

線形プッシュダウン・木オートマトン

藤芳 明生 笠井 琢美

電気通信大学大学院情報工学専攻
E-mail: fujiyo-a@calvyn.cs.uec.ac.jp

1 はじめに

最近の自然言語の形式化に関する研究の関心は、文脈自由言語より大きな言語のクラスに向いている。文脈自由文法より大きな言語のクラスを生成する文法の中で、最近、Tree Adjoining Grammar (TAG)[8]が注目されている。TAGにより生成される文字列言語を $O(n^6)$ 時間[13]や $O(M(n^2))$ 時間[10]で認識するような、高速な認識アルゴリズムが知られている。また、自然言語の形式化としてそれぞれ独立して研究されていた、Head Grammar, Combinatory Categorical Grammar, Linear Indexed Grammar等により生成される文字列言語のクラスがTAGのそれと一致していることが示されたことは、注目すべきことである[14]。

TAGは木を生成する文法であり、文脈自由木文法[11]に制限を加えたものと見なすことができる。文脈自由木文法は、文脈自由文法の生成規則の右辺を文字列から木に置き換えることにより得られる文法である。しかしながら、TAGは複雑な制限が加わった文脈自由木文法となっているため、そのままでは形式的性質を研究することは容易ではない。そこで、TAGをもう一度文脈自由木文法の立場から見直し、より自然な制限を文脈自由木言語に与えることにより、Spine Grammarという、TAGと同じ文字列言語のクラスを生成する文法が定義された[5]。

文脈自由木言語を受理する木オートマトンとして、プッシュダウン・木オートマトンがIrène Guessarian [6]によって考案された。このプッシュダウン・木オートマトンは通常のプッシュダウン・オートマトン[7]とトップダウン型・木オートマトン[1][2][12]を組み合わせたものと見なすことができる。トップダウン型・木オートマトンとは、入力を根から読み始め、葉に向かいながら、分岐においては自分自身を複製しながら読み進んでいくようなタイプの木オートマトンである。Guessarianはプッシュダウン・木オートマトンにより受理される木言語のクラスと文脈自由木言語のクラスが一致することを示した。

本研究は、このプッシュダウン・木オートマトンに線形という制限をつけた線形プッシュダウン・木オートマトンを考える。線形という条件は、オートマトンが自分自身を複製する際に、それらの中の1つにのみプッシュダウン記憶の中身をつたえ、ほかのもののプッシュダウン記憶は初期化されていなくてはならないという条件である。そして、線形プッシュダウン・木オートマトンにより受理される木言語のクラスは、Spine Grammarにより生成される木言語のクラスと一致することを示す。このことより、Spine Grammarにより生成される木言語のクラスは、認識可能木言語のクラスと文脈自由木言語のクラスの間に位置する自然な木言語のクラスであることが分かる。

2 諸定義

定義 2.1 \mathcal{N}_+ を正整数全体からなる集合とする。木の台集合とは、次を満たす $(\mathcal{N}_+)^*$ の有限部分集合 D である。

- $d \in D$ かつ $d = d' \cdot d''$, $d', d'' \in (\mathcal{N}_+)^*$ ならば, $d' \in D$.
- $i, j \in \mathcal{N}_+$ に対し, $i \leq j$ かつ $d \cdot j \in D$ ならば, $d \cdot i \in D$ となる。

D の元を節点と呼ぶ。特に, $d \cdot 1 \notin D$ を満たす節点 d を葉と呼ぶ。モノイド $(\mathcal{N}_+)^*$ の単位元を λ で表し, 節点 λ を根と呼ぶ。葉でも根でもない節点を内部節点と呼ぶ。

定義 2.2 D を木の大集合とし, $d \in D$ とする. 根から d への道とは節点の集合 $\{d' \in D \mid d' \text{は } d \text{の前部分語}\}$ である.

定義 2.3 Σ を有限アルファベットとする. Σ 上の木とは, 関数 $\alpha : D \rightarrow \Sigma$ のことである. ただし, D は木の台集合である. D_α で木 α に対する木の台集合, つまり, 関数 α の定義域を表すものとする.

定義 2.4 Σ 上の木 α , 節点 $d \in D_\alpha$ に対し, $\alpha/d = \{(d', a) \in (\mathcal{N}_+)^* \times \Sigma \mid (d \cdot d', a) \in \alpha\}$ とする. α/d を α の節点 d における部分木と呼ぶ.

定義 2.5 ε を特別な記号とする.

定義 2.6 Σ を有限アルファベットとする. Σ は特別な記号 ε を含んでいてもよいものとする. 木 α の *yield*を, Σ 上の木全体の集合から Σ^* への関数として, 次のように定義する.

- $D_\alpha = \{\lambda\}$ かつ $\alpha(\lambda) \neq \varepsilon$ ならば, $\text{yield}(\alpha) = \alpha(\lambda)$.
- $D_\alpha = \{\lambda\}$ かつ $\alpha(\lambda) = \varepsilon$ ならば, $\text{yield}(\alpha) = \lambda$. ここで, λ はモノイド Σ^* の単位元.
- $D_\alpha \neq \{\lambda\}$ ならば, $\text{yield}(\alpha) = \text{yield}(\alpha/1) \cdots \text{yield}(\alpha/n)$. ここで, $n = \max\{i \in \mathcal{N}_+ \mid i \in D_\alpha\}$.

定義 2.7 階層化アルファベットとは, 対 (Σ, r) のことである. ここで, Σ は有限アルファベットで ε を含んでいてもよい. また, r は Σ から \mathcal{N} (自然数全体)への関数で, もし, $\varepsilon \in \Sigma$ ならば, $r(\varepsilon) = 0$ である. $\Sigma_n = r^{-1}(n)$ とする. $r(a)$ を記号 a のランクと呼ぶ.

定義 2.8 Σ を階層化アルファベットとする. すべての $d \in D_\alpha$ に対し, $r(\alpha(d)) = \max\{i \in \mathcal{N}_+ \mid d \cdot i \in D_\alpha\}$ であるような Σ 上の木 α 全体からなる集合を T_Σ と書きあらわす.

定義 2.9 Σ を階層化アルファベット, I を Σ と共通部分を持たない有限集合とする. I をインデックスの集合とする Σ 上のインデックス付き木とは, 次を満たす関数 $\alpha : D_\alpha \rightarrow \Sigma \cup I$ である. 任意の $d \in D_\alpha$ に対し, $\alpha(d) \in \Sigma$ ならば, $r(\alpha(d)) = \max\{i \in \mathcal{N}_+ \mid d \cdot i \in D_\alpha\}$, $\alpha(d) \in I$ ならば, d は葉となっている. ただし, D_α は木の台集合であり, Σ は階層化アルファベットである. I をインデックスの集合とする Σ 上のインデックス付き木全体からなる集合を $T_\Sigma(I)$ とかく.

定義 2.10 ここで, 木 $\alpha \in T_\Sigma \cup T_\Sigma(I)$ に対する表現を, Σ および I の元と括弧を用い次のように定義する.

- $\alpha(\lambda) = a \in \Sigma_0$ ならば, 木 α を a で表す.
- $\alpha(\lambda) = \iota \in I$ ならば, 木 α を ι で表す.
- $\alpha(\lambda) = b \in \Sigma_n$ ($n \geq 1$) かつ, それぞれの $1 \leq i \leq n$ に対し α/i の表現が α_i ならば, 木 α を $b(\alpha_1 \cdots \alpha_n)$ で表す.

定義 2.11 $\alpha, \beta \in T_\Sigma$, $d \in D_\alpha$ とする. このとき, $\alpha(d \leftarrow \beta) = \{(d', a) \mid (d', a) \in \alpha \text{ and } d \text{は } d' \text{の前部分語でない}\} \cup \{(d \cdot d'', b) \mid (d'', b) \in \beta\}$ と定める. つまり, 木 $\alpha(d \leftarrow \beta)$ は, 部分木 α/d を β で置き換えた結果である.

定義 2.12 $X = \{x_1, x_2, \dots\}$ を固定された変数の集合とする. $X_n = \{x_1, x_2, \dots, x_n\}$ で, X の先頭の n 個の元からなる部分集合を表す. $X_0 = \emptyset$ である. また, X_1 の場合, その唯一の元 x_1 を x で表してもよいものとする.

定義 2.13 $\alpha \in T_\Sigma(X_n)$, $\beta_1, \dots, \beta_n \in T_\Sigma(X)$ とする. ここで, 代入の概念を導入する. 木 α の x_i をラベルとするそれぞれの節点に β_i を代入した結果を $\alpha[\beta_1 \cdots \beta_n]$ で表し, 次のように定義する.

- $\alpha = a \in \Sigma_0$ ならば, $\alpha[\beta_1 \cdots \beta_n] = a$.
- $\alpha = x_i \in X_n$ ならば, $x_i[\beta_1 \cdots \beta_n] = \beta_i$.
- $\alpha = b(\alpha_1 \cdots \alpha_k)$ かつ $k \geq 1$ ならば, $\alpha[\beta_1 \cdots \beta_n] = b(\alpha_1[\beta_1 \cdots \beta_n] \cdots \alpha_k[\beta_1 \cdots \beta_n])$.

3 文脈自由木文法と Spine Grammar

定義 3.1 文脈自由木文法とは, $G = (N, \Sigma, P, S)$ のことである。ここで,

- N と Σ は, それぞれ非終端記号と終端記号の有限集合である。 N と Σ は共通部分を持たないものとする。
- P は生成規則の有限集合で, それぞれ次のいずれかの形をしている。 $A(x_1 \cdots x_n) \rightarrow \alpha$, $n \geq 1$, $A \in N_n$, $\alpha \in T_{N \cup \Sigma}(X_n)$ 。 または, $A \rightarrow \alpha$, $A \in N_0$, $\alpha \in T_{N \cup \Sigma}$ 。
- S は N_0 の特別な元で, 開始記号である。

定義 3.2 与えられた文脈自由木文法 G に対し, $T_{N \cup \Sigma}(X) \times T_{N \cup \Sigma}(X)$ 上の関係 \xrightarrow{G} を次のように定義する。 $\alpha \in T_{N \cup \Sigma}(X)$, $d \in D_\alpha$ に対し, $\alpha/d = A$, $A \in N_0$, $A \rightarrow \beta \in P$ ならば, $\alpha \xrightarrow{G} \alpha(d \leftarrow \beta)$, また, $\alpha/d = A(\alpha_1 \cdots \alpha_n)$, $A \in N_n$, $\alpha_1, \dots, \alpha_n \in T_{N \cup \Sigma}$, $A(x_1 \cdots x_n) \rightarrow \beta \in P$ ならば, $\alpha \xrightarrow{G} \alpha(d \leftarrow \beta[\alpha_1 \cdots \alpha_n])$ とする。

n ステップの導出とは, 木の有限列 $\alpha_0, \dots, \alpha_n \in T_{N \cup \Sigma}(X)$ のことである。ただし, $n \geq 0$, $\alpha_0 \xrightarrow{G} \alpha_1 \xrightarrow{G} \cdots \xrightarrow{G} \alpha_n$ である。導出 $\alpha_0, \dots, \alpha_n$ が存在するとき, $\alpha_0 \xrightarrow{*} \alpha_n$ と書く。

後の証明における便宜上, 導出を変数付きの木の有限列と定義した。しかし, 言語を定義するだけならば, \xrightarrow{G} を $T_{N \cup \Sigma} \times T_{N \cup \Sigma}$ 上の関係, 導出を $T_{N \cup \Sigma}$ の元の有限列と定義するだけで十分である。

導出 $\alpha_0, \dots, \alpha_n$ がアウトサイド・イン (OI) であるとは, すべての $i \in \{0, \dots, n-1\}$ に対し, α_{i+1} から α_i 得られるとき生成規則が節点 $d \in D_{\alpha_i}$ に適応されたとする, d よりも根に近いすべての節点は既に終端記号に書き換えられていなければならないという条件を満たしていることである。 $\alpha \xrightarrow{*} \alpha'$ であるならば, α より $\alpha' \in T_\Sigma$ を得る OI-導出が必ず存在することが知られている [3]。

定義 3.3 G を文脈自由木文法とする。 G により生成される言語とは, 集合 $L(G) = \{\alpha \in T_\Sigma \mid S \xrightarrow{*} \alpha\}$ のことである。 また, G により生成される文字列言語とは, $L_S(G) = \{\text{yield}(\alpha) \mid \alpha \in L(G)\}$ のことである。

定義 3.4 G と G' を文脈自由木文法とする。 G と G' が $L(G) = L(G')$ であるとき, 等価であるという。 また, G と G' が $L_S(G) = L_S(G')$ であるとき, 弱等価であるという。

これより, Spine Grammar を定義するために, 背骨型という文脈自由木文法の制限を定義する。 そのため, 各非終端記号にヘッドという自然数を新たに割り当てる。

定義 3.5 ヘッド割り当て階層化アルファベットとは, (N, r, h) である。 ここで, (N, r) は階層化アルファベット。 h は N から \mathcal{N} (自然数全体) への関数で, 各 $A \in N$ に対し, $r(A) \geq 1$ ならば, $1 \leq h(A) \leq r(A)$, また, $h(A) = 0$ となっている。 $h(A)$ を記号 A のヘッドと呼ぶ。

定義 3.6 $G = (N, \Sigma, P, S)$ を文脈自由木文法とする。 ただし, N がヘッド割り当て階層化アルファベットとなっているものとする。 $n \geq 1$ について, 生成規則 $A(x_1 \cdots x_n) \rightarrow \alpha \in P$ が次を満たすとき, 背骨型であるという。

- α には, $x_{h(A)}$ をラベルとする葉が正確に一つ存在する。 根からこの葉までの道を, α の背骨と呼ぶ。
- 節点 $d \in D_\alpha$ において, d が背骨上にあり $\alpha(d) = B \in N$ であるならば, 節点 $d \cdot h(B)$ も背骨上にある。
- $X_n - \{x_{h(A)}\}$ をラベルとするすべての節点は, 背骨上にある節点の直接の子供となっている。

文脈自由木文法 $G = (N, \Sigma, P, S)$ が背骨型であるとは, すべての $n \geq 1$ なる生成規則 $A(x_1 \cdots x_n) \rightarrow \alpha \in P$ が, 背骨型であることである。 背骨型文脈自由木言語を Spine Grammar と呼ぶことにする。

定義 3.7 $G = (N, \Sigma, P, S)$ を Spine Grammar とする。 G が標準形であるとは, G が次を満たすことである。

- すべての $A \in N$ に対し, $r(A) = 0$ または, $r(A) = 1$ である.
- $A \in N_0$ に対し, $A \rightarrow \alpha \in P$ ならば, $\alpha = a$, $a \in \Sigma_0$ または, $\alpha = B(C)$, $B \in N_1$, $C \in N_0$ である.
- $A \in N_1$ に対し, $A(x) \rightarrow \alpha \in P$ ならば, $\alpha = B_1(\dots(B_m(x))\dots)$, $m \geq 0$, $B_1, \dots, B_m \in N_1$ であるか, または, $\alpha = b(C_1 \dots C_n)$, $n \geq 1$, $b \in \Sigma_n$, C_1, \dots, C_n は正確に一つだけ $C_i = x$ となっているが他はすべて N_0 の元である.

文献 [5] において, Spine Grammar の標準形と Tree Adjoining Grammar [14] との関係が示されている.

定理 1 任意の Spine Grammar に対し, それと等価な標準形の Spine Grammar が存在する.

定理 2 Tree Adjoining Grammar により生成される文字列言語のクラスは, Spine Grammar により生成される文字列言語のクラスと一致する.

4 線形プッシュダウン・木オートマトン

定義 4.1 プッシュダウン・木オートマトン (PDTA) とは, $M = (Q, \Sigma, \Gamma, q_0, Z_0, R)$ のことである. ここで,

Q は状態の有限集合である.

Σ は有限階層化アルファベットで, 入力アルファベットである.

Γ は $\Gamma = \Gamma_0 \cup \Gamma_1$ をみたす有限階層化アルファベットで, プッシュダウン・アルファベットである.

$q_0 \in Q$ は初期状態である.

$Z_0 \in \Gamma_0$ は開始記号である.

R は次の形の規則の集合である.

読み込み規則 :

$$(i) \quad q(a, A) \rightarrow a$$

ここで, $a \in \Sigma_0$, $q \in Q$, $A \in \Gamma_0$.

$$(ii) \quad q(b(x_1 \dots x_n), B) \rightarrow b(q_1(x_1, \pi_1) \dots q_n(x_n, \pi_n))$$

ここで, $n \geq 1$, $b \in \Sigma_n$, $q, q_1, \dots, q_n \in Q$, $B \in \Gamma_1$, $\pi_1, \dots, \pi_n \in \Gamma_1^* \Gamma_0 \cup \Gamma_1^*$.

ε -規則 :

$$(iii) \quad q(x, A) \rightarrow q'(x, \pi)$$

ここで, $q, q' \in Q$, $A \in \Gamma_0$, $\pi \in \Gamma_1^* \Gamma_0$.

$$(iv) \quad q(x, B) \rightarrow q'(x, \pi)$$

ここで, $q, q' \in Q$, $B \in \Gamma_1$, $\pi \in \Gamma_1^* \Gamma_0 \cup \Gamma_1^*$.

定義 4.2 $M = (Q, \Sigma, \Gamma, q_0, Z_0, R)$ を PDTA とする. M の即席表現とは, 三つ組み $q(\alpha, \pi) \in Q \times T_\Sigma \times \Gamma_1^* \Gamma_0$ のことである. ID で M の即時表現全体からなる集合を表すことにする. M の様相を $T_\Sigma(ID)$ の元と定義する. 関係 $\stackrel{|_M}{\rightarrow}$ を次のように定義する. 任意の様相 $c, c' \in T_\Sigma(ID)$ に対し, 次の条件の内の一つが成り立つとき, $c \stackrel{|_M}{\rightarrow} c'$ となる.

- タイプ (i) の規則 $q(a, A) \rightarrow a \in R$ が存在して, ある $d \in D_c$ に対し, $c/d = q(a, A)$ かつ $c' = c(d \leftarrow a)$ である.
- タイプ (ii) の規則 $q(b(x_1 \dots x_n), B) \rightarrow b(q_1(x_1, \pi_1) \dots q_n(x_n, \pi_n)) \in R$ が存在して, ある $d \in D_c$, $\alpha_1, \dots, \alpha_n \in T_\Sigma$, $\rho \in \Gamma_1^* \Gamma_0$ に対し, $c/d = q(b(\alpha_1 \dots \alpha_n), B\rho)$ かつ $c' = c(d \leftarrow b(q_1(\alpha_1, \pi_1) \dots q_n(\alpha_n, \pi_n)))$ である. ただし, それぞれの $1 \leq i \leq n$ に対し, もし $\pi_i \in \Gamma_1^* \Gamma_0$ ならば $\pi'_i = \pi_i$ であり, もし $\pi_i \in \Gamma_1^*$ ならば $\pi'_i = \pi_i \rho$ である.

- タイプ (iii) の規則 $q(x, A) \rightarrow q'(x, \pi) \in R$ が存在して, ある $d \in D_c$, $\alpha \in T_\Sigma$ に対し, $c/d = q(\alpha, A)$ かつ $c' = c(d \leftarrow q'(\alpha, \pi))$ である.
- タイプ (iv) の規則 $q(x, B) \rightarrow q'(x, \pi) \in R$ が存在して, ある $d \in D_c$, $\alpha \in T_\Sigma$, $\rho \in \Gamma_1^* \Gamma_0$ に対し, $c/d = q(\alpha, B\rho)$ かつ $c' = c(d \leftarrow q'(\alpha, \pi'))$ である. ただし, もし $\pi \in \Gamma_1^* \Gamma_0$ ならば $\pi' = \pi$ であり, もし $\pi \in \Gamma_1^*$ ならば $\pi' = \pi\rho$ である.

(n ステップの) 計算とは, 様相の有限列 $c_0, \dots, c_n \in T_\Sigma(ID)$ ($n \geq 0$) で, $c_0 \stackrel{*}{\vdash}_M c_1 \stackrel{*}{\vdash}_M \dots \stackrel{*}{\vdash}_M c_n$ を満たすものである. 計算 c_1, \dots, c_n が存在するとき, $c_0 \stackrel{*}{\vdash}_M c_n$ と書く.

定義 4.3 $M = (Q, \Sigma, \Gamma, q_0, Z_0, R)$ を PDTA とする. M によって受理される木言語とは, 集合 $T(M) = \{\alpha \in T_\Sigma \mid q_0(\alpha, Z_0) \stackrel{*}{\vdash}_M \alpha\}$ のことである.

文献 [6] において, Guessarian は PDTA のバリエーションのいくつかを紹介し, それらにより受理される木言語のクラスが一致することを示している. 本論文にある PDTA の定義は, 文献 [6] の言葉を用いると, “restricted PDTA accepting by empty store” と言い表すことができる.

これより, 線形という制限のされた PDTA を考える. そして, 線形な PDTA が受理する言語のクラスと Spine Grammar により生成される言語のクラスが一致することを示す.

定義 4.4 $M = (Q, \Sigma, \Gamma, q_0, Z_0, R)$ を PDTA とする. M が線形であるとは, M が次を満たすことである.

- すべてのタイプ (ii) の規則 $q(b(x_1 \dots x_n), B) \rightarrow b(q_1(x_1, \pi_1) \dots q_n(x_n, \pi_n)) \in R$ に対し, $\{|i| 1 \leq i \leq n, \pi_i \in \Gamma_1^*\} = 1$ である. つまり, タイプ (ii) の規則が適応されるときに一つをのぞくすべてのプッシュダウン記憶は初期化されなくてはならない.
- すべてのタイプ (iv) の規則 $q(x, B) \rightarrow q'(x, \pi) \in R$ に対し, $\pi \in \Gamma_1^*$ である.

補題 4.5 任意の Spine Grammar \mathcal{G} に対し, 線形な PDTA M が存在して, $T(M) = L(\mathcal{G})$ をみたす.

(証明) 省略 □

補題 4.6 任意の線形な PDTA M に対し, Spine Grammar \mathcal{G} が存在して, $L(\mathcal{G}) = T(M)$ をみたす.

(証明) 省略 □

補題 4.5 と補題 4.6 の結果より, 線形プッシュダウン・木オートマトンについて次のような結果が得られる.

定理 3 線形プッシュダウン・木オートマトンによって受理される木言語のクラスは, Spine Grammar によって生成される木言語のクラスと一致する.

参考文献

- [1] W. S. Brainerd, Tree generating regular systems, *Information & Control*, vol.14, no.2, pp.217–231, 1969.
- [2] J. Engelfriet, Bottom-up and top-down tree transformations—A comparison, *Mathematical System Theory*, vol.9, no.3, pp.198–231, 1975.
- [3] J. Engelfriet and E. M. Schmidt, IO and OI, *J. Computer & System Science*, vol.15, no.3, pp.328–353, 1977 and vol.16, no.1, pp.67–99, 1978.
- [4] A. Fujiyoshi and T. Kasai, Multi-Phase Tree Transformations, *IEICE Trans. Fundamentals*, vol.E80-A, no.4, pp.761–768, 1997.
- [5] 藤芳明生, 笠井琢美, Tree Adjoining Grammar の文脈自由木文法による特徴づけ, 1998年夏のLA シンポジウム, 情報基礎理論ワークショップ, pp.34–39, 1998.
- [6] I. Guessarian, Pushdown tree automata, *Mathematical System Theory*, vol.16, no.4, pp.237–263, 1983.
- [7] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison Wesley, Reading, MA, 1979
- [8] A. K. Joshi, L.S. Levy and M. Takahashi, Tree adjunct Grammars, *J. Computer & System Science*, vol.10, no.1, pp.136–163, 1975.
- [9] S. Rajasekaran, Tree-adjoining language parsing in $O(n^6)$ time, *SIAM J. Comput.*, vol.25, no.4, pp.862–873, 1996.
- [10] S. Rajasekaran and S. Yooseph, TAL Recognition in $O(M(n^2))$ time, *J. Computer & System Science*, vol.56, no.1, pp.83–89, 1998.
- [11] W. C. Rounds, Mapping and grammars on trees, *Mathematical System Theory*, vol.4, no.3, pp.257–287, 1970.
- [12] J. W. Thatcher, Tree automata: an informal survey, in *Currents in the theory of computing*, ed. A. V. Aho, pp.143–172, Prentice-Hall, Englewood Cliffs, 1973.
- [13] K. Vijay-Shanker and A. K. Joshi, Some computational properties of tree adjoining grammars, *Proc. 23rd Meeting of the Association for Computational Linguistics*, pp.82–93, 1985.
- [14] K. Vijay-Shanker and D. J. Weir, The equivalence of four extensions of context-free grammars, *Mathematical System Theory*, vol.27, no.6, pp.511–546, 1994.