A-001

TEP を解く Semantic Read-once Branching Program の超多項式下界の拡張
Extension for Super-polynomial Lower Bounds of Semantic Read-once BP Solving TEP

佐藤響樹 [*1]　長尾篤樹 [*1]
Hibiki Sato　Atsuki Nagao

1　はじめに

理論計算機科学分野の大きなテーマとして計算量クラスの分離問題が挙げられる．計算量クラス L が計算量クラス P に真に含まれるかを問う “L vs. P” 問題は計算時間と計算領域のトレードオフの観点からも非常に重要なテーマである．L $\neq$ P を示すための解析対象として Tree Evaluation Problem（TEP）が有力な候補として広く研究されており，先行研究にて三分木上の TEP を解く Semantic Read-once Branching Program の超多項式下界が証明されている．本研究ではこの結果をより拡張し，d を奇数として d 分木上の TEP を解く Semantic Read-once Branching Program においても超多項式下界が得られることを示す．

2　数学的準備

2.1　Branching Program

まず，本研究で用いる計算モデルの定義を紹介する．

Definition 2.1. *Nondeterministic Branching Program* (*NBP*) は，有向非閉路グラフで表現される計算モデルである．NBP は開始頂点と終了頂点を 1 つずつ持つ．終了頂点以外の頂点は入力変数をラベルとして持ち，出次辺は頂点が持つラベルの変数に割り当てられる値を持つ．NBP のサイズはグラフの頂点数で定義される．ここで，NBP による関数 $f : [k]^n \to \{0,1\}$ の計算は次のように行う．入力を受け取り，開始頂点から頂点の持つラベルの変数において入力に合致する有向辺を辿り次の頂点へと向かう．これにより終了頂点に到達するパスが 1 つでも存在すれば入力を受理し，存在しない場合は拒否をする．

ある問題が L に属さないことを示すには，その問題を解く任意の Branching Program の超多項式サイズ下界を示せば良いと知られている [1]．しかし，一般の Branching Program に対し $\Omega(n^2/\log^2 n)$ [2] より大きい下界は見つかっていない．そこで，NBP に制約を課して超多項式下界を得てから制約を緩和し，一般の下界へアプローチする手法が取られている．

Definition 2.2. *Semantic Read-once* NBP（以下、*S1NBP* とする）は，いずれかの受理される入力（Yes インスタンス）に合致する開始頂点から終了頂点へのパスすべてで，各入力変数が高々 1 度読まれる NBP である．

2.2　Tree Evaluation Problem

次に, 本研究で扱う問題である *Tree Evaluation Problem(TEP)* を紹介する．本研究では [3] による以下の完全 d 分木を持つ TEP の定義を用いる．

Definition 2.3. d 分木 TEP は, 高さ h，葉が d^{h-1} 個の完全 d 分木を持つ．葉には入力変数が割り当てられ，内部ノードには関数 $F : [k]^d \to [k]$ が割り当てられる．入力として d^{h-1} 個の葉へ割り当てる値を受け取り，根ノードの出力が閾値以内であれば受理，そうでなければ拒否を返す．

2.3　三分木 TEP の領域計算量下界

S1NBP を用いた既存研究 [3] では，高さ h の三分木の TEP が超多項式サイズ下界になることが証明されている．

Definition 2.4. $k \in \mathbb{N}, 0 < \epsilon < 1$ をパラメタとし，高さ h の三分木に対する TEP を以下で定義する．

入力：$n = 3^{h-1}$ 個の葉へ割り当てる値 $\vec{\xi} \in [k]^n$

出力：根ノードの出力が $k^{1-\epsilon}$ 以下ならば受理，そうでなければ拒否

Theorem 2.5. [3] 任意の h と十分大きい $k(k > 2^{42h})$ に対し, 三分木 TEP を解く任意の S1NBP がサイズ $\Omega(\frac{k}{n^{26}\log k})^h$ を要するような ϵ と内部関数 $\vec{F}$ の割り当てが存在する．

3　d 分木 TEP の領域計算量下界

本章では既存の三分木 TEP の証明手法を応用し，d を任意の奇数とした d 分木に対し超多項式サイズ下界が得られることを証明する．以降，d は任意の奇数とする．

Definition 3.1. $k \in \mathbb{N}, 0 < \epsilon < 1$ をパラメタとし，高さ h の d 分木に対する TEP を以下で定義する．

入力：$n = d^{h-1}$ 個の葉へ割り当てる値 $\vec{\xi} \in [k]^n$

出力：根ノードの出力が $k^{1-\epsilon}$ 以下ならば受理，そうでなければ拒否

Theorem 3.2. 任意の $h \geq 3$ と十分大きい $k(k > 2^{31h\log d})$ に対し，d 分木の TEP を解く任意の S1NBP がサイズ $\Omega(k^{\frac{1}{2}(d-1)(1-\frac{1}{h})}/n^{\frac{7}{2}(d+1)}\log k)^h$ となるような ϵ と内部関数 $\vec{F}$ の割り当てが存在する．

Proof. 本証明では d 分木の内部ノードに割り当てる関数として，$\boldsymbol{d+1}$ **可逆関数**を用いる．

Definition 3.3. 関数 $F : [k]^d \to [k]$ について，d 個の入力のうち $d-1$ 個と出力が与えられたとき，残り 1 つの入力が常に高々 $d+1$ 通りになるならば，F は $d+1$ 可逆であるという．

Yes インスタンスのみで構成される長方形のサイズを議論するため，Yes インスタンスが十分多く存在する TEP を扱うこととする．Lemma 3.4 により，内部ノードへの起こりうる関数割り当てに対し，ほとんどの場合において TEP が Yes インスタンスを十分多く持つことが示される．以降，F を $d+1$ 可逆関数 $[k]^d \to [k]$，$\vec{F}$ を TEP の内部関数を示す F のベクトルとし，F 上の一様分布を $\mathcal{F}$，$\vec{F}$ 上の一様分布を $\vec{\mathcal{F}}$ とする．ここで，パラメタ k と ϵ を調整することで以下の補題が保証できる．

Lemma 3.4. $\vec{\mathcal{F}}$ から無作為に抽出した $\vec{F}$ に対し Yes インスタンスの数が期待値の $\frac{1}{6}$ 以下になる事象を $Bad(\vec{F})$ とすると，$\Pr_{\vec{F}\sim\vec{\mathcal{F}}}[Bad(\vec{F})] \leq \frac{1}{10}$ である．

ここで，十分多くの Yes インスタンスを持つ状況のもとで Yes インスタンスの中に長方形が存在することを示し，長方形のサイズが十分大きいと仮定する流れを考える．

長方形の存在を示すため，各 Yes インスタンスについて対応する S1NBP 中の計算パスをそれぞれ 1 つに固定し，根から 1 つの葉に向かう**ラベル付パス** $\boldsymbol{P = v_h v_{h-1} \ldots v_1}$ と，葉 v_1 に対応する **S1NBP の頂点** $\boldsymbol{q}$ を得る．Lemma 3.5 は，任意の Yes インスタンスに対して P が持つ $\boldsymbol{Redtree(v_i)}$, $\boldsymbol{Whitetree(v_i)}$, $\boldsymbol{Thirdtree(v_i)}$ が特定の条件を満たすように P と q を得ることができると保証している．

Lemma 3.5. ある Yes インスタンスに対応する S1NBP の計算パスの中で，q 以前に読まれる変数を赤で彩色し，q よ

[*1] お茶の水女子大学 Ochanomizu University

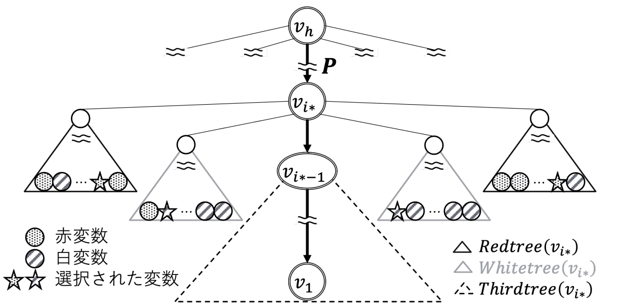

図 1　ラベル付パスと v_{i*} のイメージ

り後に読まれる変数を白で彩色する．このとき，以下を満たすような (P,q) が存在する（ただし，$2 \le i \le h$）．

- $\frac{1}{2}(d-1)$ 個の Redtree(v_i) それぞれが赤変数に対応する葉を $(\frac{d+1}{2})^{i-2}$ 個より多く持つ
- $\frac{1}{2}(d-1)$ 個の Whitetree(v_i) それぞれが白変数に対応する葉を $(\frac{d+1}{2})^{i-2}$ 個より多く持つ
- 1 個の Thirdtree(v_i) が赤・白変数に対応する葉をそれぞれ高々 $(\frac{d+1}{2})^{i-2}$ 個持つ
- Thirdtree(v_i) の根ノードが v_{i-1} になる

Lemma 3.5 により，すべての Yes インスタンスは一意に (P,q) を得る．最も多くの Yes インスタンスが示した (P,q) を考え，この (P,q) を示した Yes インスタンスのみを残し，その他を除外する．

次に v_i が持つ Redtree(v_i),Whitetree(v_i) において，最も多くの Yes インスタンスで赤, 白に彩色されている変数を各部分木から 1 つ選択し，赤のインデックス集合，白のインデックス集合にその変数のインデックスを加える．以降，すべての v_i が持つ部分木に前述の操作をして得られた赤・白のインデックス集合に含まれるインデックスに対応するすべての変数が赤, 白で彩色されている Yes インスタンスのみを扱う．

その後，どちらのインデックス集合にも含まれないインデックスを持つ変数の割り当てが 1 通りになるよう Yes インスタンスを刈り込むことで，Yes インスタンスの長方形が得られる．さらにパラメタ r, s を適切に設定すると，P に含まれるノードの中に以下に示す**特別な条件を満たすノード v_{i*}** が必ず存在することが示され，前述の長方形をさらに刈り込み $r \times r$ の長方形（正方形）を得ることができる．

Lemma 3.6. $\overline{Bad(\vec{F})}$ を満たす $\vec{F}$ を持つ d 分木 TEP を解くサイズ s の S1NBP を $\mathcal{B}$ とする．このとき，両方の大きさが $\frac{1}{2}(d-1)(h-1)$，サイズが $k^{(d-1)(h-1)-\epsilon}/6s2^{h^2(\frac{d-1}{2}+\log d)}$ 以上である埋め込み長方形が存在し，長方形が示すすべての入力が $\mathcal{B}$ に受理される．さらに，$r = (d+1)^{\frac{3}{2}(d+1)(h-2)}/\epsilon$，$s \le (k^{\frac{1}{2}(d+1)(1-\frac{1}{h})}/n^{\frac{7}{2}(d+1)}\log k)^h$ とすると，P に含まれるあるノード v_{i*} に対し．$\frac{1}{2}(d-1)$ 個の Redtree(v_{i*}) から 1 つずつ選択された赤変数による $\frac{1}{2}(d-1)$ 個組に r 通りの値割り当てをし，$\frac{1}{2}(d-1)$ 個の Whitetree(v_{i*}) から 1 つずつ選択された白変数による $\frac{1}{2}(d-1)$ 個組に r 通りの値割り当てをするように長方形を刈り込むことが可能であり，刈り込みにより $r \times r$ の長方形（正方形）が得られる．

内部関数が $d+1$ 可逆であることと，Lemma 3.6 で得た正方形に含まれる Yes インスタンスでは v_{i*} が持つ部分木から得た赤・白変数の $\frac{1}{2}(d-1)$ 個組への割り当てがすべて異なることを利用すると，v_{i*} が持つ関数 F_* の情報を F_* なしで，特別な条件を持つ $\mathcal{F}$ から得ることができる．

Lemma 3.7. $\overline{Bad(\vec{F})}$ を満たす $\vec{F}$ を持つ d 分木 TEP を解くサイズの小さい $\mathcal{B}$ が存在すると仮定する．このとき，高々 $(d+1)hr\log k$ ビットで記述可能なベクトル $L_{\vec{F}}$ が存在する．
$L_{\vec{F}}$ と，v_{i*} が持つ関数 F_* を除くすべての内部ノードの関数を記述したベクトル $\vec{F_*}$ を用いて v_{i*} への入力を推測し，それぞれの入力に対する出力の集合を推測する．推測した入力すべてに対し，推測した出力の集合に実際の出力が含まれている事象を E とすると，その確率は $\Pr_{F\sim\mathcal{F}}[E] \le \exp\left(-\frac{7}{9(d+1)^{(d-1)(h-2)}}\epsilon r^2 \log k\right)$ である．

ここで次のような TEP の符号化を考える．$\vec{\mathcal{F}}$ から無作為に $\vec{F}$ を選び，

1. $Bad(\vec{F})$ ならば各関数で $\mathcal{H}(\mathcal{F})$ ビット，全体で $\mathcal{H}(\vec{\mathcal{F}})$ ビットを使用して符号化する．
2. $Bad(\vec{F})$ でないならば $L_{\vec{F}}, \vec{F_*}$ と F_* を符号化する（ただし，F_* は一様分布 $(\mathcal{F} \mid E)$ からの要素）．

2. の符号化は 1. と比較すると $|L_{\vec{F}}|$ ビット余分に記述する一方，F_* が $(\mathcal{F} \mid E)$ からの要素であるため $\log \frac{1}{\Pr_{F\sim\mathcal{F}}[E]}$ ビットが節約できる．2. が可能なのは $\overline{Bad(\vec{F})}$ のときのみであるから，Lemma 3.4, 3.7 より全体で節約されるビット数は

$$
\begin{aligned}
&(1 - \Pr_{\vec{F}\sim\vec{\mathcal{F}}}[Bad(\vec{F})])\left(\log\frac{1}{\Pr_{F\sim\mathcal{F}}[E]} - |L_{\vec{F}}|\right) \\
&\ge \left(1-\frac{1}{10}\right)\left(\frac{7}{9(d+1)^{(d-1)(h-2)}}\epsilon r - (d+1)h\right) r \log k
\end{aligned}
$$

になる．ここで，$r = (d+1)^{\frac{3}{2}(d+1)(h-2)}/\epsilon$ から全ての $h \ge 3$ で $\frac{7}{9(d+1)^{(d-1)(h-2)}}\epsilon r - (d+1)h \ge 1$ が成立する．ゆえに

$$
\left(1-\frac{1}{10}\right)\left(\frac{7}{9(d+1)^{(d-1)(h-2)}}\epsilon r - (d+1)h\right) r \log k \ge r
$$

となるため，最適な符号化より $r > 1$ ビット以上短い符号化が存在してしまう．ゆえに，サイズの小さい S1NBP は存在しないことが示される．□

4　まとめと今後の課題

三分木 TEP を解く S1NBP の超多項式サイズ下界の証明を拡張し，一般の奇数 d に対する d 分木 TEP でも超多項式下界を得ることができた．Semantic Read-once でない一般の Branching Program に対しても超多項式下界が保たれるならば，$\mathsf{L} \subsetneq \mathsf{P}$ が証明できる．

一方で今回得られた下界を下回る上界が新たに示されている [4]．これにより制約を緩和すると下界は必ず下がると考えられる．TEP が L に属する可能性だけでなく，L = P である可能性も残されており，どちらの結論になるか確認するために，上界・下界両方からのアプローチが必要である．

参考文献

[1] Pavel Pudlak and Stanislav Zak. Space complexity of computations. *Preprint Univ. of Prague*, 1983.

[2] Edward I Nechiporuk. A boolean function. engl. transl. In *Sov. Phys. Dokl*, volume 10, pages 1–8, 1966.

[3] Venkatesh Medabalimi. *Branching Program Lower Bounds.* Doctoral Theses, University of Toronto (Canada), 2018.

[4] James Cook and Ian Mertz. Tree evaluation is in space. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC2024)*. ACM, 2024.