情報処理学会論文誌 Vol. 52 No. 4 1667–1677 (Apr. 2011)

推薦論文

非決定性回路族における深さと非決定性ゲート数の関係

岩本宙造[†1] 小野優介[†2]
森田憲一[†1] 今井克暢[†1]

言語を，計算の複雑さに基づいてクラス分けし，クラス間の包含関係を解明することは，理論計算機科学分野における最も基本的なテーマといえる．なかでも，P 対 NP 問題の証明は，特に重要であり，解明に向けて様々な側面から研究が行われている．クラス NC は，ゲート数が n の多項式，深さ（段数）が $\log n$ の多項式の対数一様論理回路族で受理される言語のクラスであり，$\mathrm{NNC}(g(n))$ は，NC に $g(n)$ 個の非決定性ゲートを導入したクラスである．$\mathrm{NNC}(g(n))$ は，$g(n)$ を大きくすることで，クラス NP と等しくなることが知られており，NC，P，NP 間のクラスの包含・等価関係を探求するのに適している．本稿では，非決定性回路族の深さと非決定性ゲート数の関係について調べる．ゲート数が n の多項式，深さが $\log n$ の多項式の任意の非決定性回路族は，非決定性ゲート数を多項式の範囲で増やすことで，深さを $O(\log n)$ まで小さくできることを証明する．また，非決定性チューリング機械を，効率良く模倣する非決定性回路族の深さとゲート数についても述べる．

Relationship between Depth and Nondeterministic Gates in Nondeterministic Circuit Families

Chuzo Iwamoto,[†1] Yusuke Ono,[†2] Kenichi Morita[†1]
and Katsunobu Imai[†1]

One of the most fundamental themes in theoretical computer science is to classify languages based on their computational complexity, and clarify the relationship among complexity classes. Especially, proving the “P vs NP problem” is the most important, and there have been a huge amount of literatures on this topic. NC is the class of languages accepted by logspace-uniform circuits of size polynomial and depth polylog, and $\mathrm{NNC}(g(n))$ is the class of languages accepted by NC-circuits with $g(n)$ nondeterministic gates. It is known that class $\mathrm{NNC}(g(n))$ changes from NC to NP as $g(n)$ grows, so $\mathrm{NNC}(g(n))$ is suitable for investigating the inclusion and equivalent relations among NC, P, and NP. In this paper, we study the relationship between depth and nondeterministic gates of NNC-circuits. It is shown that every nondeterministic circuit family of depth polylog and size polynomial can be simulated by a nondeterministic circuit family of depth $O(\log n)$ and size polynomial by increasing nondeterministic gates polynomially. We also investigate the depth and size of nondeterministic circuits which efficiently simulate nondeterministic Turing machines.

1. はじめに

言語を，計算の複雑さに基づいてクラス分けし，クラス間の包含関係を解明することは，理論計算機科学分野における最も基本的なテーマといえる．なかでも，$\mathrm{P} \neq \mathrm{NP}$ 予想の証明は，クレイ数学研究所のミレニアム懸賞問題の 1 つであり[1)]，解明に向けて様々な側面から研究が行われている．

クラス P に包含される代表的クラスとして NC があり，ゲート数（サイズ）が n の多項式，深さ（段数）が $\log n$ の多項式の対数一様論理回路族で受理される言語のクラスと定義される[8)]．クラス NC は，深さが $\log n$ の何乗の回路族で受理されるかという観点で，さらに分類されており，$\mathrm{NC}^1 \subseteq \mathrm{NC}^2 \subseteq \cdots \subseteq \mathrm{NC}^k \subseteq \cdots \subseteq \mathrm{NC} \subseteq \mathrm{P} \subseteq \mathrm{NP}$ という包含関係がある．これらのクラスでは，$\mathrm{P} \neq \mathrm{NP}$ だけでなく，$\mathrm{NC}^1 \neq \mathrm{NP}$ の証明さえも，未解決のままである．したがって，$\mathrm{P} \neq \mathrm{NP}$ 予想に対して，最初に証明すべき課題は，$\mathrm{NC}^1 \neq \mathrm{NP}$ であるとの指摘がある[7)]．

文献 9) は，クラス NC^k に，計算複雑さのクラスを分類するための計算資源の制約として，非決定性ゲートの数を導入した．$\mathrm{NNC}^k(g(n))$ は，$g(n)$ 個の非決定性ゲートを持つ NC^k 回路族で受理される言語のクラスである．この文献では，$g(n) = O(\log n)$ のとき，$\mathrm{NNC}^k(g(n)) = \mathrm{NC}^k$ であること，$g(n)$ が n の多項式のときは，$\mathrm{NNC}(g(n)) = \mathrm{NP}$ であることなどを証明している．また，$g(n)$ が $\log n$ の多項式のときに，$\mathrm{NNC}^k(g(n))$ で特徴づけられる具体的問題を提案している．この文献では，$\mathrm{NC}^1 \neq \mathrm{NP}$ 予想の証明には至っていないが，$\mathrm{NC}^k \subseteq \mathrm{P} \subseteq \mathrm{NP}$ 間のクラスの包含・等価関係にいくつかの知見を与えた．

†1 広島大学工学研究院
Graduate School of Engineering, Hiroshima University
†2 シャープ株式会社
Sharp Corporation
本稿の内容は 2009 年 10 月の電気・情報関連学会中国支部連合大会にて報告され，中国支部長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

本稿では，非決定性回路族の深さと非決定性ゲート数の関係について調べる．ゲート数が n の多項式，深さが $\log n$ の多項式の任意の非決定性回路族は，非決定性ゲート数を多項式の範囲で増やすことで，深さを $O(\log n)$ まで小さくできることを証明する．この結果から，$\mathrm{NNC}^1(\mathrm{poly}) = \mathrm{NNC}(\mathrm{poly})$ が導かれる．また，非決定性チューリング機械を，効率良く模倣する非決定性回路族の深さとゲート数についても述べる．$g(n)$ 回の非決定性動作を行う $t(n)$ 時間 $s(n)$ 領域 l テープ非決定性チューリング機械を模倣する対数一様論理回路族で，深さが $O(t(n)\log s(n))$，ゲート数が $O(t(n)s(n))$，非決定性ゲート数が $g(n)$ のもの，および，深さが $O(t(n))$，ゲート数が $O(t(n)(s(n))^{l+2})$，非決定性ゲート数が $g(n)$ のものを設計する．

非決定性ゲートを含まない論理回路族のゲート数と深さに基づく階層定理は，文献 4)，5) に与えられている．また，様々な一様性条件の下で，論理回路族の階層定理を示した文献として 3) がある．さらに，文献 6) は，階層定理のサーベイである．

次章で，定義と結果を与える．定理の証明は，3，4，5 章で与える．

2. 定義と結果

本稿では，読者はチューリング機械（Turing Machine，TM）について，基本的な知識があることを前提とする．TM の詳しい定義については，文献 2) を見よ．以下の一様論理回路族の定義は，主として，文献 8) を参考にした．

論理回路は，有向非巡回グラフであり，その各ノード（ゲート）の入次数は 0〜2 である．入次数 2 のゲートは，ブール関数 AND または OR でラベル付けされており，入次数 1 のゲートは，ブール関数 NOT でラベル付けされている．入次数 0 のゲートは，入力ゲートで，入力 $x \in \{0,1\}$ でラベル付けされている．出次数 0 のゲートは，出力ゲートと呼ばれる．本稿の論理回路は，出力ゲートが 1 つだけとする．

α_n を n 個の入力ゲートを持つ論理回路とする．列 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ を論理回路族と呼ぶ．各論理回路 α_n では，すべてのゲートは番号付けられており，番号 0 は出力ゲート，番号 $1, 2, \ldots, n$ は入力ゲートである．それぞれの自然数 n に対して，論理回路 α_n のゲート数が $z(n)$ で抑えられるとき，論理回路族 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ のゲート数（サイズ）は $z(n)$ であるという．論理回路 α_n において，出力ゲートから入力ゲート方向に遡る最長の経路上にあるゲートの数を，α_n の深さと呼ぶ．それぞれの自然数 n に対して，論理回路 α_n の深さが $t(n)$ で抑えられるとき，論理回路族 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ の深さは $t(n)$ であるという．

論理回路族 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ が，集合 $L \subseteq \{0,1\}^*$ を受理するというのは，それぞれの自然数 n に対して，論理回路 α_n が $L \cap \{0,1\}^n$ を受理するときをいう．つまり，入力 $x_1x_2\cdots x_n \in \{0,1\}^n$ 上の論理回路 α_n の出力値が 1 であるとき，かつそのときに限って，$x_1x_2\cdots x_n \in L$ となるときをいう．

ゲート g の左入力（右入力）の値を与えるゲートを g_L (g_R) と記す．論理回路 α_n の標準的符号化 $\overline{\alpha}_n$ とは，4 項組 $\langle g, t, g_L, g_R\rangle$ を連結した文字列である．ここで，g は，$\{0,1\}$ 上の文字列で表現されたゲート番号であり，$t \in \{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}, 0, 1\}$ はラベルである．4 項組 $\langle g, t, g_L, g_R\rangle$ は，「ゲート g は t でラベル付けられており，ゲート g の左右の入力は，それぞれ，g_L と g_R である」ことを表現している．$t \in \{0,1\}$ のときは（$t = \mathrm{NOT}$ のときは），4 項組の中の第 3・4 項目は（第 4 項目は）省略される．ゲート数 $z(n)$ の論理回路族 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ が，対数一様であるというのは，マッピング $n \to \overline{\alpha}_n$ が $O(\log z(n))$ 領域の決定性チューリング機械で計算できるときをいう．

NC^k は，深さが $O(\log^k n)$，ゲート数が n の多項式で抑えられる対数一様論理回路族で受理できる言語のクラスを表す．また，クラス NC は，すべての自然数 $k \geq 0$ に対して，NC^k の和集合をとったもの，つまり，$\mathrm{NC} = \cup_{k\geq 0}\mathrm{NC}^k$ である．以下では，ゲート数が n の多項式で抑えられ，深さが $O(\log^k n)$（深さが $\log n$ の多項式）で抑えられる対数一様論理回路族を，単に，NC^k 回路族（NC 回路族）と呼ぶ．

クラス NC に，非決定性の概念を導入したクラス Nondeterministic NC（略して，クラス NNC）を定義する．$\mathrm{NNC}(g(n))$ は，非決定性ゲートと呼ばれる特別なゲートを $g(n)$ 個持った NC 回路族で受理される言語のクラスである[9]．非決定性ゲートを $g(n)$ 個持った NC 回路を，$\mathrm{NNC}(g(n))$ 回路と呼ぶ．非決定性ゲートを持つ論理回路（略して，非決定性論理回路）α_n には，n 個の入力ゲートに対する入力列 $x = x_1x_2\cdots x_n \in \{0,1\}^n$，および，$g(n)$ 個の非決定性ゲートに対する非決定性入力列 $y = y_1y_2\cdots y_{g(n)} \in \{0,1\}^{g(n)}$ が回路に供給される．$\mathrm{NNC}(g(n))$ 回路が，入力列 x を受理するとは，その回路の出力を 1 にさせる非決定性ビット列 y が存在するときをいう．クラス NNC^k と $\mathrm{NNC}^k(g(n))$ 回路も，同様に定義される．

文献 9) では，$g(n) = O(\log n)$ のとき，$\mathrm{NNC}^k(g(n)) = \mathrm{NC}^k$ であること，$g(n)$ が多項式のときは，$\mathrm{NNC}(g(n)) = \mathrm{NP}$ であることが証明されている．これらの結果，および，非決定性ゲート数と深さの自明な性質から，**図 1** の包含関係が導かれる（図 1 では，$n^{O(1)}$，$(\log n)^{O(1)}$ を，それぞれ，poly，polylog と表現している．また，$k \geq 1$ は任意の自然数である）．

本稿では，非決定性論理回路族において，非決定性ゲートの多い回路族と，深さの大きい回路族の能力の関係を調べた．その結果，深さが $O(\log n)$ で非決定性ゲート数が n の多項式のクラス $\mathrm{NNC}^1(\mathrm{poly})$（図 1 の左上のクラス）は，深さが $\log n$ の多項式で，非決定性ゲート数が $O(\log n)$ のクラス $\mathrm{NNC}(O(\log n))$（図 1 の右下のクラス）を包含していることが分かった．つまり，$\mathrm{NNC}(O(\log n)) \subseteq \mathrm{NNC}^1(\mathrm{poly})$ という包含関係が成り立つ．この証明は，3 章で与える．この証明において，$\mathrm{NNC}(O(\log n)) \subseteq \mathrm{NNC}^1(\mathrm{poly})$ の左辺に対応する論理回路に，多項式個の非決定性ゲートを加えたとしても，右辺の論理回路に同数の非決定性ゲートを加えることで，同様の包含関係が得られる（3 章の最後の段落を参照）．したがって，次の定理が成り立つ．

定理 1　$\mathrm{NNC}(\mathrm{poly}) \subseteq \mathrm{NNC}^1(\mathrm{poly})$.

この定理から，ゲート数が n の多項式，深さが $\log n$ の多項式の任意の非決定性回路族は，非決定性ゲート数を多項式の範囲でいくらでも増やしてよいとすると，深さを $O(\log n)$ に圧縮できることが分かる．

次に，非決定性ゲート数を固定した場合について，非決定性回路族の能力を，非決定性チューリング機械で特徴づける．関数 $t(n) \geq n$, $s(n)$, $g(n)$ を，$O(\log t(n))$ 領域の決定性チューリング機械で計算可能な任意の関数とする．ただし，$s(n) \leq t(n)$ かつ $g(n) \leq t(n)$ である．このとき，次の 2 つの定理が成り立つ（これらの定理の証明は，それぞれ，4，5 章で与える）．

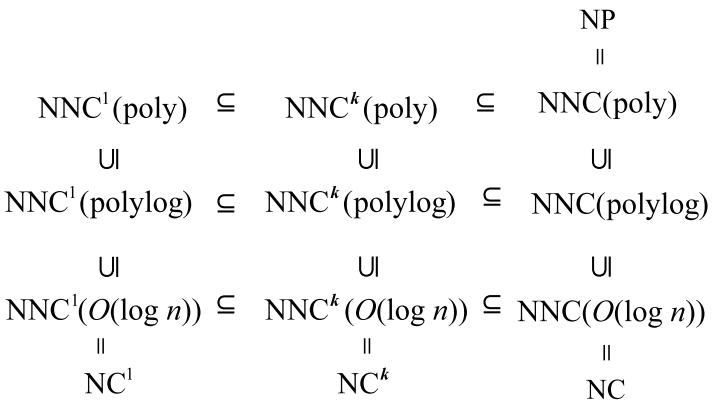

図 1　クラス $\mathrm{NNC}^k(g(n))$ の包含関係
Fig. 1　Inclusion relations among classes $\mathrm{NNC}^k(g(n))$.

定理 2　L を，$g(n)$ 回の非決定性動作を行う $t(n)$ 時間 $s(n)$ 領域の非決定性チューリング機械で受理できる任意の言語とする．このとき，L を受理する対数一様論理回路族で，非決定性ゲート数が $g(n)$，ゲート数が $O(t(n)s(n))$，深さが $O(t(n)\log s(n))$ のものが存在する．

定理 3　L を，$g(n)$ 回の非決定性動作を行う $t(n)$ 時間 $s(n)$ 領域，l テープの非決定性チューリング機械で受理できる任意の言語とする．このとき，L を受理する対数一様論理回路族で，非決定性ゲート数が $g(n)$，ゲート数が $O(t(n)(s(n))^{l+2})$，深さが $O(t(n))$ のものが存在する．

3. 定理 1 の証明

本章では，説明を簡単にするため，まず，$\mathrm{NC} \subseteq \mathrm{NNC}^1(\mathrm{poly})$ を証明する．次に，その証明において，左辺の論理回路に非決定性ゲートを多項式の範囲で追加することで，定理の包含関係 $\mathrm{NNC}(\mathrm{poly}) \subseteq \mathrm{NNC}^1(\mathrm{poly})$ を示す．

$C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ を，任意の対数一様論理回路族とし，そのゲート数を $z(n)$ とする．以下では，回路族 C を模倣する対数一様論理回路族 $C' = (\alpha'_1, \alpha'_2, \ldots, \alpha'_n, \ldots)$ を構成し，その深さが $O(\log z(n))$，ゲート数が $O(z(n))$，非決定性ゲート数が $z(n)$ で抑えられることを証明する．C のゲート数 $z(n)$ が，n の多項式で抑えられ，深さが $\log n$ の多項式で抑えられるとき，以下の記述は，$\mathrm{NC} \subseteq \mathrm{NNC}^1(\mathrm{poly})$ の証明となる．$\mathrm{NC} = \mathrm{NNC}(O(\log n))$ より[9]，$\mathrm{NNC}(O(\log n)) \subseteq \mathrm{NNC}^1(\mathrm{poly})$ が得られる．

回路 α_n のそれぞれのゲート g_i について（**図 2** (a) 参照），その出力線を，途中で分断する．分断した端点を，g_i に近いほうから o_i，p_i と呼ぶ（図 2 (b) 参照）．ゲート g_i の出力端点 o_i を，比較器 cmp_i の一方の入力とする（**図 3** 参照）．ここで，比較器 cmp_i は，2 つの入力値 x_1，x_2 から，出力値 $x'_1 := (x_1 \wedge x_2) \vee ((\neg x_1) \wedge (\neg x_2))$ を計算する回路であり，

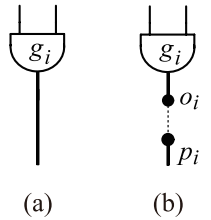

図 2　ゲート g_i の出力線を，端点 o_i，p_i に分ける
Fig. 2　The output line of gate g_i is cut into two ports o_i and p_i.

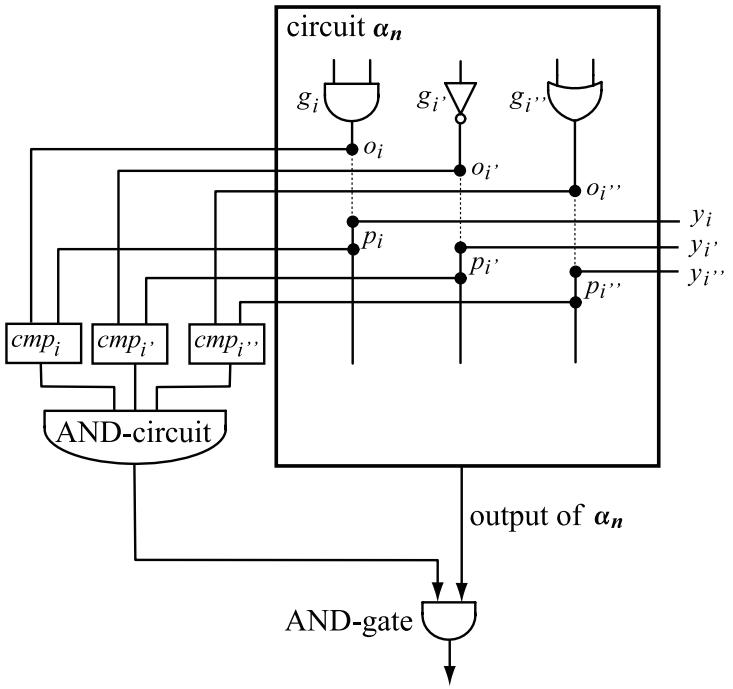

図 3　論理回路 α'_n．非決定性ゲート y_i から値を端点 p_i に与える．端点 o_i，p_i の値を回路 cmp_i で比較する
Fig. 3　Circuit α'_n. Nondeterministic gate y_i provides a value to p_i. Circuit cmp_i compares values of o_i and p_i.

ゲート数 5，深さ 3 で実現できる．容易に確かめられるように，$x_1 = x_2$ のとき，かつそのときに限って，$x'_1 = 1$ となる．

端点 p_i には，分断前は，ゲート g_i の出力値が供給されていた．分断後は，その代わりに，非決定性ゲート y_i を接続し，そこから g_i の値を推測して供給する．推測した値が，g_i の本来の出力値と一致しているかを確認するため，端点 p_i に与えられた値を，比較器 cmp_i のもう一方の入力とする．比較器 $cmp_1, cmp_2, \ldots, cmp_i, \ldots, cmp_{z(n)}$ の出力線は，$z(n)$ 本の入力線を持つ AND 回路に接続される．この AND 回路は，ゲート数 $z(n)-1$，深さ $\lceil \log z(n) \rceil$ の論理回路で実現できる．

回路 α_n に，以上の変更を施した後，回路 α_n の元々の出力線と，AND 回路の出力線を，新たな AND ゲートの入力とする．こうして構成された回路を，α'_n とする．α'_n の出力値は，最後に加えた AND ゲートの出力値である．

以上の構成から，すべての非決定性ゲート y_i が，それぞれ，ゲート g_i の出力値を正しく推測した場合に限り，AND 回路の出力値は 1 となり，α'_n の出力値が 1 となりうることが分かる．つまり，元々の回路 α_n が，入力列 $x_1x_2\cdots x_n \in \{0,1\}^n$ を受理するとき（つまり，α_n が 1 を出力するとき），かつそのときに限って，α'_n が入力列 $x_1x_2\cdots x_n$ を受理する（つまり，AND 回路の出力を 1 にする非決定性ビット列 $y \in \{0,1\}^{z(n)}$ が存在して，α'_n が 1 を出力する）．

α'_n において，出力ゲートから，入力ゲートまたは非決定性ゲートへの最長経路上には，最後に加えた AND ゲート，AND 回路を構成するゲート，比較器 cmp_i を構成するゲート，ゲート g_i，g_i の入力を与えるゲートの値を推測する非決定性ゲートがある．つまり，α'_n の深さは，たかだか $1 + \lceil \log z(n) \rceil + 3 + 1 + 1 = O(\log z(n))$ で抑えられる．

また，α'_n のゲート数は，$1 + (z(n) - 1) + 5z(n) + 2z(n) = O(z(n))$ である．ここで，$1 + (z(n) - 1)$ は，最後の AND ゲートと AND 回路を構成するゲート数である．$5z(n)$ は比較器 cmp_i のゲート数の総和，$2z(n)$ は，α_n の元々のゲート数と非決定性ゲート数の和である．したがって，$\mathrm{NC} \subseteq \mathrm{NNC}^1(\mathrm{poly})$ が成立する．

以上の記述において，元々の回路 α_n に多項式個の非決定性ゲートが含まれていたとしても，α'_n の対応する場所に同数の非決定性ゲートを加えるだけで，回路の模倣の関係は，そのまま成立する．したがって，$\mathrm{NNC}(\mathrm{poly}) \subseteq \mathrm{NNC}^1(\mathrm{poly})$ が得られる．

4. 定理 2 の証明

L を，$g(n)$ 回の非決定性動作を行う $t(n)$ 時間 $s(n)$ 領域の非決定性チューリング機械（Turing Machine，TM）で受理できる任意の言語とする．本章では，L を受理する対数一様論理回路族 $C = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots)$ で，非決定性ゲート数が $g(n)$，ゲート数が $O(t(n)s(n))$，深さが $O(t(n)\log s(n))$ のものを構成する．

一般には，$g(n)$ 回の非決定性動作を行う $t(n)$ 時間の非決定性 TM は，$g(n)$ 回の非決定性動作を，$t(n)$ 時間の動作中の任意の時刻に実行する．しかし，この $g(n)$ 回の非決定性動作を，最初の $g(n)$ ステップで連続して行い，残りの動作時間は，決定性 TM として動作すると仮定しても，L は $O(t(n))$ 時間 $s(n)$ 領域で受理できる（テープを新たに 1 本追加し，最初の $g(n)$ ステップで，$g(n)$ 回の非決定的選択を表す長さ $g(n)$ の $\{0,1\}$ 上の列を，追加したテープ上に非決定的に書き出しておく．残りの時刻は，この列を参考に動作させればよ

い）．以下の記述では，M を，そのように非決定性動作を行う TM とする．

以下では，M を模倣する非決定性論理回路 α_n を設計し，そのゲート数が $O(t(n)s(n))$，深さが $O(t(n)\log s(n))$，非決定性ゲート数が $g(n)$ であることを示す．

非決定性 TM の推移規則の定義：TM M の状態数を p とし，状態集合を $\{r_1, r_2, \ldots, r_p\}$ とする．ここで，r_1 を初期状態，r_2 を唯一の受理状態とする．M のテープ記号集合を $\{0,1\}$ とする（本稿では，記述を簡単にするために，M の記号数は 2 としている．しかし，以下の記述において，M のテープ記号数を 3 以上に拡張することは難しくない）．M は，右方向に無限の長さを持つテープを l 本持つとし，それぞれをテープ i と呼ぶ．ただし，$i \in \{1,2,\ldots,l\}$ である．

M の推移規則は，

$$\begin{aligned}&\delta(q;\, a_1, a_2, \ldots, a_l)\\ &= \{(q';\, a_1', a_2', \ldots, a_l';\, d_1', d_2', \ldots, d_l'), (q'';\, a_1'', a_2'', \ldots, a_l'';\, d_1'', d_2'', \ldots, d_l'')\}\end{aligned} \tag{1}$$

の形式の列で表現される．これは，M の状態が q で，l 本のテープのヘッドが，それぞれ，$a_1, a_2, \ldots, a_l$ を読んでいるとき，次の (a)，(b) の動作を非決定的に実行することを表している．(a) 状態を q' に変え，各ヘッドはそれぞれ $a_1', a_2', \ldots, a_l'$ を書き，それぞれ $d_1', d_2', \ldots, d_l'$ の方向へ動かす，または，(b) 状態を q'' に変え，各ヘッドはそれぞれ $a_1'', a_2'', \ldots, a_l''$ を書き，それぞれ $d_1'', d_2'', \ldots, d_l''$ の方向へ動かす．ここで，各 $i \in \{1,2,\ldots,l\}$ に対して，$d_i', d_i'' \in \{0,1\}$ は，値が 0 のときヘッドを左に，値が 1 のときヘッドを右に動かすことを意味する．推移規則が決定性であるとは，$q' = q''$，および，$a_i' = a_i''$，かつ，$d_i' = d_i''$ を，すべての $i \in \{1,2,\ldots,l\}$ に対して満足するときである．すべての推移規則が決定性である TM を，決定性 TM と呼ぶ．

TM の計算状況を表現するゲートの定義：M の状態 r_k は，自然数 $k \in \{1,2,\ldots,p\}$ を，長さ $\lfloor 1+\log p \rfloor$ の 2 進数で表現したビット列で書き表せる．時刻 $\tau \in \{0,1,2,\ldots,t(n)\}$ における M の状態を表すビット列を $state(\tau)$ で表す．これは，回路上では，$\lfloor 1+\log p \rfloor$ 個のゲートの値で表現できる．

$w_1(\tau,i,j)$ は，0 または 1 の値をとる 1 ビットの列で，$w_1(\tau,i,j)=1$ であるのは，時刻 $\tau \in \{0,1,2,\ldots,t(n)\}$ において，テープ $i \in \{1,2,\ldots,l\}$ のセル $j \in \{1,2,\ldots,s(n)\}$ の内容が 1 であるとき，かつそのときだけである．同様に，$w_0(\tau,i,j)=1$ であるのは，時刻 τ において，テープ i のセル j の内容が 0 であるとき，かつそのときだけである．

$h(\tau,i,j)$ は，0 または 1 の値をとる 1 ビットの列で，$h(\tau,i,j)=1$ であるのは，時刻 $\tau \in \{0,1,2,\ldots,t(n)\}$ において，テープ $i \in \{1,2,\ldots,l\}$ のセル $j \in \{1,2,\ldots,s(n)\}$ にヘッドが置かれているとき，かつそのときだけである．以上のように定義した $w_1(\tau,i,j)$，$w_0(\tau,i,j)$，$h(\tau,i,j)$ は，回路上では，それぞれ，1 個のゲートの値で表現できる．

y_τ を，各 $\tau \in \{0,1,2,\ldots,g(n)-1\}$ に対して，非決定性ゲートとする．$\tau \geq g(n)$ に対しては，y_τ は定数値 1 をとる定数ゲートとする．回路 α_n において，$y_\tau = 1$ のとき（$y_\tau = 0$ のとき），式 (1) の右辺の前者（後者）の推移を選択することを意味する．

式 (1) から，M の推移規則の数は，たかだか $p2^l$ であることが分かる．ただし，p は状態数であり，2 は記号数，l はテープ本数である．これらの推移規則を辞書式順序に並べたときに，第 m 番目に現れる推移規則を，推移規則 m と呼ぶ．ただし，$m \in \{1,2,\ldots,p2^l\}$ である．推移規則 m に現れる状態と記号，ヘッド移動方向を，それぞれ，$s(m)$，$s'(m)$，$s''(m)$；$a_i(m)$，$a_i'(m)$，$a_i''(m)$；$d_i'(m)$，$d_i''(m)$ で表現する．ただし，$m \in \{1,2,\ldots,p2^l\}$，$i \in \{1,2,\ldots,l\}$ である．また，それぞれは，推移規則 m の次の部分を表している．

$$\begin{aligned}&\delta\big(s(m);\, a_1(m), a_2(m), \ldots, a_l(m)\big)\\ &= \big\{\big(s'(m);\, a_1'(m), a_2'(m), \ldots, a_l'(m);\, d_1'(m), d_2'(m), \ldots, d_l'(m)\big),\\ &\quad \big(s''(m);\, a_1''(m), a_2''(m), \ldots, a_l''(m);\, d_1''(m), d_2''(m), \ldots, d_l''(m)\big)\big\}\end{aligned} \tag{2}$$

これらの値は，回路 α_n の中では，定数ゲートとして与えられている．

初期状況の構成：時刻 $\tau=0$ において，$state(\tau)$ の値は，長さ $\lfloor 1+\log p \rfloor$ の列 $00\cdots01$ である．これは，初期状態 r_1 を表している．列 $00\cdots01$ は，回路 α_n において，定数値 $0,0,\cdots,0,1$ をとる $\lfloor 1+\log p \rfloor$ 個の定数ゲートで表現される．M の入力列 $x = x_1x_2\cdots x_j\cdots x_n \in \{0,1\}^n$ は，$w_1(0,1,j)$，$w_0(0,1,j)$ で表現される．つまり，各 $j \in \{1,2,\ldots,n\}$ に対して，$w_1(0,1,j)$ は，回路 α_n の j 番目の入力ゲートで，その値は x_j である．また，$w_0(0,1,j) := \neg w_1(0,1,j)$ である．それ以外のすべての $w_1(0,i,j)$，$w_0(0,i,j)$ の値は 0 であり，これらのセルには，空白記号が書かれていることを意味する．

時刻 $\tau=0$ において，各テープ $i \in \{1,2,\ldots,l\}$ のヘッドは，セル 1 にある．したがって，$h(0,i,1)=1$ である．その他の $j \in \{2,3,\ldots,s(n)\}$ に対しては，$h(0,i,j)=0$ である．

時刻 τ から $\tau+1$ への動作：いま，M の時刻 τ の計算状況が，$state(\tau)$，$w_1(\tau,i,j)$，$w_0(\tau,i,j)$，$h(\tau,i,j)$ で表現できているとする．時刻 $\tau+1$ の計算状況を以下のように計算する．まず，時刻 τ の計算状況に対して，どの推移規則が適用可能かを調べる．

たとえば，式 (2) に示した推移規則 m を考える．この規則が，時刻 τ の計算状況に対して，適用可能か否かを確かめるためには，まず，状態 $state(\tau)$ と $s(m)$ が一致するかを確認する必要がある．これは，次の回路で実現できる．

$$match\text{-}state(\tau, m) = XNOR(state(t), s(m)) \tag{3}$$

ここで，$XNOR$ は，2 つのビット列 x，y が等しいかを確認する回路で，

$$XNOR(x, y) = \bigwedge_{u}(x_u y_u \vee (\neg x_u)(\neg y_u))$$

である．ただし，x_u，y_u は，それぞれ，x，y の第 u ビットを表している．

すべての $i \in \{1, 2, \ldots, l\}$ に対して，テープ i のヘッドが読んでいる記号が，$a_i(m)$ と一致してるかを確かめるため，式 (4)，(5) を計算し，それらから，式 (6) を計算する．式 (4) は，$w_1(\tau, i, j) \wedge h(\tau, i, j)$ によって，ヘッドが読んでいるセルの内容を取り出し，その値を $a_i(m)$ と比較している．式 (6) は，すべてのテープ $i \in \{1, 2, \ldots, l\}$ に対して，式 (4) または (5) の値が真になるかを計算している．

$$match\text{-}symbol_1(\tau, m, i) = \left(\bigvee_{1 \le j \le s(n)} \Big(w_1(\tau, i, j) \wedge h(\tau, i, j) \Big) \right) \wedge a_i(m) \tag{4}$$

$$match\text{-}symbol_0(\tau, m, i) = \left(\bigvee_{1 \le j \le s(n)} \Big(w_0(\tau, i, j) \wedge h(\tau, i, j) \Big) \right) \wedge \neg a_i(m) \tag{5}$$

$$match\text{-}symbol(\tau, m) = \bigwedge_{1 \le i \le l} \Big(match\text{-}symbol_1(\tau, m, i) \vee match\text{-}symbol_0(\tau, m, i) \Big) \tag{6}$$

式 (3)，(6) より，時刻 τ の計算状況に対して，推移規則 m が適用可能であるのは，次の式 (7) の値が 1 のときである．

$$match(\tau, m) = match\text{-}state(\tau, m) \wedge match\text{-}symbol(\tau, m) \tag{7}$$

時刻 $\tau + 1$ の状態は，式 (7) を利用して，次のように計算される．ただし，y_τ は，$0 \le \tau \le g(n) - 1$ のときは，1 または 0 の値をとる非決定性ゲート，$g(n) \le \tau$ のときは，値 1 の定数ゲートである．

$$state(\tau + 1) = \bigvee_{1 \le m \le p2^l} \Big(\Big((s'(m) \wedge y_\tau) \vee (s''(m) \wedge \neg y_\tau) \Big) \wedge match(\tau, m) \Big) \tag{8}$$

時刻 $\tau + 1$ のテープのセルの内容は，ヘッドがそのセルに置かれていたときは，推移規則に従って記号が書き込まれ，ヘッドが置かれていなかったときは，内容に変化はない．つまり，次の式 (9)，(10) のように計算される．

$$\begin{aligned} &w_1(\tau + 1, i, j) \\ &= \left(\bigvee_{1 \le m \le p2^l} \Big(\Big((a'_j(m) \wedge y_\tau) \vee (a''_j(m) \wedge \neg y_\tau) \Big) \wedge match(\tau, m) \Big) \right) \wedge h(\tau, i, j) \\ &\quad \vee \Big(w_1(\tau, i, j) \wedge \neg h(\tau, i, j) \Big) \end{aligned} \tag{9}$$

$$\begin{aligned} &w_0(\tau + 1, i, j) \\ &= \left(\bigvee_{1 \le m \le p2^l} \Big(\Big((\neg a'_j(m) \wedge y_\tau) \vee (\neg a''_j(m) \wedge \neg y_\tau) \Big) \wedge match(\tau, m) \Big) \right) \wedge h(\tau, i, j) \\ &\quad \vee \Big(w_0(\tau, i, j) \wedge \neg h(\tau, i, j) \Big) \end{aligned} \tag{10}$$

時刻 $\tau + 1$ のテープヘッドの位置は，時刻 τ の位置から，推移規則に従って，右または左に 1 セルだけ移動するので，次式のように計算される．

$$\begin{aligned} h(\tau + 1, i, j) = &\Big(h(\tau, i, j - 1) \wedge \Big((d'_i(m) \wedge y_\tau) \vee (d''_i(m) \wedge \neg y_\tau) \Big) \Big) \\ &\vee \Big(h(\tau, i, j + 1) \wedge \Big((\neg d'_i(m) \wedge y_\tau) \vee (\neg d''_i(m) \wedge \neg y_\tau) \Big) \Big) \end{aligned} \tag{11}$$

最終状態の出力：TM M の唯一の最終状態 r_2 を，長さ $\lfloor 1 + \log p \rfloor$ の記号列で表現した文字列 $0 \cdots 010$ と，時刻 $\tau = t(n)$ の状態 $state(t(n))$ が一致しているかをチェックする．式 (12) の結果が，回路 α_n の出力となる．

$$XNOR\Big(state(t(n)), \underbrace{0 \cdots 010}_{\lfloor 1 + \log p \rfloor} \Big) \tag{12}$$

α_n のゲート数と深さの解析：α_n のゲートのうち，$state(\tau)$，$w_1(\tau, i, j)$，$w_0(\tau, i, j)$，$h(\tau, i, j)$ の数は，$\tau \in \{0, 1, 2, \ldots, t(n)\}$，$i \in \{1, 2, \ldots, l\}$，$j \in \{1, 2, \ldots, s(n)\}$ なので，$O(t(n)s(n))$ で抑えられることが分かる．

それぞれの $\tau \in \{0, 1, 2, \ldots, t(n)\}$ に対して，時刻 τ の計算状況から $\tau + 1$ の計算状況を計算する回路のゲート数について考える．まず，式 (6) と式 (8)〜(10) に含まれる AND 回路 $\wedge_{1 \le i \le l}$ と OR 回路 $\vee_{1 \le m \le p2^l}$ の入力線数は定数であり，回路全体のゲート数のオーダ的評価に影響を与えない．

次に，式 (4)（または (5)）に着目する．ゲート $match\text{-}symbol_1(\tau, m, i)$ の数は，$\tau \in \{0, 1, 2, \ldots, t(n)\}$ だから，$O(t(n))$ で抑えられる．各 $match\text{-}symbol_1(\tau, m, i)$ の計算では，

$\vee_{1\le j\le s(n)}$ で表された OR 回路を使用しており，このゲート数は，$O(s(n))$ である．したがって，本段落の解析でも，ゲート数は $O(t(n)s(n))$ で抑えられる．ゆえに，回路 α_n 全体のゲート数は，$O(t(n)s(n))$ である．

時刻 τ の計算状況から $\tau+1$ の計算状況を計算する回路の深さは，式 (4)，(5) より，$O(\log s(n))$ で抑えられる．式 (3)，(6)〜(12) は，深さが定数の回路で構成できる．したがって，回路 α_n 全体の深さは，$O(t(n)\log s(n))$ である．

5. 定理 3 の証明

定理 3 は，定理 2 の深さ $O(t(n)\log s(n))$ を，$O(t(n))$ に改善するものである．

変数の拡張：4 章では，時刻 $\tau\in\{0,1,2,\ldots,t(n)\}$ における M の状態を表すビット列は，$state(\tau)$ であった．本章では，$state(\tau)$ の代わりに，$state(\tau;\, j_1,j_2,\ldots,j_l)$ を用いる（この手法は，文献 5) の技術を参考にしている）．ここで，各 $i\in\{1,2,\ldots,l\}$ に対して，$j_i\in\{1,2,\ldots,s(n)\}$ である．つまり，各 τ に対して，$state(\tau;\, j_1,j_2,\ldots,j_l)$ は，$(s(n))^l$ 個存在する．また，$j_1,j_2,\ldots,j_l$ は，$state(\tau;\, j_1,j_2,\ldots,j_l)$ を構成するゲートのゲート番号の一部になっている．

これら $(s(n))^l$ 個の $state(\tau;\, j_1,j_2,\ldots,j_l)$ のうち，$j_1,j_2,\ldots,j_l$ が，現在のヘッド位置を正しく表しているものだけが，時刻 τ における M の状態を保持するようにする．それ以外の $state(\tau;\, j_1,j_2,\ldots,j_l)$ には，値 0 を保持させる．

$state(\tau;\, j_1,j_2,\ldots,j_l)$ は，$state(\tau)$ と同様に，回路上では，$\lfloor 1+\log p\rfloor$ 個のゲートの値で表現する．直感的には，l 本のヘッド位置 $j_1,j_2,\ldots,j_l$ の移動に合わせて，状態の値を保持する $state(\tau;\, j_1,j_2,\ldots,j_l)$ を変えるようにする．

本章では，このような拡張を，$w_1(\tau,i,j)$，$w_0(\tau,i,j)$，$h(\tau,i,j)$ にも行う．4 章では，$match\text{-}symbol_1(\tau,m,i)$ の計算において（式 (4) 参照），$h(\tau,i,j)=1$ となる j の値を探索するといった作業が必要であった．そのために，入力線数 $s(n)$ の OR 回路 $\vee_{1\le j\le s(n)}$ を用いていた．本章では，ヘッドの動きに合わせて，情報を保持するゲートを変えているので，$h(\tau,i,j)=1$ となる j の値を探索する必要がなくなる（詳しくは，式 (15) で説明する）．

初期状況の構成：時刻 $\tau=0$ においては，すべてのテープのヘッドは，先頭のセル 1 を読んでいる．したがって，初期状態 r_1 の情報は，$state(0;\, 1,1,\ldots,1)$ に保持されており，それ以外の $state(0;\, j_1,j_2,\ldots,j_l)$ には，値 0 が保持されている．

テープ記号についても，$w_1(\tau,i,j)$，$w_0(\tau,i,j)$ は，それぞれ，$w_1(\tau,i,j;\, j_1,j_2,\ldots,j_l)$，$w_0(\tau,i,j;\, j_1,j_2,\ldots,j_l)$ に置き換える．時刻 $\tau=0$ においては，各 $j\in\{1,2,\ldots,n\}$ に対して，$w_1(0,1,j;\, 1,1,\ldots,1)$ に，入力列 $x_1x_2\cdots x_j\cdots x_n\in\{0,1\}^n$ が与えられ，$w_0(0,1,j;\, 1,1,\ldots,1):=\neg w_1(0,1,j;\, 1,1,\ldots,1)$ である．その他の $w_1(0,i,j;\, j_1,j_2,\ldots,j_l)$ と $w_0(0,i,j;\, j_1,j_2,\ldots,j_l)$ の値は，すべて 0 である．

ヘッド位置 $h(\tau,i,j)$ は，$h(\tau,i,j_i;\, j_1,j_2,\ldots,j_i\ldots,j_l)$ に置き換える．時刻 $\tau=0$ では，すべてのテープ $i\in\{1,2,\ldots,l\}$ のヘッドは，セル 1 を読んでいるので，$h(0,i,1;\, 1,1,\ldots,1)=1$ である．その他の $h(0,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)$ の値は 0 とする．

$state(\tau;\, j_1,j_2,\ldots,j_l)$ において，τ と $j_1,j_2,\ldots,j_l$ は，ゲート番号の一部である．$j_1,j_2,\ldots,j_l$ が，時刻 τ の実際のヘッド位置に一致しているか否か，つまり，時刻 τ の状態を保持すべきゲートが，$state(\tau;\, j_1,j_2,\ldots,j_l)$ でよいか否かは，次式の値が 1 かどうかで判断できる（$h\text{-}pos$ は，head positions の略である）．

$$h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)=\bigwedge_{1\le i\le l}h(\tau,i,j_i\,;\, j_1,j_2,\ldots,j_i\ldots,j_l)\tag{13}$$

時刻 $\tau=0$ では，$h\text{-}pos(0;\, 1,1,\ldots,1)=1$ であり，その他の $h\text{-}pos(0;\, j_1,j_2,\ldots,j_l)$ の値は 0 である．

時刻 τ から $\tau+1$ への動作：現在の TM の状態と，推移規則 m の状態 $s(m)$ が一致しているかを調べるには，式 (3) を拡張した式 (14) を用いる．これは，$(s(n))^l$ 個の $state(t;\, j_1,j_2,\ldots,j_l)$ の中から，ヘッド位置を正しく表すものを，$h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)$ で取り出して，その値を $s(m)$ と比較している．

$$\begin{aligned}&match\text{-}state(\tau,m;\, j_1,j_2,\ldots,j_l)\\&=XNOR\Big(\Big(state(\tau;\, j_1,j_2,\ldots,j_l)\wedge h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)\Big),s(m)\Big)\end{aligned}\tag{14}$$

テープ i のヘッドが読んでいる記号が，$a_i(m)$ と一致してるかを確かめるには，式 (4)，(5) の代わりに，式 (15)，(16) を用いる．式 (15)，(16) において右辺の w_1，w_0 の第 3 番目の変数は，j ではなく，j_i になっている．これにより，ヘッドが読んでいる記号のみを取り出せる．右辺の $h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)$ で，$j_1,j_2,\ldots,j_l$ がヘッド位置を正しく表している $w_1(\tau,i,j_i;\, j_1,j_2,\ldots,j_l)$ を取り出して，$a_i(m)$ と比較している．この仕組みにより，式 (4)，(5) で使っていた OR 回路 $\vee_{1\le j\le s(n)}$ が，式 (15)，(16) では不要になっている．

$$\begin{aligned}&match\text{-}symbol_1(\tau,m,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\\&=w_1(\tau,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\wedge h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)\wedge a_i(m)\end{aligned}\tag{15}$$

$$\begin{aligned}&match\text{-}symbol_0(\tau,m,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\\&=w_0(\tau,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\wedge h\text{-}pos(\tau;\, j_1,j_2,\ldots,j_l)\wedge\neg a_i(m)\end{aligned}\tag{16}$$

したがって，すべてのテープ $i \in \{1,2,\ldots,l\}$ において，ヘッドが読んでいる記号が，それぞれ，$a_i(m)$ と一致してるかを確かめるには，次の式 (17) を用いればよい（式 (17) は，式 (6) を拡張したものである）．

$$\begin{aligned}&\mathit{match\text{-}symbol}(\tau,m;\, j_1,j_2,\ldots,j_l)\\&=\bigwedge_{1\le i\le l}\Big(\mathit{match\text{-}symbol}_1(\tau,m,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\\&\qquad\vee\mathit{match\text{-}symbol}_0(\tau,m,i,j_i;\, j_1,j_2,\ldots,j_i,\ldots,j_l)\Big)\end{aligned}\tag{17}$$

式 (14)，(17) より，時刻 τ の計算状況に対して，推移規則 m が適用可能であるか否かは，時刻 τ のヘッド位置 $j_1,j_2,\ldots,j_l$ の値を，ゲート番号として正しく含んでいる次式 $\mathit{match}(\tau,m;\, j_1,j_2,\ldots,j_l)$ が，値 1 をとるときである．

$$\begin{aligned}&\mathit{match}(\tau,m;\, j_1,j_2,\ldots,j_l)\\&=\mathit{match\text{-}state}(\tau,m;\, j_1,j_2,\ldots,j_l)\wedge\mathit{match\text{-}symbol}(\tau,m;\, j_1,j_2,\ldots,j_l)\end{aligned}$$

次にすることは，時刻 $\tau+1$ の計算状況の構成である．まず，時刻 τ において，推移規則 m が使われた場合に，各テープ i で，ヘッドがどちらの方向に動くのかを表す関数を作る．式 (18) のゲート $\mathit{direct}_1(\tau,m;\, d_1,d_2,\ldots,d_l)$ は，ゲート番号として $d_1,d_2,\ldots,d_l\in\{0,1\}$ を含んでいる．$\mathit{direct}_1(\tau,m;\, d_1,d_2,\ldots,d_l)=1$ となるのは，推移規則 m の右辺の第 1 項（式 (2) 参照）が使われた場合に，各テープ $i\in\{1,2,\ldots,l\}$ のヘッドが d_i 方向へ動くとき，かつそのときに限られる．$d_1,d_2,\ldots,d_l\in\{0,1\}$ だから，各 (τ,m) に対して，$\mathit{direct}_1(\tau,m;\, d_1,d_2,\ldots,d_l)$ は 2^l 個存在する．2^l 個の中のちょうど 1 つだけが，値 1 をとる．また，式 (19) の $\mathit{direct}_2(\tau,m;\, d_1,d_2,\ldots,d_l)$ は，推移規則 m の右辺の第 2 項について同様である．

$$\mathit{direct}_1(\tau,m;\, d_1,d_2,\ldots,d_l)=\bigwedge_{1\le i\le l}\mathit{XNOR}\Big(d_i,d_i'(m)\Big)\tag{18}$$

$$\mathit{direct}_2(\tau,m;\, d_1,d_2,\ldots,d_l)=\bigwedge_{1\le i\le l}\mathit{XNOR}\Big(d_i,d_i''(m)\Big)\tag{19}$$

それぞれの $i\in\{1,2,\ldots,l\}$ に対して，$d_i=0$ のとき $\hat{d}_i=-1$，$d_i=1$ のとき $\hat{d}_i=+1$ と定義する．時刻 τ において，ヘッド位置が $j_1,j_2,\ldots,j_l$ だったとする．時刻 $\tau+1$ への推移において，各テープ i のヘッドが，d_i 方向へ動いたと仮定すると，時刻 $\tau+1$ のヘッド位置は，$j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l$ と表現できる．

時刻 τ の状態は，ヘッド位置 $j_1,j_2,\ldots,j_l$ を，ゲート番号として含む $\mathit{state}(\tau;\, j_1,j_2,\ldots,j_l)$ に保持されている．時刻 $\tau+1$ の状態を引き継ぐゲートは，各テープ i のヘッドが d_i 方向へ動くとすると，$\mathit{state}(\tau;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l)$ である．以下では，これを実現する回路を作る．

まず，すべての推移規則 $m\in\{1,2,\ldots,p2^l\}$，すべてのヘッド移動方向の組合せ $d_1,d_2,\ldots,d_l\in\{0,1\}$，1 ステップ後のすべてのベッド位置の組合せ $j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l\in\{1,2,\ldots,s(n)\}$ に対して，次式を計算しておく．

$$\begin{aligned}&\mathit{state}(\tau+1,m;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l;\, d_1,d_2,\ldots,d_l)\\&=\Big(s'(m)\wedge y_\tau\wedge\mathit{match}(\tau,m;\, j_1,j_2,\ldots,j_l)\wedge\mathit{h\text{-}pos}(\tau;\, j_1,j_2,\ldots,j_l)\\&\qquad\wedge\mathit{direct}_1(\tau,m;\, d_1,d_2,\ldots,d_l)\Big)\\&\vee\Big(s''(m)\wedge\neg y_\tau\wedge\mathit{match}(\tau,m;\, j_1,j_2,\ldots,j_l)\wedge\mathit{h\text{-}pos}(\tau;\, j_1,j_2,\ldots,j_l)\\&\qquad\wedge\mathit{direct}_2(\tau,m;\, d_1,d_2,\ldots,d_l)\Big)\end{aligned}\tag{20}$$

次に，式 (20) から，時刻 τ において適用可能な推移規則 $m\in\{1,2,\ldots,p2^l\}$ を取り出す（式 (21) 参照）．つまり，$\mathit{match}(\tau,m;\, j_1,j_2,\ldots,j_l)=1$ となる m を探索する．さらに，式 (21) から，各値 $d_1,d_2,\ldots,d_l\in\{0,1\}$ が，実際のヘッド移動方向を表しているものだけを取り出す（式 (22) 参照）．

$$\begin{aligned}&\mathit{state}(\tau+1;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l;\, d_1,d_2,\ldots,d_l)\\&=\bigvee_{1\le m\le p2^l}\mathit{state}(\tau+1,m;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l;\, d_1,d_2,\ldots,d_l)\end{aligned}\tag{21}$$

$$\begin{aligned}&\mathit{state}(\tau+1;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l)\\&=\bigvee_{d_1,d_2,\ldots,d_l\in\{0,1\}}\mathit{state}(\tau+1;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l;\, d_1,d_2,\ldots,d_l)\end{aligned}\tag{22}$$

同様に，時刻 $\tau+1$ のテープの内容も，各テープ i のヘッドが d_i 方向へ動いた後のヘッド位置 $j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l$ を，ゲート番号として持つゲートに，値を引き継ぐ必要がある．まず，式 (20) と同様に，すべての $m\in\{1,2,\ldots,p2^l\}$，すべての $d_1,d_2,\ldots,d_l\in\{0,1\}$，すべての $j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_l+\hat{d}_l\in\{1,2,\ldots,s(n)\}$ に対して，次の関数を計算しておく．

$$w_1(\tau+1,m,i,j;\, j_1+\hat{d}_1,j_2+\hat{d}_2,\ldots,j_i+\hat{d}_i,\ldots,j_l+\hat{d}_l;\, d_1,d_2,\ldots,d_i,\ldots,d_l)\tag{23}$$

式 (23) の w_1 は，値 j をゲート番号の一部として含んでいる（式 (23) の第 4 番目の変数

j を見よ)．この値が $j = j_i$ を満たすゲート w_1 は，式 (24) で定義され，$j \neq j_i$ のゲート w_1 は，式 (25) で定義される．つまり，テープ i のヘッドが読んでいるセル j_i については，式 (24) のように，推移規則に従って，内容が書き換えらる．ヘッドが読んでいないセルは，式 (25) のように，1 ステップ前のセルの内容を引き継ぐ．したがって，式 (23) の w_1 の値を計算する回路は，ゲート番号の一部である j の値によって，結線構造が異なることになる．

$$
\begin{aligned}
&w_1(\tau+1, m, i, j_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_i, \ldots, d_l)\\
&= \Big(a'_i(m) \wedge y_\tau \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_i, \ldots, j_l)\\
&\qquad \wedge \mathit{direct}_1(\tau, m;\, d_1, d_2, \ldots, d_i, \ldots, d_l)\Big)\\
&\vee \Big(a''_i(m) \wedge \neg y_\tau \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_i, \ldots, j_l)\\
&\qquad \wedge \mathit{direct}_2(\tau, m;\, d_1, d_2, \ldots, d_i, \ldots, d_l)\Big) \qquad (24)\\
&w_1(\tau+1, m, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l)\\
&= \Big(w_1(\tau, i, j;\, j_1, j_2, \ldots, j_l) \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_l) \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_l)\\
&\qquad \wedge \mathit{direct}_1(\tau, m;\, d_1, d_2, \ldots, d_l) \wedge y_\tau\Big)\\
&\vee \Big(w_1(\tau, i, j;\, j_1, j_2, \ldots, j_l) \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_l) \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_l)\\
&\qquad \wedge \mathit{direct}_2(\tau, m;\, d_1, d_2, \ldots, d_l) \wedge \neg y_\tau\Big) \qquad (25)
\end{aligned}
$$

式 (21)，(22) と同様に，式 (23) から，時刻 τ において適用可能な推移規則 m を取り出し，次いで，ヘッドの移動方向 $d_1, d_2, \ldots, d_l \in \{0,1\}$ が，実際の移動方向を表しているものだけを取り出す．すると，式 (26)，(27) を順に得る．

$$
\begin{aligned}
&w_1(\tau+1, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l)\\
&= \bigvee_{1 \le m \le p2^l} w_1(\tau+1, m, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l) \qquad (26)\\
&w_1(\tau+1, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l)\\
&= \bigvee_{d_1, d_2, \ldots, d_l \in \{0,1\}} w_1(\tau+1, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l) \qquad (27)
\end{aligned}
$$

$w_0(\tau+1, i, j;\, j_1, j_2, \ldots, j_l)$ も同様である．

時刻 $\tau+1$ のヘッド位置でも，まず，式 (28) を計算しておく．

$$
\begin{aligned}
&h(\tau+1, m, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_i, \ldots, d_l)\\
&= \Big(h(\tau, i, j_i;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_i, \ldots, j_l)\\
&\qquad \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{direct}_1(\tau, m;\, d_1, d_2, \ldots, d_i, \ldots, d_l) \wedge y_\tau\Big)\\
&\vee \Big(h(\tau, i, j_i;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{match}(\tau, m;\, j_1, j_2, \ldots, j_i, \ldots, j_l)\\
&\qquad \wedge \mathit{h\text{-}pos}(\tau;\, j_1, j_2, \ldots, j_i, \ldots, j_l) \wedge \mathit{direct}_2(\tau, m;\, d_1, d_2, \ldots, d_i, \ldots, d_l) \wedge \neg y_\tau\Big) \qquad (28)
\end{aligned}
$$

式 (26)，(27) と同様に，時刻 τ において適用可能な推移規則 m と，ヘッドの移動方向 $d_1, d_2, \ldots, d_l \in \{0,1\}$ を取り出して，式 (29)，(30) を得る．

$$
\begin{aligned}
&h(\tau+1, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_i, \ldots, d_l)\\
&= \bigvee_{1 \le m \le p2^l} h(\tau+1, m, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_i, \ldots, d_l) \qquad (29)\\
&h(\tau+1, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l)\\
&= \bigvee_{d_1, d_2, \ldots, d_l \in \{0,1\}} h(\tau+1, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_i+\hat{d}_i, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_i, \ldots, d_l) \qquad (30)
\end{aligned}
$$

最終状態の出力：TM M が唯一の受理状態 r_2 に入るのは，すべてのテープのヘッドをセル 1 に戻した状況のときのみと仮定しても一般性を失わない．この仮定の下で，式 (12) は，次の式 (31) に置き換える．

$$
\mathit{XNOR}\Big(\mathit{state}\Big(t(n);\, 1, 1, \ldots, 1\Big),\, \underbrace{0 \cdots 010}_{\lfloor 1+\log p \rfloor}\Big) \qquad (31)
$$

α_n のゲート数と深さの解析：式 (20)，(23)，(28) の左辺に現れるゲートは，

$$
\begin{aligned}
&\mathit{state}(\tau+1, m;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l),\\
&w_1(\tau+1, m, i, j;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l),\\
&h(\tau+1, m, i, j_i+\hat{d}_i;\, j_1+\hat{d}_1, j_2+\hat{d}_2, \ldots, j_l+\hat{d}_l;\, d_1, d_2, \ldots, d_l)
\end{aligned}
$$

である．テープ i と推移規則 m は，$1 \le i \le l$，$1 \le m \le p2^l$ を満たす．また，$d_1, d_2, \ldots, d_l \in \{0,1\}$ である．ここで，テープ本数 l と状態数 p は，ともに定数である．一方，

$0 \le \tau \le t(n)-1$，$1 \le j \le s(n)$，かつ，各 $1 \le i \le l$ に対して，$1 \le j_i + \hat{d}_i \le s(n)$ である．したがって，回路 α_n 全体のゲート数は，ゲート w_1 の数に依存し，$O(t(n)(s(n))^{l+1})$ で抑えられる．ここで，4 章の第 2 段落でおいた仮定を思い出されたい．その段落では，新たなテープを 1 本追加しており，追加後のテープ数を 4 章では l としていた．テープ追加前のテープ数を，改めて l とおくと，回路 α_n 全体のゲート数は，$O(t(n)(s(n))^{l+2})$ となることが分かる．

式 (13)〜(30) に現れる OR 回路 $\vee_{1\le i\le l}$，$\vee_{1\le m\le p2^l}$，$\vee_{d_1,d_2,\ldots,d_l\in\{0,1\}}$ の入力線数は，すべて定数であり，それらの深さも定数となる．式 (13)〜(30) は，各 $\tau \in \{0,1,\ldots,t(n)-1\}$ に対して，時刻 τ から $\tau+1$ への回路を表したものである．したがって，回路 α_n 全体の深さは，$O(t(n))$ で抑えられる．

6. ま と め

本稿では，非決定性回路族の深さと非決定性ゲート数の関係について調べた．その結果，NNC(poly) $\subseteq$ NNC^1(poly) なる包含関係を得た．また，非決定性 TM を，効率良く模倣する非決定性回路族を設計した．その結果，$g(n)$ 回の非決定性動作を行う $t(n)$ 時間 $s(n)$ 領域の l テープ非決定性 TM は，非決定性ゲート数が $g(n)$，ゲート数が $O(t(n)s(n))$，深さが $O(t(n)\log s(n))$ の非決定性論理回路で模倣できること，および，ゲート数を $O(t(n)(s(n))^{l+2})$ に増やせば，深さ $O(t(n))$ で模倣できることが分かった．

参 考 文 献

1) http://www.claymath.org/millennium/
2) Hopcroft, J.E., Motwani, R. and Ullman, J.D.: *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (2006).
3) Iwamoto, C., Hatayama, N., Morita, K., Imai, K. and Wakamatsu, D.: Hierarchies of DLOGTIME-Uniform Circuits, Machines, *Computations and Universality* (*Proc. MCU 2004*), Margenstern, M. (Ed.), Saint-Petersburg, Sep. 21–26, 2004, LNCS, Vol.3354, Springer, pp.211–222 (2005).
4) Iwamoto, C., Hatayama, N., Nakashiba, Y., Morita, K. and Imai, K.: Translational Lemmas for DLOGTIME-Uniform Circuits, Alternating TMs and PRAMs, *Acta Informatica*, Vol.44, No.5, pp.345–359 (2007).
5) Iwama, K. and Iwamoto, C.: Parallel Complexity Hierarchies Based on PRAMs and DLOGTIME-Uniform Circuits, *Proc. 11th Annual IEEE Conf. on Computational Complexity*, Philadelphia, pp.24–32 (1996).
6) 岩本宙造：階層定理，電子情報通信学会，知識ベース 6 群 2 編「計算論とオートマトン」5 章「決定性，非決定性計算の複雑さ」，第 2 節．http://www.ieice-hbkb.org/portal/
7) Johnson, D.S.: A Catalog of Complexity Classes, *Handbook of Theoretical Computer Science*, van Leeuwen, J. (Ed.), Vol.A, pp.69–161, MIT Press, Amsterdam (1990).
8) Ruzzo, W.L.: On Uniform Circuit Complexity, *Journal of Computer and System Sciences*, Vol.22, pp.365–383 (1981).
9) Wolf, M.J.: Nondeterministic Circuits, Space Complexity and Quasigroups, *Theoretical Computer Science*, Vol.125, pp.295–313 (1994).

(平成 22 年 9 月 17 日受付)
(平成 23 年 1 月 14 日採録)

推　薦　文

計算の複雑さをクラスによって表現する計算複雑性理論において，クラス P が内包するクラス NC を段階的に回路（計算機）の深さおよび非決定性ゲート数を変化させ大きなクラスを定義する研究が行われてきた．推薦論文では，従来知られていなかった回路の深さの増大と非決定性ゲート数の増加により拡張した受理言語クラス間の包含関係を明らかにした画期的な成果である．（情報処理学会中国支部長　北村俊明）

岩本　宙造（正会員）

昭和 42 年生．平成 7 年九州大学大学院工学研究科情報工学専攻博士後期課程修了．九州大学工学部情報工学科助手を経て，平成 7 年九州芸術工科大学講師．平成 9 年広島大学工学部第二類助教授．現在，広島大学大学院工学研究科情報工学専攻准教授．計算の複雑さの理論，オートマトン理論等の研究に従事．電子情報通信学会会員．

小野　優介

平成 20 年広島大学大学院工学研究科情報工学専攻博士前期課程修了．同年シャープ株式会社入社．

森田　憲一（正会員）

昭和 24 年生．昭和 48 年大阪大学大学院基礎工学研究科修士課程修了．工学博士．昭和 49 年大阪大学基礎工学部助手．昭和 62 年山形大学工学部助教授．平成 2 年山形大学工学部教授．平成 5 年広島大学工学部第二類（電気）教授．平成 13 年広島大学大学院工学研究科情報工学専攻教授．この間，パリ第 13 大学客員教授（平成 5 年），Metz 大学（フランス）客員教授（平成 9 年），Rovira i Virgili 大学（スペイン）国際 Ph.D. コース客員教授（平成 14 年〜現在）．オートマトン理論（特にセルオートマトン），計算の基礎理論，可逆コンピューティング，形式言語理論等に関する研究に従事．

今井　克暢

昭和 40 年生．平成 4 年大阪大学大学院基礎工学研究科物理系専攻博士前期課程修了．現在，広島大学大学院工学研究科情報工学専攻助教．セルオートマトンの研究に従事．電子情報通信学会会員．