$\log^{()} n$ の領域強構成可能性について

上原隆平
東京女子大学 情報処理センター

1993 年 夏の LA

Abstract

本稿では確率 Turing Machine(PTM) における、小さな領域量に関する領域構成可能性をあつかう。まず $\log\log$ の Monte Carlo(MC-) 領域強構成可能性に関する新しい証明を示す。これは従来の証明における不十分な部分を修正し、より厳密に MC-領域強構成可能性を示すものである。次に PTM において、正解を出力する確率についての制限を緩和した計算を考える。こうした計算において、関数 $\log\log$ がほとんどの入力に対して領域強構成可能であり、さらに一般の関数 $\log\log\log$, $\log\log\log\log$, が平均的に領域強構成可能であることを示す。

1 はじめに

入力の長さ n に対して、ごく小さな領域量に関する研究がこれまでいくつか行われてきた。特に関数 $\log\log n$ に関しては、決定性 Turing Machine(TM) では領域強構成可能 (fully space constructible) ではないことが知られており [1, 6]、また非決定性 TM においては、領域強構成可能ではないであろうと予想されている [7]。

それに対し、確率 Turing Machine(PTM) の一種である Monte Carlo Turing Machine (MTM) では、関数 $\log\log n$ が Monte Carlo(MC-) 領域強構成可能であることが証明されている [5]。しかしながら、実はこの証明は不十分なものであり、厳密に言えば、関数 $\log\log n$ の領域強構成可能性を示していない。

本稿では、まずこの点を改善する新しい証明を示す。本稿で示すアルゴリズムでは、出力される値の確率分布や平均、分散を実際に計算することができるので、正しい答を出力する確率を具体的に評価することができ、その結果 $\log\log n$ の MC-領域強構成可能性を、より厳密に示すことができる。

次に PTM における、より小さな領域に関する領域構成可能性について検討する。(こうした問題に関しては、上記の文献 [5] で $\log\log\log n$ が MC- 領域強構成可能であるかどうかが Open Problem として示されている。) 本稿ではこうした問題へのアプローチとして、PTM における、正解を出力する確率についての制限を緩和した計算を導入する。従来の確率アルゴリズムでは、すべての入力に対して、正解を出力する確率が一定の値以上である場合を考えていた。それに対してここでは、入力は長さに関して無作為に与えられると仮定し、正解を出力する確率は、入力の長さに対しての平均的な確率で定義する。つまりこの場合、特定の長さの入力に対しては、正解を出力する確率が必ずしも高くなくてもかまわない。まずこうした計算において、関数 $\log\log n$ がほとんどの入力に対して領域強構成可能であることを示す。さらにその証明で用いるアルゴリズムを再帰的に組み合わせることにより、一般の関数 $\log\log\log n, \log\log\log\log n, \cdots$ が平均的に領域強構成可能であることを示す。ここで平均的に領域強構成可能であるとは、PTM の使用する領域の期待値がちょうど $\log\log\log n, \log\log\log\log n, \cdots$となる、という意味である。

本稿で定義した領域強構成可能性について直感的に比較すれば、ほとんどの入力に対して領域強構成可能であるということは、従来の意味で領域強構成可能であることと比べても、それほど能力的な差はないと考えられる。それに対して、平均的に領域強構成可能であるということは、ごく弱い意味での構成可能性であると言える。これらの能力の違いや、例えば $\log\log\log n$ がより強い意味で領域強構成可能 (例えば MC-領域強構成可能性) であるかどうかといった問題はまだ未解決である。

2 定義

PTM の定義の詳細は [3] にゆずる。直感的には PTM とは片寄りのないコインを投げ、その結果によって遷移

先を決定するような状態をもつ TM である。ここで M が PTM であったとき、Φ_{M} を M によって計算される関数とする。f が **probabilistic** SPACE$(s(n))$(PrSPACE$(s(n))$) に含まれるとは、$\Phi_{\mathsf{M}} = f$ であるような PTM M が存在し、すべての入力 x に対して

$$\Pr\{\mathsf{M} \text{ が入力 } x \text{ に対して高々} s(|x|) \text{ 領域しか使用せずに} \Phi_{\mathsf{M}}(x) \text{ を出力する}\} > \frac{1}{2}$$

を満たすときを言う。ここで領域量が正しい答を出力するときにだけ問題とされていることに注意する。つまり一般に、上記の条件をみたす M において、間違った答を出力する場合には $s(n)$ 領域以上の領域を使用してもかまわない。

また上記の $\frac{1}{2}$ を $\frac{3}{4}$ で置き換えたものを **Monte Carlo** Turing Machine(MTM) と呼び、MTM における領域量のクラスを MSPACE$(s(n))$ と記述する [5]。

関数 $f : \mathsf{N} \to \mathsf{N}$ が MC-領域強構成可能 (fully Monte Carlo constructible) であるとは、長さ n のすべての入力に対してちょうど $f(n)$ 領域だけ使うような MTM M が存在するときを言う。

関数 $f : \mathsf{N} \to \mathsf{N}$ がほとんどの入力に対して (AI-) 領域強構成可能 (fully space constructible for almost inputs) であるとは、任意の $\epsilon(0 < \epsilon < 1)$ が与えられたとき、無作為な長さ n のすべての入力に対して、$1-\epsilon$ 以上の確率でちょうど $f(n)$ 領域だけ使うような PTM M が存在するときを言う。

すなわち関数 f が AI-領域強構成可能であるとは、入力の長さに対して平均をとったときに $f(n)$ 領域を使用する確率が $1-\epsilon$ よりも大きくなる場合をいい、任意に与えられた定数 $\epsilon(0 < \epsilon < 1)$ に対して、以下を満たす PTM M が存在するときを言う:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} \Pr\{\mathsf{M} \text{ は長さ } i \text{ のすべての入力に対して } f(i) \text{ 領域だけ使用する}\} > 1-\epsilon$$

ここで関数 f が AI-領域強構成可能である場合は、入力の長さに対して平均をとっていることに注意する。つまり一般に、特定の長さの入力に対しては、正解を出す確率が低くなってもかまわない。

関数 $f : \mathsf{N} \to \mathsf{N}$ が平均的に領域強構成可能 (fully space constructible on the average) であるとは、長さ n のすべての入力に対して、使用する領域の期待値が $f(n)$ 領域であるような PTM M が存在するときを言う。

本稿では底が 2 である対数を log で表記し、自然対数は ln で表現する。また対数を繰り返し適用する演算について、$\log^{(1)} n = \log n$, $\log^{(k+1)} n = \log \log^{(k)} n$ で定義される記述を必要に応じて使用する。

また自然対数の底を $e(\approx 2.718)$ と表記し、Eular の定数を $\gamma(\approx 0.577)$ と表記する。コイントスの結果は {'1','0'} のどちらかとし、連とは、複数回のコイントスによる列のうち、同じ結果が連続している部分列をいう。

3 $\log\log n$ の MC-領域強構成可能性の証明

$\log\log n$ が MC-領域強構成可能であることを示すには、使用する領域量が $\log\log n$ になる MTM が存在することを示せばよい。本稿では与えられた長さ n の入力に対して、他に余分な領域を使用しないで $\log n$ を計算するような MTM の存在を示すことによって証明を行う。この場合 MTM が使用する領域量は、$\log n$ を表現するのに必要な領域量だけなので、明らかに $\log\log n$ である。

文献 [5](lemma3) で示されているアルゴリズム A' では、入力の長さ n に対し、n 回コイントスを行い、'1' の連の長さの最大値 N_n を出力する。このとき重複対数の法則を利用すれば、確率 1 で $\limsup \frac{N_n}{\log n} = 1$ となる [2]。

この場合、Nn は $\log n$ のごく近傍に収束することが予想されるが、$\log n$ よりも低いオーダーの関数を誤差として含む可能性があるので、これだけでは証明としては不十分である。実際後で示すように、本稿で得られた結果を利用すると、実は N_n は $\log n$ ではなくて $\log n - 1$ になる可能性が最も高く、そしてその値も約 0.24 であるにすぎない。つまり上記の証明だけでは $\frac{3}{4}$ 以上の確率で $\log\log n$ を構成できるかどうか、といった問題は解決していない。しかもこうした事実も上記の証明からは知ることができない。ここでは、解析が容易な別のアルゴリズムを用いることによって、より厳密な証明を与える。

定理 1 $\log\log n$ は MC-領域強構成可能である。

まず定理 1の証明に用いるアルゴリズム A_0 について示し、A_0 についての補題をいくつか示し、最後に定理の証明を行う。

入力テープが終了するまで以下を繰り返す。
- 入力ヘッドを 2 つ移動する。
- コイントスの結果が直前の結果と同じである間コイントスを繰り返し、その回数をカウントする。つまり連を一つ生成して、その長さを数える。
- それまでに生成した連の長さの最大値と比較し、必要なら更新する。

図 1: アルゴリズム A_0

アルゴリズム A_0　アルゴリズム A_0は、図 1のような手順に従う。ここで連を一つ生成したときの最後のトスの結果は次の連の先頭となることに注意する。A_0は入力の長さ n に対し$\frac{n}{2}$個の連を生成し、生成したすべての連の長さの最大値 L_nを計算するようなアルゴリズムである。A_0は明らかに L_nを記憶する領域しか使用しない PTM で実現できる。後は入力の長さ n に対して L_n、つまり連の長さの最大値が $\log n$ になることを示せばよい。

補題 **1** 任意の整数 lに対して、$L_n = l$になる確率は

$$\Pr\{L_n = l\} = \left(1 - \left(\tfrac{1}{2}\right)^l\right)^{\frac{n}{2}} - \left(1 - \left(\tfrac{1}{2}\right)^{l-1}\right)^{\frac{n}{2}}$$

となる。さらに十分大きな n について、次が成立する。ただし c は任意の定数とする。

$$\Pr\{L_n = \log n + c\} \sim e^{-\left(\frac{1}{2}\right)^{c+1}} - e^{-\left(\frac{1}{2}\right)^c} \qquad (*)$$

証明　ここで A_0が連を一つ生成するステップにおいて、連の長さが i になる確率は$\left(\frac{1}{2}\right)^i$である。(一つ目の連とそれ以外の連とでは別々に考察する必要があるが、結果は同じである。) したがって連の長さが l以下になる確率は、$\sum_{i=1}^{l}\left(\frac{1}{2}\right)^i = 1 - \left(\frac{1}{2}\right)^l$となり、$L_n$が l以下になる確率は$\left(1 - \left(\frac{1}{2}\right)^l\right)^{\frac{n}{2}}$となる。したがって任意の lに対して、L_nが lになる確率は

$$\Pr\{L_n = l\} = \Pr\{\ L_n \le l\} - \Pr\{L_n \le l-1\} = \left(1 - \left(\frac{1}{2}\right)^l\right)^{\frac{n}{2}} - \left(1 - \left(\frac{1}{2}\right)^{l-1}\right)^{\frac{n}{2}}$$

となる。ここで定数 Cに対して、$\lim_{n\to\infty}(1 - C\frac{1}{n})^n = e^{-C}$が成立することから、十分大きな n について、次が成立する。

$$\begin{aligned}\Pr\{L_n = \log n + c\} &= \left(1 - \left(\frac{1}{2}\right)^{\log n + c}\right)^{\frac{n}{2}} - \left(1 - \left(\frac{1}{2}\right)^{\log n + c - 1}\right)^{\frac{n}{2}} \\ &= \left(1 - \left(\frac{1}{2}\right)^{c+1}\left(\frac{2}{n}\right)\right)^{\frac{n}{2}} - \left(1 - \left(\frac{1}{2}\right)^{c}\left(\frac{2}{n}\right)\right)^{\frac{n}{2}} \sim e^{-\left(\frac{1}{2}\right)^{c+1}} - e^{-\left(\frac{1}{2}\right)^c}\end{aligned}$$

∎

ここで以下の証明で用いる公式を示す [8]。

公式 **1** ガンマ関数$\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$、プサイ関数$\psi(z) = \frac{d}{dz}\ln\Gamma(z)$ について、以下が成立する。

$$\frac{d^n}{dz^n}\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}(\ln t)^n dt = (\psi(z))^n\,\Gamma(z)$$

また$\Gamma(z)$ および$\psi(z)$ は $1 \le z \le 2$ の範囲では定数に収束し、$\Gamma(1) = 1, \psi(1) = -\gamma = -0.5772$ であることから、特に次が成立する。

$$\begin{aligned}\int_0^\infty e^{-t}\ln t\,dt &= \int_0^\infty e^{-t}t^{1-1}\ln t\,dt = \psi(1)\Gamma(1) = -\gamma = -0.5772 \\ \int_0^\infty e^{-t}(\ln t)^2 dt &= \int_0^\infty e^{-t}t^{1-1}(\ln t)^2 dt = (\psi(1))^2\,\Gamma(1) = (-\gamma)^2 = 0.3332\end{aligned}$$

補題 **2** 十分大きな n について、L_nの期待値μ_{L_n}は、$|\mu_{L_n} - \log n| < 1$ を満たし、L_nの分散$\sigma^2_{L_n}$は定数となる。

証明　補題 1より、十分大きな n について、$\Pr\{L_n = \log n + c\} \sim e^{-(\frac{1}{2})^{c+1}} - e^{-(\frac{1}{2})^{c}}$ が成立する。ここで $f(x) = \ln 2 \left(\frac{1}{2}\right)^x e^{-(\frac{1}{2})^x}$ と書けば、$\int_c^{c+1} f(x)dx = e^{-(\frac{1}{2})^{c+1}} - e^{-(\frac{1}{2})^{c}}$ である。したがって定義より期待値μ_{L_n}は、

$$\mu_{L_n} = \sum_{c=-\log n}^{\infty} (c + \log n) \int_c^{c+1} f(x)dx$$

となり、$\int_c^{c+1}(x-1)f(x)dx < c\int_c^{c+1} f(x)dx < \int_c^{c+1} x\, f(x)dx$ であることから、

$$\int_{-\log n}^{\infty} f(x)(x + \log n - 1)dx < \mu_{L_n} < \int_{-\log n}^{\infty} f(x)(x + \log n)dx$$

となる。ここで $t = \left(\frac{1}{2}\right)^x$ と変数変換すれば、$\int_{-\log n}^{\infty} f(x)dx = \int_0^n e^{-t}dt = 1 - e^{-n}$ となるので、

$$\int_{-\log n}^{\infty} f(x)(x + \log n)dx = (1 - e^{-n}) \log n - \frac{1}{\ln 2} \int_0^n e^{-t} \ln t\, dt$$

となる。よって公式 1と $\ln 2 = 0.6932$ であることから、十分大きな n に対して

$$\log n - 0.17 < \mu_{L_n} < \log n + 0.83$$

となる。同様に分散$\sigma^2_{L_n}$は、

$$\sigma^2_{L_n} = \sum_{c=-\log n}^{\infty} (c + \log n - \mu_{L_n})^2 \int_c^{c+1} f(x)dx$$

と定義される。ここで$\mu_{L_n} = \log n + \delta\ (-0.17 < \delta < 0.83)$ とおくことができるので、上記と同様に式を変形していくと、

$$\begin{aligned}
\sigma^2_{L_n} &= \sum_{c=-\log n}^{\infty} (c - \delta)^2 \int_c^{c+1} f(x)dx < \int_{-\log n}^{\infty} (x - \delta)^2 f(x)dx \\
&= \int_{-\log n}^{\infty} x^2 f(x)dx - 2\delta \int_{-\log n}^{\infty} x\, f(x)dx + \delta^2 \int_{-\log n}^{\infty} f(x)dx \\
&= \frac{1}{(\ln 2)^2} \int_0^n e^{-t} (\ln t)^2 dt - \frac{2\delta}{\ln 2} \int_0^n e^{-t} \ln t\, dt + \delta^2 (1 - e^{-n})
\end{aligned}$$

となる。よって公式 1と $\ln 2 = 0.6932$ であることから、十分大きな n に対して

$$\sigma^2_{L_n} < \delta^2 + 1.67\delta + 0.69 < 2.77$$

となる。 ∎

定理 1の証明　補題 2より十分大きな n に対して L_nの期待値は $\log n$ のごく近傍の値になり、分散は定数になることが示された。(実際に式 $(*)$ について計算を行うと、L_nに関する確率は、$L_n = \log n$ のときに最大値約 0.24 となり、また L_nが $\log n \pm 3$ の範囲の値をとる確率は 0.9 以上となる。)

ここで定数 kに対して、アルゴリズム A_0を k回起動し、その平均値を計算するアルゴリズムを A_0^kとする。このとき A_0^kは、A_0と同じ領域量で構成できることを示す。これには基本的には文献 [4] で示されているように、作業用テープを kトラックに分割して使用すればよい。ここでアルゴリズム A_0を k回起動すると、連を $k\frac{n}{2}$個生成することになるので、これらすべての連の中の最大値は明らかに非常に高い確率で $\log nk + \mathrm{O}(1) = \log n + \log k + C$となる。したがって $\log k$よりも十分大きな適当な長さまでを PTM の状態を利用して数え、それ以上の分を作業用テープに記憶するようにすれば、最終的な平均値以上の余分な領域を使用しない PTM を構成することができる。

以上の結果から、ある定数 k_0が存在し、$k > k_0$を満たす kに対し、アルゴリズム A_0^kは $\log n$ を$\frac{3}{4}$以上の確率で計算することがわかる。(実際に式 $(*)$ を利用して計算すると、$k_0 = 73$ であることが確認できる。) ∎

以下にアルゴリズム A_0についてのいくつかのコメントを示す。

A_0の停止性　文献 [5](lemma3) で示されているアルゴリズム A' は必ず停止する。本章で示したアルゴリズム A_0でも、図 2のテストステップを追加すれば停止性を保証することができる。このテストステップによってアルゴリズムが停止した場合は、生成した連の長さは少なくとも$\frac{n}{2}$を越えているので、正解を出力することはありえない。したがってこのテストステップが、正解を出力する場合の解析に影響を与えないのは明らか。

現在生成している連の長さを 1 つ増加させるごとに、以下のテストを行う:
- 現在の連の長さの数だけ入力ヘッドを左に移動し、入力テープの先頭に到着するかどうかテストする
- 上のステップで移動する前の元の位置にもどる
- 現在の連の長さの数だけ入力ヘッドを右に移動し、入力テープの終端に到着するかどうかテストする
- 上のステップで移動する前の元の位置にもどる

もし左右ともにヘッドが入力テープの外側に飛び出すならば、停止する。

図 2: テストステップ

for ($i = 1, 2, 3, \cdots$) do
入力ヘッドを 2^i回右に移動する。
途中で入力テープからはみ出してしまったときは $i-1$ を計算して停止する。
そうでなければ入力ヘッドを入力テープの先頭にもどす。

図 3: アルゴリズム A_1の流れ

A' と A_0の関係　A_0において、連を一つ構成するときの連の長さの期待値を$\bar{l}$とすると、定義より$\bar{l} = \sum_{i=1}^{\infty} \left(\frac{1}{2}\right)^i i$ である。このとき $l_k = \sum_{i=1}^{k} \left(\frac{1}{2}\right)^i i = 2 - \left(\frac{1}{2}\right)^k (k+1)$ とすれば、$\bar{l} = \lim_{k\to\infty} l_k = 2$ である。したがって十分大きな n に対して、A_0のコイントスの回数の期待値は n である。よって実はこのアルゴリズム A_0は、'0', '1' 両方の連を数えていることを除けば [5](lemma3) で示されているアルゴリズム A' と同じふるまいをする。逆にいうと A' は A_0の半分の個数の連を生成して、その長さの最大値を計算するアルゴリズムと同じふるまいをすることがわかる。したがって明らかに A' は $\log n$ でなく $\log n - 1$ を計算する確率が最も高く、その具体的な値は約 0.24 となる。

$\log\log\log n$ への拡張　アルゴリズム A_0を再帰的にくみあわせて、例えば次のようなアルゴリズムを構成する。

> 入力の長さ n に対し、$\frac{n}{2}$個の連を生成し、各連に対して、その長さの$\frac{1}{2}$個の連を生成し、それらの中の最大値を計算する。

これは、連の長さの期待値が 2 であることから、最終的にやはり$\frac{n}{2}$個の連を生成して最大値を求めることになるので、こうした方法では$\log\log n$ を計算することはできない。

4　$\log^{(k)} n$ の領域強構成可能性

本章ではまず、関数 $\log\log n$ が AI-領域強構成可能であることを示す。これはアルゴリズム A_0を用いて証明することも可能であるが、ここでは後で $\log^{(k)} n$ の証明に用いるアルゴリズムの元となるアルゴリズムを用いて証明を行う。

補題 3 関数 $\log\log n$ は AI-領域強構成可能である。

まず補題 3の証明に用いるアルゴリズム A_1について示し、A_1についての補題をいくつか示し、最後に補題 3の証明を行う。

アルゴリズム A_1　アルゴリズム A_1は無作為な長さ n の任意の入力に対して、余分な領域を使用しないで $\log n$ を計算する。このとき $\log n$ を表現するために $\log\log n$ 領域を使用することになる。A_1の全体の処理の流れは図 3のようになっている。ここで入力の長さ n と計算される $i-1$ との間には、$2^{i-1} \le n < 2^i$ という関係が成立するので、アルゴリズム A_1が停止したときに計算されている値は $\lfloor \log n \rfloor$ となる。A_1は、2^iを生成する部分をのぞけば $i-1$ を表現するための領域以外の領域を使用しない PTM で構成できる。したがって $\log i$ 領域で表現された i に対して、余分な領域を使用しないで 2^i回ヘッドを右に移動するアルゴリズムが存在することを示せばよい。

```
counter0,counter1 を 0 で初期化する。
counter0 < c である間以下を順に繰り返す。
  - biased_coin(i) = 0 なら counter0 ← counter0 + 1
                   = 1 なら counter1 ← counter1 + 1
  - counter1 = c ならヘッドを右に一つ移動し、counter1 ← 0 とする。
```

図 4: 平均をとるアルゴリズム

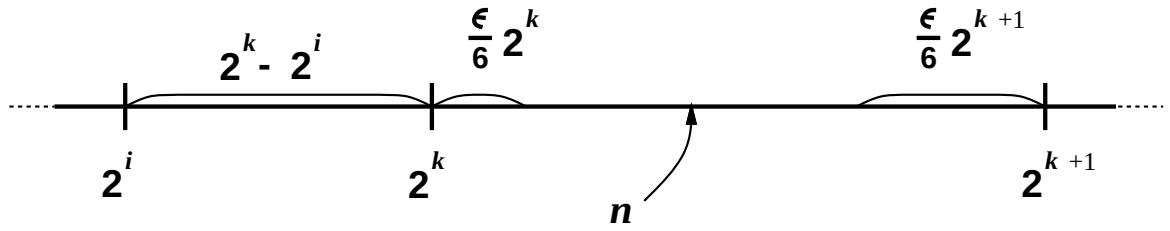

図 5: A_1が正解を出力する状況

補題 4 $i = 1, 2, 3 \cdots$に対し、i を表現するための領域以外の領域を使用せず、ヘッドを右に移動する回数の期待値μ_iが 2^i, 分散σ_i^2がμ_i^2以下であるようなアルゴリズムが存在する。さらに任意に与えられた $c > 0$ に対して、分散が$\frac{1}{c}\mu_i^2$以下となるように、このアルゴリズムを改良することができる。

証明 コインをi回投げて全部‘0’だったときだけ‘0’を返すサブルーチン biased_coin(i) を考える。biased_coin(i) は入力 i に対して確率$\left(\frac{1}{2}\right)^i$で‘0’、確率 $1 - \left(\frac{1}{2}\right)^i$で‘1’を返す。この biased_coin(i) を使って入力ヘッドを右に移動するサブルーチン run(i) を考える。すなわち run(i) は、biased_coin(i) の結果が‘1’であれば、入力ヘッドを右に 1 つ移動し、繰り返し biased_coin(i) を実行する。biased_coin(i) の結果が‘0’であれば停止する。ここで biased_coin(i), run(i) は i を表現する領域以外の領域を使用せずに構成できることに注意する。

ここで run(i) を実行したときにヘッドをl回右に移動する確率は、biased_coin(i) を繰り返し実行したときに返す値が‘1^l0’になる場合であるから、明らかに$\left(\frac{1}{2}\right)^i \left(1 - \left(\frac{1}{2}\right)^i\right)^l$である。したがって run(i) がヘッドを移動する回数の期待値$\bar{l_i}$は$\bar{l_i} = \sum_{l=0}^{\infty} \left(\frac{1}{2}\right)^i \left(1 - \left(\frac{1}{2}\right)^i\right)^l l = 2^i - 1$となり、分散$\sigma_i^2$は、$\sigma_i^2 = \sum_{l=0}^{\infty} \left(\frac{1}{2}\right)^i \left(1 - \left(\frac{1}{2}\right)^i\right)^l l^2 - \bar{l_i}^2 = 2^i(2^i - 1)$となる。したがって run(i) において biased_coin(i) の出力が‘0’であったとき、単に終了するのでなく、ヘッドを一つ余分に右に移動してから終了するようにしたものを新たに run(i) とすれば、これは期待値$\mu_i = 2^i$、分散$\sigma_i^2 = 2^i(2^i - 1) < \mu_i^2$を満たすアルゴリズムである。

最後に任意の正整数 c に対して run(i) の分散を改善する方法を示す。これは原理的には biased_coin(i) による‘1’の連を c 個作成し、その長さの平均の分だけヘッドを移動すればよい。実際には biased_coin(i) が‘0’を c 回返すまでに返してきた‘1’の個数を数えて、その‘1’の個数の$\frac{1}{c}$回ヘッドを右に移動すればよい。この処理の実現には、‘0’を数えるカウンタ counter0 と‘1’を数えるカウンタ counter1 とを用意すればよい。‘1’の個数の$\frac{1}{c}$回ヘッドを移動する部分は、counter1 で‘1’の個数を数え、counter1 の値が c になる度にヘッドを一つ移動して、counter1 の値をクリアすればよい。これらをまとめると、例えば図 4のように実現することができる。ここでそれぞれのカウンタの値は 0 以上 c 以下であるので、PTM の状態だけで実現できることに注意する。

以上から明らかに補題が成立する。 ∎

補題 3の証明　ここで $0 < \epsilon < 1$ をみたす任意のϵが与えられたとき、$c > \frac{168}{\epsilon^3}$を満たすような適当な定数 c を選び、この c によって改善されたアルゴリズム run(i) を組み込んで A_1を構成する。この A_1が、無作為な長さの入力に対して、$1 - \epsilon$以上の確率で正しい答を出力することを示せばよい。

ここで入力の長さを n とすると、$2^k \leq n < 2^{k+1}$を満たす自然数 kが存在する。ここで $i = 1, 2, \cdots, k-1, k$のときに生成したすべての連の長さが n を越えず、かつ $i = k + 1$ のときに生成した連の長さが n を越えれば、A_1は

正しい答を出力する。したがって以下の条件をすべて満たすように動作したときは、少なくとも A_1 は正しい答を出力する (図 5)。

- 入力の長さ n について、n が $2^k + \frac{\epsilon}{6}2^k \leq n < 2^{k+1} - \frac{\epsilon}{6}2^{k+1}$ の範囲にあった。(これを満たす確率を p_n とする。)
- $i = 1, 2, \cdots, k-2, k-1$ について、run(i) の出力が $2^i \pm (2^k - 2^i)$ の範囲にあった。(各 i に対してこれを満たす確率をそれぞれ p_i とする。)
- $i = k, k+1$ について、run(i) の出力が $2^i \pm \frac{\epsilon}{6}2^i$ の範囲にあった。(各 i に対してこれを満たす確率をそれぞれ p_i とする。)

以下でこれらの個々の確率について考える。

確率 p_n について、

$$p_n = \frac{(2^{k+1} - 2^k) - \frac{\epsilon}{6}2^k - \frac{\epsilon}{6}2^{k+1}}{2^{k+1} - 2^k} = 1 - \frac{\epsilon}{2}$$

となる。また $i = 1, 2, \cdots, k-2, k-1$ について、チェビシェフの不等式 [2] より、

$$p_i \geq 1 - \frac{\frac{1}{c}2^i(2^i - 1)}{(2^k - 2^i)^2} > 1 - \frac{1}{c(2^{k-i} - 1)^2} > 1 - \frac{1}{c(2^{k-i-1})^2} = 1 - \frac{4}{c}\frac{1}{4^{k-i}}$$

となり、したがって、$P_{k-1} = \prod_{i=1}^{k-1} p_i$ とおくと、

$$\begin{aligned} P_{k-1} &> \prod_{i=1}^{k-1}\left(1 - \frac{4}{c}\frac{1}{4^{k-i}}\right) = \prod_{i=1}^{k-1}\left(1 - \frac{4}{c}\frac{1}{4^i}\right) > P_{k-2}\left(1 - \frac{4}{c}\frac{1}{4^{k-1}}\right) \\ &> P_{k-2} - \frac{4}{c}\frac{1}{4^{k-1}} > P_{k-3} - \frac{4}{c}\left(\frac{1}{4^{k-1}} + \frac{1}{4^{k-2}}\right) > \cdots > 1 - \frac{4}{c}\left(\frac{1}{4^{k-1}} + \cdots + \frac{1}{4^2}\right) \\ &= 1 - \frac{4}{c}\frac{4}{3}\left(\frac{1}{4^2} - \frac{1}{4^k}\right) > 1 - \frac{4}{c}\frac{4}{3}\frac{1}{4^2} = 1 - \frac{1}{3c} > 1 - \frac{\epsilon^3}{504} > 1 - \frac{\epsilon}{14} \end{aligned}$$

となる。さらに $i = k, k+1$ について、チェビシェフの不等式より、

$$p_i \geq 1 - \frac{\frac{1}{c}2^i(2^i - 1)}{\left(\frac{\epsilon}{6}2^i\right)^2} > 1 - \frac{\frac{1}{c}(2^i)^2}{\left(\frac{\epsilon}{6}2^i\right)^2} = 1 - \frac{1}{\left(\frac{\epsilon}{6}\right)^2 c} > 1 - \frac{3\epsilon}{14}$$

となる。したがって A_1 が正解を出す確率 p について、

$$\begin{aligned} p &> p_1 p_2 \cdots p_{k-1} p_k p_{k+1} p_n = P_{k-1} p_k p_{k+1} p_n \\ &> \left(1 - \frac{\epsilon}{14}\right)\left(1 - \frac{3\epsilon}{14}\right)^2\left(1 - \frac{\epsilon}{2}\right) > \left(1 - \frac{\epsilon}{14}\right)\left(1 - \frac{3\epsilon}{7}\right)\left(1 - \frac{\epsilon}{2}\right) > \left(1 - \frac{\epsilon}{2}\right)\left(1 - \frac{\epsilon}{2}\right) > 1 - \epsilon \end{aligned}$$

が成立する。 ∎

定理 2 任意の正整数 k について、$\log^{(k)} n$ は平均的に領域強構成可能である。

証明 $k < 3$ のときは明らか。まず $k = 3$ のときを考える。ここで補題 4 で構成した run(i) に手を加え、ヘッドを移動する変わりにコイントスを行なうようにする。そしてそのコイントスの結果がすべて '0' だったときだけ '0' を返すサブルーチン $\mathtt{biased_coin}^2\mathtt{(i)}$ を考える。このときコイントスの回数の期待値は 2^i となるから、$\mathtt{biased_coin}^2\mathtt{(i)}$ は平均的に、入力 i に対して確率 $\left(\frac{1}{2}\right)^{2^i}$ で '0'、確率 $1 - \left(\frac{1}{2}\right)^{2^i}$ で '1' を返す。したがって $\mathtt{biased_coin}^2\mathtt{(i)}$ を使用してヘッドの移動を行なうサブルーチン $\mathtt{run}^2\mathtt{(i)}$ を考えれば、$\mathtt{run}^2\mathtt{(i)}$ がヘッドを l 回移動する確率は $\left(1 - \left(\frac{1}{2}\right)^{2^i}\right)^l \left(\frac{1}{2}\right)^{2^i}$ となり、移動回数の期待値は補題 4 と同様に、$\sum_{l=0}^{\infty}\left(\frac{1}{2}\right)^{2^i}\left(1 - \left(\frac{1}{2}\right)^{2^i}\right)^l l = 2^{2^i} - 1$ となる。したがってヘッドを一つ余分に移動するようにしたものを新たに $\mathtt{run}^2\mathtt{(i)}$ とおき、これを用いてアルゴリズム A_1^2 を構成すれば、A_1^2 は平均的に $\log\log\log n$ 領域を用いて $\log\log n$ を計算する。したがって $\log^{(3)} n$ は平均的に領域強構成可能である。

さらにこの方法を繰り返し用いれば、平均的に $\log^{(4)} n$ 領域を用いて $\log^{(3)} n$ を計算するアルゴリズム A_1^3 を構成することができ、$\log^{(4)} n$ は平均的に領域強構成可能となる。以下同様に任意の定数 $k > 3$ に対してアルゴリズム A_1^k を構成することができ、これは平均的に $\log^{(k+1)} n$ 領域を使用するので、定理が成立する。 ∎

謝辞

今研究のきっかけを与えてくれた三重大学の陳致中助手と、適切な助言を与えてくれた電気通信大学の戸田誠之助助教授に深謝します。

参考文献

[1] H. Alt and K. Mehlhorn. Lower bounds for the space complexity of context-free recognition. *Proceedings, Int. Colloq. Automate Lang. & Programm. '76*, pages 338–351, 1976.

[2] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley & Sons, Inc., second edition, 1957.

[3] John Gill. Computational complexity of probabilistic turing machines. *SIAM J. Comput.*, 6(4):675–695, Dec. 1977.

[4] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, 1979.

[5] Marek Karpinski and Rutger Verbeek. On the monte carlo space constructible functions and separation results for probabilistic complexity classes. *Information and Computation*, 75:178–189, 1987.

[6] Desh Ranjan, Richard Chang, and Juris Hartmanis. Space bounded computations: review and new separation results. *Theoretical Computer Science*, 80:289–302, 1991.

[7] Andrzej Szepietowski. Some notes on strong and weak $\log\log n$ space complexity. *Information Processing Letters*, 33:109–112, Nov. 1989.

[8] 森口繁一他. 岩波数学公式, volume III 特殊関数. 岩波書店, 1960.