

否定数限定回路の複雑さについて

西野 哲朗

田中 圭介

Tetsuro Nishino

Keisuke Tanaka

北陸先端科学技術大学院大学 情報科学研究科

1 はじめに

$B = \{0, 1\}$, $B_n = \{f | f : B^n \rightarrow B\}$ とする. Markov [5] は, すべての $F \subseteq B_n$ を計算するには, $b(n) = \lceil \log(n+1) \rceil$ 個の NOT ゲートが必要かつ十分であることを示した. また Fischer [3] は, 回路中で使用できる NOT ゲートの個数を $b(n)$ に制限したときの回路計算量 (否定数限定回路計算量) について考察し, すべての $F \subseteq B_n$ について, その否定数限定回路計算量は, 高々その組合せ計算量の 2 倍に $O(n^2(\log n)^2)$ の項を加えた値であることを示した.

本稿ではまず, n 変数の反転の否定数限定回路計算量について述べる. 次に, そこで得られた結果を用いて, 上記 Fischer の命題の上界が改良できることを示す. さらに, 回路中で使用できる NOT ゲートの個数を, 徐々に制限していったときの回路サイズの増加率についても考察を加える.

2 反転の複雑さ

本節では, n 変数を反転する否定数限定回路のサイズについて考える. 次の定理は, Jaikumar Radhakrishnan 氏が以前 King 氏 (当時 UCLA) から聞いたものである.¹

定理 1 n 変数を反転するには $\lceil \log(n+1) \rceil$ 個の NOT ゲートが必要である.

証明. l 個の NOT ゲートを含む回路 C が n 変数を反転するものとする. このとき $y_1, y_2, \dots, y_l$ をその l 個の NOT ゲートの出力とする (図 1).

ここでテスト入力 $T_i \in \{0, 1\}^n$ を以下のように定義する.

$$T_i = (\underbrace{1, \dots, 1}_i, \underbrace{0, \dots, 0}_{n-i}) \quad (0 \leq i \leq n)$$

すなわち,

$$T_0 = (0, 0, 0, \dots, 0)$$

$$T_1 = (1, 0, 0, \dots, 0)$$

¹King 氏の現在の所属は不明である.

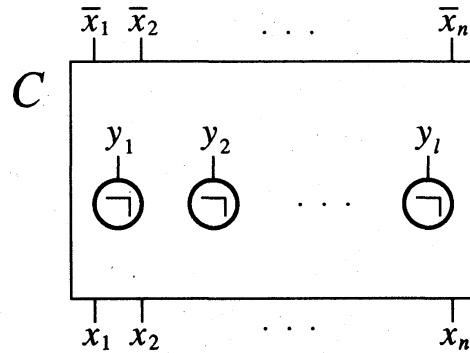


図 1: 回路 C

$$T_2 = (1, 1, 0, \dots, 0)$$

$$\vdots$$

$$T_n = (1, 1, 1, \dots, 1)$$

とする. また Y_i ($0 \leq i \leq n$) を以下のように定義する.

$$Y_i \stackrel{\text{def}}{=} (y_1(T_i), y_2(T_i), \dots, y_l(T_i)) \quad (1)$$

主張 $i \neq j$ ならば $Y_i \neq Y_j$ である.

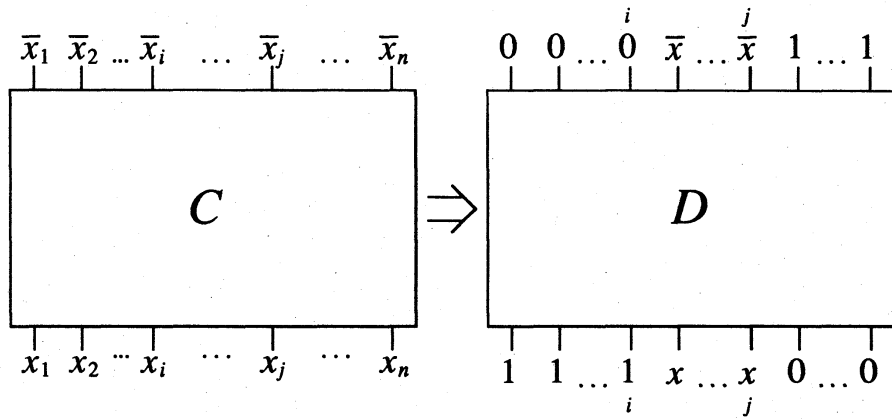
証明. 背理法による. ある i, j ($i < j$) に対して $Y_i = Y_j$ だったとする. このとき NOT ゲートを用いずに x を反転する回路が存在することを示し矛盾を導く. 回路 C の入力を $x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_n$ とし, 出力を $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_i, \dots, \bar{x}_j, \dots, \bar{x}_n$ とする. 回路 C から新たな回路 D を次のように構成する (図 2).

1. $x_1, \dots, x_i$ を 1 に固定する.
2. $x_{j+1}, \dots, x_n$ を 0 に固定する.
3. $x_{i+1}, \dots, x_j$ を x に置き換える.

このとき, 回路 D の構成法から, D は x を反転する回路となる (図 2 参照). また回路 D に対する入力は, $x = 0$ ならば T_i となり, $x = 1$ ならば T_j となる. ところが仮定より, T_i と T_j に対しては $Y_i = Y_j$ となるので, D 内のすべての NOT ゲートは, x に依存しない定数で置き換えることができる. $\square$

定理 1 の証明の続き. Y_i ($0 \leq i \leq n$) は全部で $(n+1)$ 個存在し, 上の主張よりそれらはすべて異なっていなければならない. 一方, 式 (1) の右辺のような形の組は 2^l 個存在しうるので,

$$2^l \geq n+1$$

図 2: 回路 C からの回路 D の構成

が成り立つ. よって以下を得る.

$$l \geq \lceil \log(n+1) \rceil \quad \square$$

以下では 組合せ回路 の 基底 を $\{\wedge, \vee, \neg\}$ とし, 単調回路 の基底を $\{\wedge, \vee\}$ とする. また, n 個の ブール変数 の集合 $\{x_1, x_2, \dots, x_n\}$ を X_n で表し, $\#_0(X_n)$ は X_n 中の 0 に固定された変数の個数, $\#_1(X_n)$ は X_n 中の 1 に固定された変数の個数を表すものとする.

n 変数の k -しきい値関数 $T_k^n(X_n)$ を次のように定義する.

$$T_k^n(X_n) = \begin{cases} 0 & \text{if } \#_1(X_n) < k, \\ 1 & \text{if } \#_1(X_n) \geq k. \end{cases}$$

以下では $b(n) = \lceil \log(n+1) \rceil$ とし, $C^{b(n)}(f)$ は否定の個数を $b(n)$ 個に限定したときの, 関数 f の回路計算量を表すものとする. また $V_n = \{\neg x_1, \neg x_2, \dots, \neg x_n\}$ と定義する. $C^{b(n)}(V_n) = O(n^2(\log n)^2)$ であることが知られているが [3], 次の定理は, [3] と [8] を組み合わせることによって, この上界が改善できることを示している.

定理 2 $C^{b(n)}(V_n) = O(n^2)$

証明. i ($0 \leq i \leq n$) に対して, $T_{k,i}^n(X_n)$ を次のように定義する.

$$T_{k,i}^n(X_n) = \begin{cases} 0 & \text{if } \#_1(X_n - \{x_i\}) < k, \\ 1 & \text{if } \#_1(X_n - \{x_i\}) \geq k. \end{cases}$$

ただし $T_{k,0}^n = T_k^n$ とする. そして, 以下のような事実を用いる.

$$\begin{aligned} \neg x_i = 1 &\Leftrightarrow x_i = 0 \\ &\Leftrightarrow \forall k [T_k^n(X_n) = T_{k,i}^n(X_n)] \\ &\Leftrightarrow \forall k [T_k^n(X_n) \rightarrow T_{k,i}^n(X_n)] \\ &\Leftrightarrow \bigwedge_{k=1}^n (\neg T_k^n(X_n) \vee T_{k,i}^n(X_n)) \end{aligned}$$

したがって, $\neg x_i$ は関数 $\neg T_k^n(X_n)$ ($1 \leq k \leq n$), $T_{k,i}^n(X_n)$ ($1 \leq k \leq n$), および $(n-1)$ 個の $\wedge$, n 個の $\vee$ によって計算できる. よって以下では, $U_n = \{\neg T_k^n(X_n) | 1 \leq k \leq n\}$, および $I_n = \{T_{k,i}^n(X_n) | 1 \leq k \leq n, 0 \leq i \leq n\}$ の計算について示す. I_n については [8] と類似の方法により, 以下を得る.

$$C^m(I_n) = \Theta(n^2)$$

ここで $Y_n = \{y_1, \dots, y_n\}$ とし, $y_i = T_i^n(X_n)$ ($1 \leq i \leq n$) と設定する. このとき $y_1, \dots, y_n$ は, $x_1, \dots, x_n$ に対する入力値を降順にソートした列となる. $\{T_i^n(X_n) | 1 \leq i \leq n\} \subseteq I_n$ であることより, Y_n は I_n によって計算できる.

U_n を計算するには, $\gamma_i^n(Y_n) = \neg T_i^n(Y_n)$ となるような関数の集合 $\Gamma_n = \{\gamma_1^n, \dots, \gamma_n^n\}$ を計算する回路 M_n を構成すれば十分である.

回路 M_n は, [3] と同様の手法により構成する. ここで $n = 2^r - 1$, $m = 2^{r-1}$, また $1 \leq j \leq m-1$ に対して,

$$\delta_j^n(y_1, \dots, y_n) = (y_j \wedge \neg y_m) \vee y_{m+j}$$

とし, Γ_n を r に対して以下のように再帰的に定義する.

$$i = 1: \gamma_1^1(y_1) = \neg y_1$$

$$i > 1:$$

$$\gamma_i^n(Y_n) = \begin{cases} \gamma_i^{m-1}(\delta_1^n(Y_n), \dots, \delta_{m-1}^n(Y_n)) \wedge \neg y_m & \text{if } 1 \leq i \leq m-1 \\ \neg y_m & \text{if } i = m \\ \gamma_{i-m}^{m-1}(\delta_1^n(Y_n), \dots, \delta_{m-1}^n(Y_n)) \vee \neg y_m & \text{if } m+1 \leq i \leq n \end{cases}$$

Γ_n を計算する回路 M_n は図 3 のようになる.

このとき確かに, M_n の出力は $\{\neg T_k^n(X_n) | 1 \leq k \leq n\}$ であり, M_n 内には NOT ゲートが $\lceil \log(n+1) \rceil$ 個だけ現れる. ここで, $S(n)$ を M_n 内で使用されたゲートの個数とすると, 以下を得る.

$$S(n) = 2n - 1 + S\left(\frac{n-1}{2}\right)$$

ゆえに,

$$S(n) \leq 4n - O(\log n)$$

よって, $C^{b(n)}(M_n) = O(n)$ であることより題意を得る. $\square$

次に $C^{b(n)}(V_n)$ の上界が, さらに改良できることを示す. 入力 $x_1, \dots, x_n$, 出力 $y_1, \dots, y_n$ の単調 n -sorter とは, 以下のような性質を満たす単調回路のことをいう.

$$y_i = T_i^n(X_n) \quad \text{for } 1 \leq i \leq n$$

また, 入力 $x_1, \dots, x_n$, 出力 $y_1, \dots, y_n$ の単調 (k, n) -inverter とは, 以下のような性質を満たす単調回路のことをいう.

$$y_i = \begin{cases} 0 & \text{for } 1 \leq i \leq n, \text{ if } \#_1(X_n) < k, \\ \bar{x}_k & \text{for } 1 \leq i \leq n, \text{ if } \#_1(X_n) = k, \\ 1 & \text{for } 1 \leq i \leq n, \text{ if } \#_1(X_n) > k. \end{cases}$$

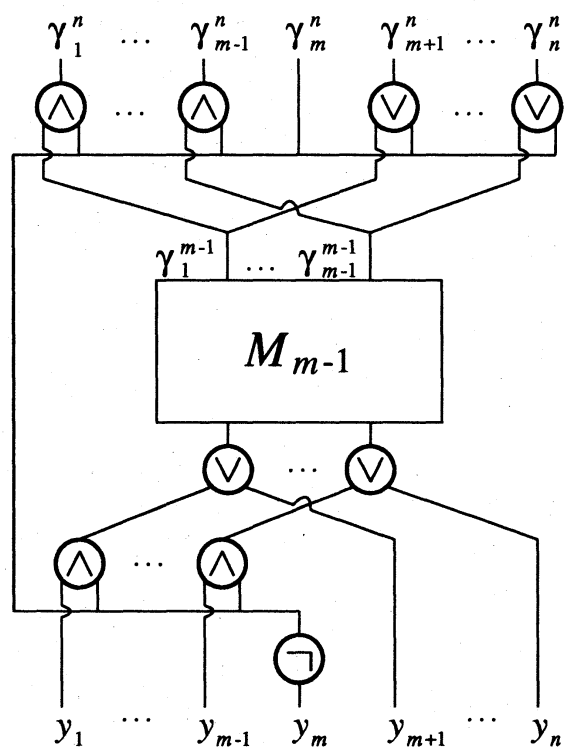


図 3: Γ_n を計算する回路 M_n

定理 3

$$C^{b(n)}(V_n) = O(n(\log n)^2)$$

証明. 実際に回路を構成することにより証明する. 回路は, 単調 n -sorter, 定理 2 の証明の中で用いた M_n と, 単調 $(n, 2n)$ -inverter の 3 つの部分から成る.

まず, n -sorter の入力を $x_1^S, \dots, x_n^S$, 出力を $y_1^S, \dots, y_n^S$ とする. n -sorter の各入力, 次のように固定する.

$$x_i^S = x_i \quad (1 \leq i \leq n)$$

次に, M_n の入力を $x_1^M, \dots, x_n^M$, 出力を $y_1^M, \dots, y_n^M$ とする. M_n の各入力, 次のように固定する.

$$x_i^M = y_i^S \quad (1 \leq i \leq n)$$

このとき定理 2 の証明より,

$$y_i^M = \neg T_i^n(X_n) \quad (1 \leq i \leq n)$$

となる.

最後に $(n, 2n)$ -inverter の入力を $x_1^I, \dots, x_{2n}^I$, 出力を $y_1^I, \dots, y_{2n}^I$ とする. $(n, 2n)$ -inverter の各入力, 次のように固定する.

$$x_i^I = \begin{cases} x_i & (1 \leq i \leq n), \\ y_i^M & (n+1 \leq i \leq 2n). \end{cases}$$

ここで構成した回路を図 4 に示す. 定理 4 の証明より次の関係が成り立つ.

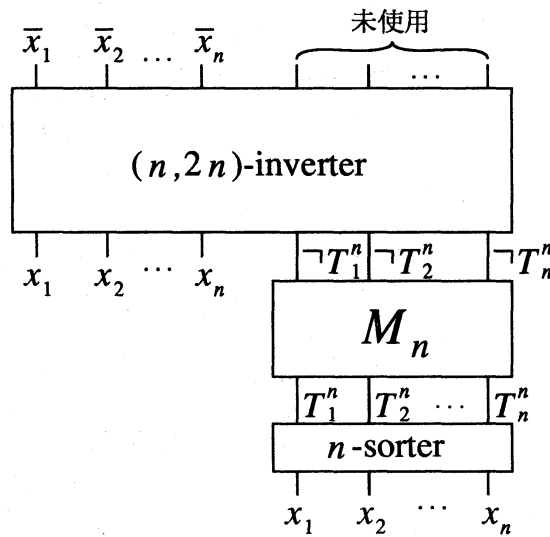


図 4: V_n を計算する回路

$$\#_1(M_n(X_n)) = \#_0(X_n)$$

また, 明らかに $\#_0(X_n) = n - \#_1(X_n)$ なので, $\#_1(M_n(X_n)) = n - \#_1(X_n)$ となる. よって以下を得る.

$$\begin{aligned} \#_1(x_1^I, \dots, x_{2n}^I) &= \#_1(x_1^I, \dots, x_n^I) + \#_1(x_{n+1}^I, \dots, x_{2n}^I) \\ &= \#_1(X_n) + \#_1(y_1^M, \dots, y_n^M) \\ &= \#_1(X_n) + \#_1(M_n(X_n)) \\ &= \#_1(X_n) + n - \#_1(X_n) \\ &= n \end{aligned}$$

よって, すべての i ($1 \leq i \leq 2n$) に対して $y_i^I = \overline{x_i^I}$ となり, 特に $y_i^I = \overline{x_i}$ ($0 \leq i \leq n$) となる (図 4 参照).

単調 n -sorter のサイズは $O(n \log n)$ であることが知られており [1], 単調 (k, n) -inverter のサイズは $O(n(\log n)^2)$ であることが知られている [7]. また定理 2 の証明から, $C^{b(n)}(M_n) = O(n)$ である. よって題意は示された.

□

3 否定数限定回路とその複雑さ

本節では, すべてのブール関数, および対称ブール関数の否定数限定回路計算量について考察する. まず, ブール関数 $f(X_n)$ に対して, f の k -スライス関数 $f_k(X_n)$ を次のように定義する.

$$f_k(X_n) = \begin{cases} 0 & \text{if } \#_1(X_n) < k, \\ f(X_n) & \text{if } \#_1(X_n) = k, \\ 1 & \text{if } \#_1(X_n) > k. \end{cases}$$

次の定理は [5] で証明された. 以下ではこの定理の別証明を与える.

定理 4 (A. A. Markov) 任意の $F \subseteq B_n$ は $\lceil \log(n+1) \rceil$ 個の NOT ゲートを用いれば計算できる.

定理 4 の別証明. 実際に回路を構成することにより証明する. 構成は混乱を避けるために, F が単元集合 (singleton set) の場合, すなわち $|F| = 1$ の場合についてのみ示すことにする.

関数 f の $(n+1)$ 個のスライス関数 $f_0, \dots, f_n$ と, n 個のしきい値関数の否定 $\neg T_1^n, \dots, \neg T_n^n$ が与えられれば,

$$f = \bigvee_{k=0}^n [\neg T_{k+1}^n \wedge f_k]$$

によって関数 f を計算することができる. ただし $\neg T_{n+1}^n$ は定数関数 1 とする.

定理の証明は, 以下の 2 段階から成る. (1) 関数 f を計算する回路から, スライス関数 f_k ($0 \leq k \leq n$) を計算する単調回路を構成し, (2) $U_n = \{\neg T_k^n | 1 \leq k \leq n\}$ を計算するような, NOT ゲートを $\lceil \log(n+1) \rceil$ 個だけ含む組合せ回路を構成する. (2) の構成は, 定理 3 の証明と同様に単調 n -sorter と M_n を組合わせて行なう.

よって以下では, (1) の構成を示す. そのためにまず, 関数 f を計算する組合せ回路 $K(x_1, \dots, x_n)$ を用い,

$$f_k = (f \wedge T_k^n) \vee T_{k+1}^n$$

によって関数 f_k を計算する組合せ回路 $K'(x_1, \dots, x_n)$ を構成する.

つぎに, $K'(x_1, \dots, x_n)$ から同じ関数を計算する standard circuit $K''(x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n)$ を構成する. 最後に [2] と同様の手法を用い, $K''(x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n)$ の各入力 $\bar{x}_i$ ($1 \leq i \leq n$) をその pseudo-complement $T_k^n(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ で置き換える. しきい値関数は単調回路で実現できるので, 以上のことより, f_k が単調回路で構成できることが示された.

ここで構成した回路を図 5 に示す. $|F| \geq 2$ の場合も, M_n は共通に使えるので, 上と同様の証明が適用できる.

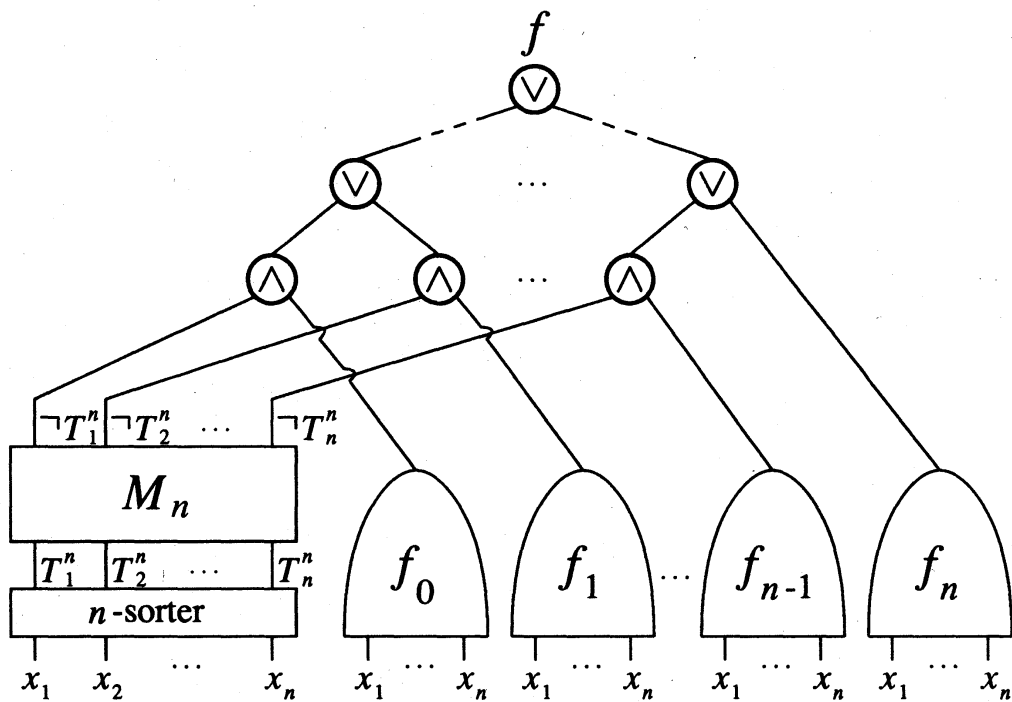


図 5: f_k により f を計算する回路

□

次に, 初めに述べた Fischer の命題の上界が改良できることを示す.

系 1 すべての $F \subseteq B_n$ に対して, $C^{b(n)}(F) \leq 2C(F) + O(n(\log n)^2)$

証明. f を実現する組合せ回路は standard circuit に変換でき, そのサイズは元の回路のサイズの高々 2 倍である. その standard circuit の否定の入力を V_n を実現する回路で置き換えれば, その回路は確かに f を計算し, しかも $b(n)$ 個の NOT ゲートしか含まない. よって定理 3 より題意を得る. □

特に定理 4 において, F が単元集合 (singleton set) の場合について考えると以下を得る.

系 2 任意の $f \in B_n$ に対して, $C^{b(n)}(f) \leq 2C(f) + O(n(\log n)^2)$ □

次の命題は定理 4 の別証明の系として得られる.

系 3 すべての $f \in B_n$ に対して, $C^{b(n)}(f) \leq C^m(f_0, \dots, f_n) + O(n \log n)$

証明. 定理 4 の別証明より明らか (図 5 参照). □

系 2 は $C(f) = O(n(\log n)^2)$ であるすべての $f \in B_n$ に対して, $C^{b(n)}(f) = O(n(\log n)^2)$ となることを示している. しかし系 3 を用いると, 対称関数についてはより強い上界が得られることを以下で示す. なお, S_n は n 変数の対称ブール関数のクラスを表すものとする.

系 4 すべての $f \in S_n$ に対して, $C^{b(n)}(f) = O(n \log n)$

証明. すべての $f \in S_n$ に対して, スライス関数 $f_0, \dots, f_n$ は以下のように定義される.

$$f_k(X_n) = \begin{cases} T_k^n(X_n) & \text{if } f(X_n) = 1 \text{ when } \#_1(X_n) = k, \\ T_{k+1}^n(X_n) & \text{otherwise.} \end{cases}$$

ただし T_{n+1}^n は定数関数 0 とする. ここで $\{T_1^n, T_2^n, \dots, T_n^n\}$ は単調 n -sorter で計算することになると, そのサイズは [1] より $O(n \log n)$ である. よって,

$$C^m(f_0, \dots, f_n) = O(n \log n)$$

より題意を得る. □

4 否定数の減少に対するサイズの増加率

本節では, すべてのブール関数, および, ある特定のブール関数について, その関数を計算する回路中で使用できる NOT ゲートの個数を, 徐々に制限していったときの回路サイズの増加率について述べる.

まず, n 変数の単調関数 f に対して, 使用可能な NOT ゲートの個数 t に関する関数 $K_f(t) = C^t(f)$ を考える. 定義から $K_f(0) = C^m(f)$, かつ $K_f(n) \leq 2C(f)$ である.

定理 5 すべての n 変数単調ブール関数 f について, 以下を満たす t ($0 \leq t \leq b(n) - 1$) が存在する.

$$\frac{K_f(t)}{K_f(t+1)} \geq \left(\frac{C^m(f)}{C^{b(n)}(f)} \right)^{\frac{1}{b(n)}}$$

証明. 背理法による. すべての t ($0 \leq t \leq b(n) - 1$) に対して,

$$\frac{K_f(t)}{K_f(t+1)} < \left(\frac{C^m(f)}{C^{b(n)}(f)} \right)^{\frac{1}{b(n)}} (= c)$$

だったとすると, 以下が成り立つ.

$$\begin{aligned} C^m(f) &= K_f(0) < cK_f(1) < c^2K_f(2) < \dots < c^{b(n)}K_f(b(n)) \\ &= \left(\left(\frac{C^m(f)}{C^{b(n)}(f)} \right)^{\frac{1}{b(n)}} \right)^{b(n)} C^{b(n)}(f) \\ &= \left(\frac{C^m(f)}{C^{b(n)}(f)} \right) C^{b(n)}(f) \\ &= C^m(f) \end{aligned}$$

これは矛盾である. よって題意は示された. $\square$

[6] において, 単調回路計算量の下界が $2^{cn^{1/6-o(1)}}$ (c は定数) である多項式時間計算可能単調ブール関数の存在が示された. すなわち, ある定数 c_1, c_2 が存在し, $C^m(f_0) \geq 2^{c_1 n^{1/6-o(1)}}$, および $C(f_0) = O(n^{c_2})$ を満たす n 変数関数 f_0 が存在する.

系 5 以下の関係を満たす, n 変数単調ブール関数 f , および t ($0 \leq t \leq b(n) - 1$) が存在する.

$$\frac{K_f(t)}{K_f(t+1)} = \exp(\Omega(n^{1/6-o(1)}))$$

証明. $f = f_0$ とする. 系 2 より,

$$\begin{aligned} C^{b(n)}(f) &\leq 2C(f) + O(n(\log n)^2) \\ &\leq O(n^{c_1}) + O(n(\log n)^2) \\ &\leq c_2 n^{c_1+1} \end{aligned}$$

よって定理 5 より, 以下の性質を満たす t ($0 \leq t \leq b(n) - 1$) が存在する.

$$\begin{aligned} \frac{K_f(t)}{K_f(t+1)} &\geq \left(\frac{2^{c_3 n^{1/6-o(1)}}}{c_2 n^{c_1+1}} \right)^{\frac{1}{b(n)}} \\ &= \exp \left(\frac{c_3 n^{1/6-o(1)} - (c_1 + 1) \log n - \log c_2}{\lceil \log(n+1) \rceil} \right) \\ &= \exp(\Omega(n^{1/6-o(1)})) \quad \square \end{aligned}$$

c ($0 < c < 1$) を定数とすると, cn -clique 関数は $\mathcal{NP}$ -完全であることが知られている [4].

定理 6 n を入力グラフの頂点数とし, $N = \frac{n(n-1)}{2}$ とする. このとき任意の定数 k ($k \geq 0$) に対し, $n\sqrt{1 - \frac{1}{2^{k+1}}}$ -clique 関数 f は, 十分大きな n に対して以下の性質を満たす.

$$C^{b(N)-k}(f) \leq \frac{N+1}{2^k} C(f) + O(N^2)$$

証明. 定理 4 の別証明と類似の方法で, f を計算する回路を構成する. ここで一般性を失うことなく, $N = 2^r - 1$ と仮定する. ただし, $r \geq 0$ は整数とする. このとき $b(N) - k = r - k$ 個の NOT ゲートで反転できる変数は $2^{r-k} - 1$ 個である. $c = 2^{r-k} - 1$ とする. 回路は, 単調 N -sorter, 定理 2 の証明中の M_c , および $(c+1)$ 個のスライス関数 $f_{N-c}, \dots, f_N$ を計算する単調回路から成る.

まず, N -sorter の入力を $x_1^S, \dots, x_N^S$, 出力を $y_1^S, \dots, y_N^S$ とする. N -sorter の各入力を次のように固定する.

$$x_i^S = x_i \quad (1 \leq i \leq N)$$

次に, M_c の入力を $x_1^M, \dots, x_c^M$, 出力を $y_1^M, \dots, y_c^M$ とする. M_c の各入力を次のように固定する.

$$x_i^M = y_{N-c+i}^S \quad (1 \leq i \leq c)$$

このとき定理 2 の証明より,

$$y_i^M = \neg T_{N-c+i}^N(X_N) \quad (1 \leq i \leq c)$$

となる.

ところで, $n\sqrt{1 - \frac{1}{2^{k+1}}}$ -clique は, $N - c$ 本より多くの辺を含む. なぜなら, $n\sqrt{1 - \frac{1}{2^{k+1}}}$ -clique をつくり得る最小の辺の数は,

$$\binom{n\sqrt{1 - \frac{1}{2^{k+1}}}}{2} = \left(\frac{1}{2} - \frac{1}{2^{k+2}}\right)n^2 + O(n)$$

であり, 一方,

$$\begin{aligned} N - c &= (2^r - 1) - (2^{r-k} - 1) \\ &= (N + 1)\left(1 - \frac{1}{2^k}\right) \\ &= \left(\frac{1}{2} - \frac{1}{2^{k+1}}\right)n^2 + O(n) \end{aligned}$$

である. ゆえに, 十分大きな n に対しては,

$$\binom{n\sqrt{1 - \frac{1}{2^{k+1}}}}{2} > N - c$$

となるからである. したがって, f は $\#_1(X_N) < N - c$ を満たす任意の X_N に対して $f(X_N) = 0$ となる. ゆえに,

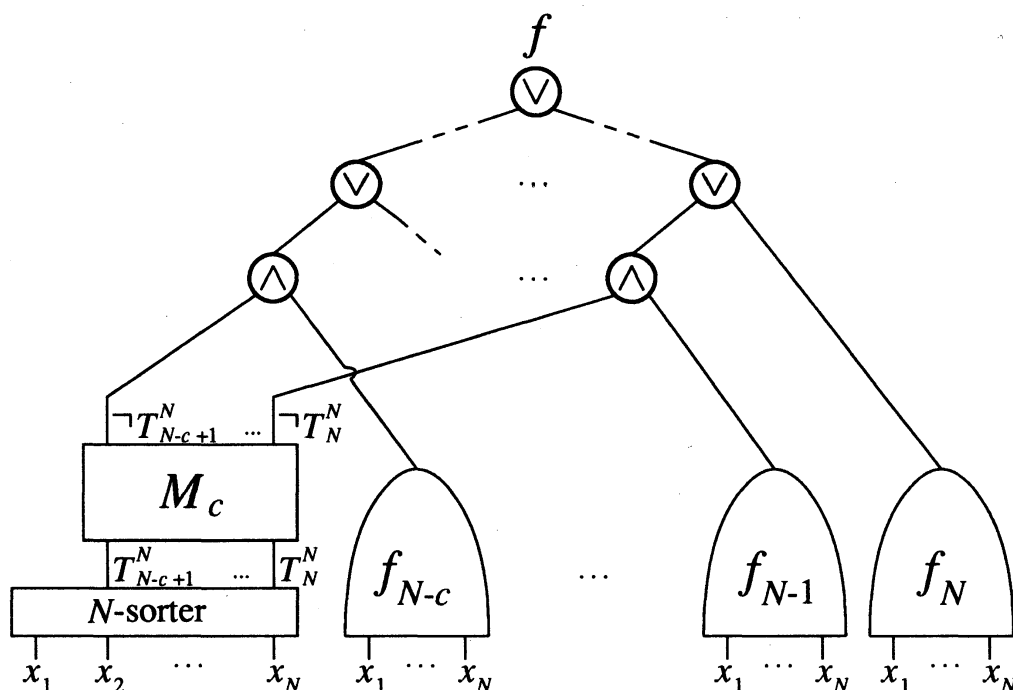
$$f = \bigvee_{i=N-c}^N [\neg T_{i+1}^N \wedge f_i]$$

によって f は計算できる (図 6 参照). ただし $\neg T_{N+1}^N$ は定数関数 1 とする.

スライス関数 $f_{N-c}, \dots, f_N$ を計算する単調回路のサイズは, [8] と類似の方法によって,

$$\begin{aligned} C^m(f_{N-c}, \dots, f_N) &\leq (c+1)C(f) + O(N^2) \\ &\leq \frac{N+1}{2^k}C(f) + O(N^2) \end{aligned}$$

となる. したがって題意は示された. $\square$

図 6: f を計算する回路

謝辞

反転の複雑さに関してコメントをくださった Jaikumar Radhakrishnan 氏 (北陸先端大学院) に感謝致します。

参考文献

- [1] M. Ajtai, J. Komlós, and E. Szemerédi, An $O(n \log n)$ sorting network, In *Proc. of the 15th Ann. ACM Symposium on Theory of Computing*, pages 1–9, 1983.
- [2] S. J. Berkowitz, On some relationships between monotone and non-monotone circuit complexity, Technical report, University of Toronto, 1982.
- [3] M. J. Fischer, The complexity of negation-limited networks—a brief survey, In *Lecture Notes in Computer Science 33*, pages 71–82. Springer-Verlag, Berlin, 1974.
- [4] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, 1979.
- [5] A. A. Markov, On the inversion complexity of a system of functions, *Journal of the ACM*, 5:331–334, 1958.
- [6] É. Tardos, The gap between monotone and non-monotone circuit complexity is exponential, *Combinatorica*, 7(4):141–142, 1987.

- [7] L. G. Valiant, Negation is powerless for Boolean slice functions, *SIAM Journal on Computing*, **15**(2):531–535, May 1986.
- [8] I. Wegener, More on the complexity of slice functions, *Theoretical Computer Science*, **43**:201–211, 1986.