

# 言語の階層を決めているのは、記憶の量ではなく型である

—  $10^{80}$  個の有限記憶より、1 本の無限スタックのほうが強い — 記憶を一段増やすごとに、判定の手間が  $n \rightarrow n^3 \rightarrow$  指数  $\rightarrow$  不可能へ移る — [論文296]

山岸 勇輝 (Yuuki YAMAGISHI)

Yamagishi Complexity Torus Lab, Japan | 山岸勇輝ラボ

2026年8月

— *From thinking to syncing.* —

## 要旨

チョムスキー階層は、言語を四段に分ける。段を分けているのは何かと問えば、ふつう「複雑さ」と答える。本稿が問うのは、その複雑さの正体である——答は、認識する機械の記憶の型である。新しい数学定理も新しい法則も主張しない。第一に、四段を記憶の型で並べる。記憶なし／スタック 1 本／入力に比例したテープ／無限のテープ（第2節）。第二に、これが本稿の芯である。有限オートマトンの状態を  $10^{80}$  個にしても  $a^n b^n$  は認識できず、スタック 1 本なら認識できる（第3節）。第三に、同値類が可算無限になる。 $a^i$  と  $a^j$  は  $b^i$  を付ければ区別でき、有限個の状態では原理的に足りない（第3節）。第四に、一段上がるごとに手間が跳ぶ。 $n = 1000$  で  $O(n)$  と  $O(n^3)$  の差は  $10^6$  倍、 $2^n$  は  $10^{301}$  である（第4節）。第五に、最上段では手間が定義されない。句構造言語の所属判定は判定不能である（第4節）。第六に、分離子は「括弧の対応が取れるか」である。深さを固定すれば有限で足りるが、任意の深さには無限が要る（第5節）。

本稿の射程 (scope note) : 新しい数学定理も新しい法則も主張しない——チョムスキー階層、マイヒル＝ネロードの定理、ポンピング補題、 $O(n^3)$  の構文解析、判定不能性は、いずれも標準的である。形式言語理論を作らない——使うのは一つの鳩の巣論法と、三つの手間の比較だけである。定理を証明しない——マイヒル＝ネロードもポンピング補題も引くだけであり、証明には立ち入らない。自然言語を論じない——人間の言語がどの段にあるかは一切扱わない。形式言語の階層だけを扱う。計算複雑性を論じない—— $P$  と  $NP$  にも、 $PSPACE$  の詳細にも立ち入らない。段ごとの手間の桁だけを並べる。 $O(n^3)$  が最良だと言わない——文脈自由言語の構文解析は行列乗算と同じ指数まで下げられることが知られている。 $n^3$  は古典的な上界である。四段が唯一の分け方だと言わない——決定性文脈自由、線形言語など中間の族が多数ある。本稿は最も粗い四段だけを扱う。比喻の「文法」と混同しない——論文87は作用原理を「物理法則の文法」と呼び、論文92は微分形式を「共通のアルファベット」と呼んだ。どちらも比喻である。本稿は文字どおりの形式文法を扱うので、同じ語だが対象が違う。既刊との関係：論文179は「構成できない」に別根があると示し、チューリングの停止問題に触れた——本稿の最上段の判定不能はその根の一つであり、ここでは階層の頂点として位置づける。論文163は「良い符号が在る」と「これが良い符号だ」を分けた——本稿も「認識できる」と「どれだけの手間で認識できる」を分ける。論文112は「同じ韻を踏む六つの別根」を数えた——本稿の四段は「複雑さ」という一語の下の中の四つの別物である。論文100は「一語・複数の別根」を体系の規律として書いた——本稿はその規律を形式言語に当てる。加えたのは四段を記憶の型で並べ直したこと、有限記憶を  $10^{80}$  個にしても  $a^n b^n$  に届かないと数で示したこと、 $n = 1000$  での手間の比を  $10^6$  倍と計算したこと、分離子を「括弧の対応」に置いたことである。

## 1. 記法と問い

設定：形式言語は、それを認識する機械によって四段に分かれる（チョムスキー階層）。

問い：段を分けているのは何か。「複雑さ」と言えば答になるが、その複雑さは何の複雑さか。

### 観察 1.1

本稿の答は記憶の型である。速さでも、規則の個数でも、状態の数でもない——何を覚えておけるかの形である。

### 観察 1.2

正直に断っておく——体系はこれまで「文法」を比喻として使ってきた。論文87は作用原理を「物理法則の文法」と呼び、論文92は微分形式を「共通のアルファベット」と呼んだ——本稿はそれらとは違い、文字どおりの形式文法を扱う。同じ語だが、対象が違う。

## 2. 四段を、記憶の型で並べる

検算 2.1 :

| 言語族  | 機械         | 記憶         | 所属判定の手間  |
|------|------------|------------|----------|
| 正規   | 有限オートマトン   | 記憶なし       | $O(n)$   |
| 文脈自由 | プッシュダウン    | スタック 1 本   | $O(n^3)$ |
| 文脈依存 | 線形拘束オートマトン | 入力に比例したテープ | PSPACE   |
| 句構造  | チューリング機械   | 無限のテープ     | 判定不能     |

### 観察 2.2

左から二列目（機械の名前）は定義に過ぎない。効いているのは三列目である——機械が違うのではなく、記憶が違う。

### 観察 2.3

三列目を上から読むと、無し → 一本 → 有界 → 無限である。増えているのは量ではなく型——「記憶なし」と「スタック 1 本」の間は、量では埋まらない（第3節）。

### 観察 2.4

右端の列は、最後だけ数でない。 $O(n)$ 、 $O(n^3)$ 、PSPACE と来て、最後は判定不能——手間が大きいのではなく、手間という量が定義されない。

## 3. 芯： $10^{80}$ 個の有限記憶より、1 本の無限スタック

設定： $a^n b^n$  という言語（ $a$  を  $n$  個、続けて  $b$  を  $n$  個）を考える。

検算 3.1：マイヒル＝ネロードの区別。 $i \neq j$  のとき、 $a^i$  と  $a^j$  は後ろに  $b^i$  を付ければ区別できる。

| 文字列                      | 言語に属するか |
|--------------------------|---------|
| $a^i b^i$                | 属する     |
| $a^j b^i$ ( $j \neq i$ ) | 属さない    |

### 観察 3.2

したがって  $a^0, a^1, a^2, \dots$  はたがいにすべて区別されねばならない。同値類は可算無限個——有限個の状態では原理的に足りない。

検算 3.3：状態を増やしてみる。

| 状態の個数 $k$ | 区別できる $a^n$              | 必ず失敗する入力                  |
|-----------|--------------------------|---------------------------|
| 2         | $a^0$ から $a^1$           | $a^2 b^2$                 |
| 10        | $a^0$ から $a^9$           | $a^{10} b^{10}$           |
| $10^6$    | $a^0$ から $a^{10^6-1}$    | $a^{10^6} b^{10^6}$       |
| $10^{80}$ | $a^0$ から $a^{10^{80}-1}$ | $a^{10^{80}} b^{10^{80}}$ |

### 観察 3.4

これが本稿の芯である。状態を観測可能な宇宙の原子の数にしても、その先で必ず破れる——一方プッシュダウンはスタック 1 本で  $a^n b^n$  を認識する。

### 観察 3.5

$10^{80}$  個の有限記憶 < 1 本の無限スタック——量ではなく型が効いていることの、最も短い言い方である。

観察 3.6

スタックがしているのは「数える」ことである。*a*を見るたび積み、*b*を見るたび降ろす——有限オートマトンには「数える」という操作が無いのではなく、上限のある数え方しか無い。

検算 3.7：では、スタック 1 本で足りるか。 $a^n b^n c^n$  を考える。

| 言語            | 認識できる機械              |
|---------------|----------------------|
| $a^n b^n$     | プッシュダウン（スタック 1 本）    |
| $a^n b^n c^n$ | プッシュダウンでは不可（線形拘束が要る） |

観察 3.8

スタックは *a* と *b* の照合で空になってしまい、*c* を照合する材料が残らない。記憶を一段増やすごとに、扱える言語が真に増える——どの段も、飛ばせない。

4. 一段上がるごとに、手間が跳ぶ

検算 4.1：入力長 *n* に対する手間を並べる。

| <i>n</i> | <i>n</i> （正規）   | <i>n</i> <sup>3</sup> （文脈自由） | 2 <sup><i>n</i></sup> （指数） |
|----------|-----------------|------------------------------|----------------------------|
| 10       | 10              | 1000                         | 1.024 × 10 <sup>3</sup>    |
| 100      | 100             | 10 <sup>6</sup>              | 1.268 × 10 <sup>30</sup>   |
| 1000     | 1000            | 10 <sup>9</sup>              | 10 <sup>301</sup>          |
| 10000    | 10 <sup>4</sup> | 10 <sup>12</sup>             | 10 <sup>3010</sup>         |

観察 4.2

*n* = 1000 で、文脈自由は正規の 10<sup>6</sup> 倍かかる。同じ長さの文を、百万倍の手間で読む——括弧を数えられるようになる代償である。

### 観察 4.3

$2^{1000} = 10^{301}$  は、観測可能な宇宙の原子 ( $10^{80}$ ) の  $10^{221}$  倍である。原理的に可能でも、実行できない——「できる」と「できる手間で」は別である（論文163 の区別と同じ形）。

### 観察 4.4

正直に書くべき点—— $O(n^3)$  は古典的な上界である。文脈自由の構文解析は行列乗算と同じ指数まで下げられる—— $n^3$  は「段が上がった」ことを示す数であって、最良の値ではない。

### 観察 4.5

最上段では、この表に行が書けない。句構造言語の所属判定は判定不能であり、どんな  $n$  に対しても手間が定義されない——論文179 が扱った「構成できない」の別根の一つが、ここで階層の頂点として現れる。

## 5. 分離子：括弧の対応が取れるかどうか

### 観察 5.1

分離子——何を動かすと、正規と文脈自由が入れ替わるか。括弧の深さに上限があるかどうかである。

検算 5.2：

| 扱う括弧      | 要る状態の数 | 段    |
|-----------|--------|------|
| 深さ 1 まで   | 2 個    | 正規   |
| 深さ 5 まで   | 6 個    | 正規   |
| 深さ 100 まで | 101 個  | 正規   |
| 任意の深さ     | 無限個    | 文脈自由 |

### 観察 5.3

最下段以外はすべて正規である。深さ 100 の括弧なら正規表現で書ける——境界は「深い」ところではなく、「上限が無い」ところにある。

### 観察 5.4

これは実務にも効く。正規表現で HTML を解析できないのはタグが任意の深さに入れ子になれるからであり、深さを 100 に制限すれば書ける——禁じているのは複雑さではなく、上限の不在である。

### 観察 5.5

論文112 は「同じ韻を踏む六つの別根」を数えた。「複雑な言語」という一語の下にも、四つの別物がある——分けているのは複雑さの度合いではなく、記憶の型という質の違いである。

### 観察 5.6

論文100 は体系の規律を「一語・複数の別根」と書いた。本稿はその規律を形式言語に当てた——比喻としてではなく、文字どおりの文法に対して。

## 6. 点呼と結び

1. 四段は記憶の型で並ぶ（検算2.1）：無し／1本／有界／無限。
2. 最後だけ数でない（観察2.4）：判定不能。
3. 同値類が可算無限（検算3.1）： $a^i b^i$  と  $a^j b^i$  で区別。
4.  $10^{80}$  状態でも破れる（検算3.3）： $a^{10^{80}} b^{10^{80}}$ 。
5.  $10^{80}$  の有限 < 1本の無限（観察3.5）：型が効く。
6.  $a^n b^n c^n$  はスタック 1本で不可（検算3.7）：段は飛ばせない。
7.  $n = 1000$  で  $10^6$  倍（検算4.1）：括弧を数える代償。
8.  $2^{1000} = 10^{301}$ （観察4.3）：原子の  $10^{221}$  倍。
9. 深さ 100 なら正規（検算5.2）：状態 101 個。

## 10. 境界は上限の不在（観察5.4）：深さではない。

結び：言語の階層を決めているのは、記憶の量ではなく型である。四段は記憶なし／スタック 1 本／入力に比例したテープ／無限のテープと並び、機械の名前ではなく記憶の形が段を分けている。最も短い言い方はこうである——有限オートマトンの状態を  $10^{80}$  個、すなわち観測可能な宇宙の原子の数にしても  $a^n b^n$  は認識できず、プッシュダウンはスタック 1 本で認識する。 $a^i$  と  $a^j$  は  $b^i$  を付ければ区別でき、同値類が可算無限個になるからである。そして一段上がるごとに手間が跳ぶ—— $n = 1000$  で  $O(n)$  と  $O(n^3)$  の差は  $10^6$  倍、 $2^n$  は  $10^{301}$ 、最上段では手間という量そのものが定義されない。分けるものは一つ——括弧の深さに上限があるかどうか。深さ 100 までなら状態 101 個の有限オートマトンで足りる。禁じているのは深さではなく、上限の不在である。

## 参考文献

---

1. Chomsky, N. (1956). Three Models for the Description of Language. *IRE Transactions on Information Theory*, 2(3).
  2. Myhill, J. (1957). *Finite Automata and the Representation of Events*. WADD Technical Report 57-624.
  3. Nerode, A. (1958). Linear Automaton Transformations. *Proceedings of the American Mathematical Society*, 9(4).
  4. Bar-Hillel, Y., Perles, M. and Shamir, E. (1961). On Formal Properties of Simple Phrase Structure Grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14.
  5. Hopcroft, J. E., Motwani, R. and Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation*. Boston: Addison-Wesley.
  6. Valiant, L. G. (1975). General Context-Free Recognition in Less Than Cubic Time. *Journal of Computer and System Sciences*, 10(2).
  7. 山岸勇輝 (2026). 物理法則の記述言語としての作用原理——粒子から場へ：一つの文法であって、一つの方程式ではない [論文87]. Zenodo. 10.5281/zenodo.21747287
  8. 山岸勇輝 (2026). 王道の交差——論文63～99の構造的解釈センサス、そして「一語・複数の別根」という一つの規律 [論文100]. Zenodo. 10.5281/zenodo.21758864
  9. 山岸勇輝 (2026). 「良い符号が在る」と「これが良い符号だ」は、別の文である [論文163]. Zenodo. 10.5281/zenodo.22092677
  10. 山岸勇輝 (2026). 「構成できない」にも別根がある [論文179]. Zenodo. 10.5281/zenodo.22098419
- 

— From thinking to syncing. —

山岸 勇輝 (Yuuki YAMAGISHI)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)