and both a and b are elements in G. We define an *monoid-program*(M-program for short) P over G to be a finite sequence $(i_1, a_1, b_1), (i_2, a_2, b_2), \ldots, (i_k, a_k, b_k)$ of M-instructions over G. For this M-program P, we call the number of M-instructions the *length of* P and denote it with $\ell(P)$. Furthermore, we call the maximum value among $i_1, i_2, \ldots, i_k$ the *input size of* P and denote it with $n(P)$.

We suppose any M-program P to compute a Boolean function in the following manner. Let n be the input size of P and let $\vec{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ be a vector of Boolean values that is given as an input to P. Then, we define the *value of an M-instruction* $\gamma_j = (i_j, a_j, b_j)$ in P, denoted by $\gamma_j(\vec{x})$, as follows:

$$\gamma_j(\vec{x}) = \begin{cases} a_j & \text{if } x_j = 0 \\ b_j & \text{if } x_j = 1 \end{cases}.$$

We further define the *value* $P(\vec{x})$ *of the M-program* P by $P(\vec{x}) = \gamma_1(\vec{x})\gamma_2(\vec{x})\cdots\gamma_k(\vec{x})$. Then we say that P *computes a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ if, for all $\vec{x} \in \{0,1\}^n$, if $f(\vec{x}) = 0$, then $P(\vec{x}) = e_G$, and otherwise, $P(\vec{x}) \neq e_G$, where e_G denotes the identity element of G. ♠

We further assume that the readers are familiar with elementary notions in group theory. Thus, we only give a breif definition for the notions of solvable/nonsolvable groups and nilpotent/nonnilpotent groups.

Definition 1.2. Let G be any finite group. For any two elements a, b of G, we define the *commutator* of a and b to be the element represented as $a^{-1}b^{-1}ab$ and denote it by $[a, b]$. We further define the *commutator subgroup* of G to be the subgroup of G generated by all commutators in G, and we denote it by $D(G)$.

Then, we inductively define $D_i(G)$, for all integers $i \geq 0$, as follows: $D_0(G) = G$, and for all $i \geq 1$, $D_i(G) = D(D_{i-1}(G))$. We say that G is *solvable* if $D_i(G) = \{e_G\}$ for some $i \geq 0$, where e_G denotes the identity element of G. If G is not solvable, we say that it is *nonsolvable.* It is easy to show that $D_{i+1}(G)$ is a subgroup of $D_i(G)$ for all $i \geq 0$. Hence, we see that, for all finite groups G, G is nonsolvable if and only if there exists a subgroup H such that $H \neq \{e_G\}$ and $H = D(H)$. We will use this fact later.

We further define $E_i(G)$ indeuctively as follows: $E_0(G) = G$, and for all $i \geq 1$, $E_i(G)$ is a subgroup of G that is generated by all elements in $\{[g, a] \ : \ g \in G, a \in E_{i-1}(G)\}$. We say that G is *nilpotent* if $E_i(G) = \{e_G\}$ for some $i \geq 0$, where e_G denotes the identity element of G. Otherwise, we say it to be *nonnilpotent.* It is obvious that $D_i(G)$ is a subset of $E_i(G)$ for all $i \geq 0$. Thus, we see that all nilpotent groups are solvable. ♠

2. On nonsolvable groups

To show our result, we use the following lemmas. The first lemma was implicitly used by Barrington in order to show that for all circuits C of depth d, the Boolean function computed by C can be computed by an M-program of length at most 4^d working over the alternating group of degree 5.

Lemma 2.1. Let G be a finite group and let e_G

be the identity element of G. Suppose that there exists a subset W of G satisfying the following two conditions:

(a) $W \neq \{e_G\}$, and

(b) for all elements $w \in W$, there are two elements $a, b \in W$ with $w = [a, b]$.

Then, for an arbitrary element $w \in W$ and all Boolean circuits C of depth d, there exists an M-program P_w over G that satisfies the conditions below.

(1) P_w is of length at most 4^d and is of the same input size as C.

(2) For all inputs $\vec{x} \in \{0,1\}^n$ where n is the input size of both C and P_w, $P_w(\vec{x}) = e_G$ if $C(\vec{x}) = 0$, and $P_w(\vec{x}) = w$ otherwise.

Proof. We show this lemma by an induction on the depth of a given circuit C. When the depth of C is 1 (that is, the Boolean function computed by C is either an identity function or its negation), it is obvious that an M-program consisting of single M-instruction computes the same function. Thus we have the lemma in this case.

Now assume, for some $d > 1$, that we have the lemma for all Boolean circuits of depth at most $d-1$ and all elements $w \in W$. Suppose further that C is of depth d, it is of input size n, and g is the output gate of C. We below consider three cases according to the type of the gate g.

Suppose g is a NOT-gate. Let h be a unique gate that gives an input value to g and let C_h denote the subcircuit of C whose output gate is h. Note that C_h is of depth at most $d-1$. Then, by inductive hypothesis, there exists an M-program Q_w that satisfies the following conditions.

(3) Q_w is of length at most 4^{d-1} and is of input size at most n.

(4) For all inputs $\vec{x} \in \{0,1\}^n$, $Q_w(\vec{x}) = e_G$ if $C_h(\vec{x}) = 0$, and $Q_w(\vec{x}) = w$ otherwise.

From this Q_w, we construct an M-program $Q_{w^{-1}}$ such that:

(5) $Q_{w^{-1}}$ is of length at most 4^{d-1} and is of input size atmost n, and

(6) for all inputs $\vec{x} \in \{0,1\}^n$, $Q_{w^{-1}}(\vec{x}) = e_G$ if $C_h(\vec{x}) = 0$, and $Q_{w^{-1}}(\vec{x}) = w^{-1}$ otherwise.

To construct $Q_{w^{-1}}$, we may first replace each M-instruction (i_j, a_j, b_j) by $(i_j, a_j^{-1}, b_j^{-1})$ and may further reverse the sequence of those M-instructions. Finally, we define P_w to be an M-program obtained from $Q_{w^{-1}}$ by replacing its first M-instruction, say (i_1, c_1, d_1), with (i_1, wc_1, wd_1). Then, we can easily see that P_w is of length at most 4^{d-1} and hence satisfies the conditions (1). We can further see that P_w satisfies the condition (2) above from its definition.

Suppose next that g is an AND-gate (with fan-in two). Let h_1 and h_2 are gates of C that give input values to g, and let C_1 and C_2 denote the subcircuits of C whose output gates are h_1 and h_2 respectively. Furthermore, let a and b be elements of W such that $w = [a, b]$. Note that C_1 and C_2 are of depth at most $d-1$. Then, by inductive hypothesis, we have two M-programs Q_a and Q_b such that:

(7) both Q_a qand Q_b are of length at most 4^{d-1} and they are of input size at most n, and

(8-1) for all inputs $\vec{x} \in \{0,1\}^n$, $Q_a(\vec{x}) = e_G$ if $C_1(\vec{x}) = 0$, and $Q_a(\vec{x}) = a$ otherwise, and

(8-2) for all inputs $\vec{x} \in \{0,1\}^n$, $Q_b(\vec{x}) = e_G$ if $C_2(\vec{x}) = 0$, and $Q_b(\vec{x}) = b$ otherwise.

Then, we define P_w by $P_w = Q_{a^{-1}}, Q_{b^{-1}}, Q_a, Q_b$, where $Q_{a^{-1}}$ and $Q_{b^{-1}}$ denote M-programs obtained from Q_a and Q_b, respectively, by using the same method as mentioned in the previous paragraph. It is not difficult to see that P_w satisfies the conditions (1) and (2) above. Thus we have the lemma in this case.

Suppose g is an OR-gate. In this case, we can obtain a desired M-program by using De Morgan's Law and the technique mentioned above.

We leave the detail to the reader. ♠

From this lemma, we may show that any finite nonsolvable group has a subset W satisfying the conditions (a) and (b) mentioned above. In fact, we will show that the conditions exactly charadcterize the nonsolvability of groups.

The following lemma is obtained by a simple calculation.

Lemma 2.2. Let G be any finite group and let a, b, c be any elements in G. Then, we have the following equations.

(1) $c^{-1}[a,b]c = [c^{-1}ac, c^{-1}bc]$.
(2) $[ab, c] = b^{-1}[a,c]b[b,c]$.
(3) $[a, bc] = [a,c]c^{-1}[a,b]c$. ♠

By using the above equations repeatedly, we can easily obtain the following lemma. We leave the detailed proof to the reader.

Lemma 2.3. Let G be any finite group, let V be a subset of G such that $V = \bigcup_{g\in G} g^{-1}Vg$, and let $a_1, \ldots, a_k$, $b_1, \ldots, b_m$ be any elements of V. Then, the commutator $[a_1 \cdots a_k, b_1 \cdots b_m]$ is represented as a product of commutators of elements in V. ♠

Lemma 2.4. For all finite groups G, G is nonsolvable if and only if G satisfies the conditions (a) and (b) mentioned in Lemma 2.1, that is, there exists a subset W of G such that:

(a) $W \neq \{e_G\}$ where e_G denotes the identity element of G, and
(b) for all elements $w \in W$, there are two elements $a, b \in W$ with $w = [a,b]$.

Proof. Suppose that there exists a subset W of G satisfing (a) and (b) above. Then, it is wasy to see, from (b) above and the definition of $D_i(G)$, that W is a subset of $D_i(G)$ for all $i \geq 0$. Combining this with (b) above, we have $D_i(G) \neq \{e_G\}$ for all $i \geq 0$. Hence G is nonsolvable.

Conversely, suppose that G is nonsolvable. Let H be a subgroup of G satisfying that $H \neq \{e_G\}$ and $H = D(H)$. Such a subgroup surely exists since G is nonsolvable. Furthermore, let S be a subset of H that generates H, and let us define U by $U = \bigcup_{g\in G} g^{-1}Sg$. Then, we inductively define a subset V_i of G, for all integers $i \geq 0$, as follows.

$$V_0 = U, \quad V_{i+1} = \{[a,b] \ : \ a, b \in V_i\} \quad (i \geq 0).$$

We below show, by induction on i, that for each $i \geq 0$,

(i) $V_i = \bigcup_{g\in G} g^{-1}V_i g$, and
(ii) V_i generates H.

From the definition of $U = V_0$, it is obvious that V_0 satisfies (i). Moreover, V_0 generates H since it includes all elements in $S = e_G^{-1}Se_G$. Assume V_i satisfies (i) and (ii). Since $H = D(H)$, each element h in H is represented as a product, say $[h_{1,1}, h_{1,2}][h_{2,1}, h_{2,2}] \cdots [h_{k,1}, h_{k,2}]$, of commutators of elements of H. Moreover, since V_i generates H, each $h_{i,j}$ is represented as a product of elements in V_i. Hence, the element h is represented as a product of elements of the form $[a_1 \cdots a_k, b_1 \ldots b_m]$ where each a_i and each b_i are elements in V_i. Then, from Lemma 2.3 and the inductive hypothesis that V_i generates H, we have that h is represented as a product of elements in V_{i+1}. Thus V_{i+1} generates H. From Lemma 2.2(1) and the inductive hypothesis, it follows that V_{i+1} satisfies the condition (i) above.

Since each V_i is a subset of G which is finite, there exists two integers $i, j \geq 0$ such that $i < j$ and $V_i = V_j$. Then, we define a desired set W by $W = \bigcup_{k=i}^{j-1} V_k$. Since $H \neq \{e_G\}$ and each V_i generates H, we have $W \neq \{e_G\}$. Moreover, from the definitions of each V_i and W, we see that for all $w \in W$, there are two elements a, b in W such that $w = [a,b]$. Thus we have the lemma. ♠

Combining Lemma 2.4 with Lemma 2.1, we immediately obtain the following theorem.

Thoerem 2.5. Let G be any finite nonsolvable group and C any circuit of depth d. Then, the

Boolean function computed by C is computed by an M-program over G of length at most 4^d. ♠

3. On nonnilpotent groups

It was shown in [BST90] that for all finite nilpotent groups G and some integer $n_G > 0$, no M-program over G can compute the conjunction of n Boolean variables for all $n \geq n_G$. Furthermore, it was shown in the same paper that for any finite nonnilpotent group G and all Boolean functions f, an M-program over G can compute f. These two results intuitively tell us that the "nonnilpotent" notion privides us with a boundary on whether M-programs over groups can compute any Boolean functions. We below observe this more precisely in a slightly strengthened form.

Theorem 3.1. Let G be any finite nonnilpotent group, let w be any element in G, and let f be any Boolean funtion with n input variables. Then, there exists an M-program P_w that computes f and is of length at most $3 \cdot 2^{2n-2} - 2^n$. ♠

4. Concluding Remarks

In [CL94], Cai and Lipton imporved Barrington's result on the alternating group of degree 5. They showed that any circuit of depth d can be simulated by an M-program over the group of length at most $2^{\lambda d}$ where $\lambda = 1.81\ldots$. However, it is unknown whether their result holds for all nonsolvable groups. They further showed a lower bound on the length of M-programs over groups: for any group G and any M-program P over G, if P computes the conjunction of n Boolean variables , then it must be of length at least $\Omega(n \log\log n)$. Hence, any M-program over any group simulating a circuit of depth d must have length asymptotically greater than 2^d.

In [Cle90], Cleve showed that for any constant $\varepsilon > 0$, a circuit of depth d can be simulated by a bounded-width branching program of length $2^{(1+\varepsilon)d}$. It would be interesting to ask whether the same result holds for M-programs over groups.

References

[Bar89] D.A.Barrington: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1, *J. of Computer and System Sciences* **38**, 150–164, 1989.

[BST90] D.A.Barrington, H.Straubing and D.Thérien: Non-uniform Automata over Groups, *Information and Computation* **89**, 109–132, 1990.

[BT88] D.A.Barrington and D.Thérien: Finite Monoids and Fine Structure of NC^1, *JACM* **35**, 1988.

[Cle90] R.Cleve: Towards optmal simulations of formulas by bounded-width branching programs, in Proc. of the 22th STOC, 271–277, 1990

[CL94] Jin-Yi Cai and R.J.Lipton: Subquadratic simulations of balanced formulae by branching programs, *SIAM J. on Computing* **23**, 563–572, 1994.

ST88 H.Straubing and D.Thérien: Finite Automata and Computational Complexity, *Lecture Notes in Computer Science* **386**, Springer-Verlag, 199–233, 1988.