Subclasses of Tree Adjoining Grammar for RNA Secondary Structure

Yuki Kato
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara 630-0192, Japan
yuuki-ka@is.naist.jp

Hiroyuki Seki
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara 630-0192, Japan
seki@is.naist.jp

Tadao Kasami
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara 630-0192, Japan
kasami@empirical.jp

Abstract

Several grammars have been proposed for representing RNA secondary structure including pseudoknots. In this paper, we introduce subclasses of multiple context-free grammars which are weakly equivalent to these grammars for RNA, and clarify the generative power of these grammars as well as closure property.

1 Introduction

Much attention has been paid to RNA secondary structure prediction techniques based on context-free grammar (cfg) since cfg can represent stem-loop structure (Figure 1 (a)) by its derivation tree and recognition (or *secondary structure prediction* in biological words) can be performed in $O(n^3)$ time where n is the length of an input sequence (primary structure). Especially, techniques based on CKY (Cocke-Kasami-Younger) algorithm have been widely investigated (Durbin et al., 1998). *Pseudoknot* (Figure 1 (b)) is one of the typical substructures found in an RNA secondary structure. An alternative representation of a pseudoknot is arc depiction in which arcs cross (see Figure 2). It has been recognized that pseudoknots play an important role in RNA functions such as ribosomal frameshifting and splicing. However, it is known that cfg cannot represent pseudoknot structure.

In bioinformatics, a few grammars have been proposed to represent pseudoknots (Uemura et al., 1999; Rivas and Eddy, 2000) (also see (Condon, 2003)). In the pioneering paper, Uemura et al. (1999) define two subclasses of tree adjoining grammar (tag) called *sl-tag* and *esl-tag*, and argue that esl-tag is appropriate for representing RNA secondary structure including pseudoknots. Rivas and Eddy (2000) provide keen observation on representation of RNA secondary structure by a sequence with a single "hole" and introduce a new class of grammars for deriving sequences with hole. These grammars have generative power stronger than cfg while recognition can be performed in polynomial time. However, relation among the generative power of these grammars and/or mildly csg has not been clarified.

In this paper, we identify grammars for RNA secondary structure (Uemura et al., 1999; Rivas and Eddy, 2000) as subclasses of multiple context-free grammar (mcfg) (Kasami et al., 1988a; Seki et al., 1991) and clarify inclusion relation among the classes of languages generated by these grammars.

The rest of this paper is organized as follows. Section 2 reviews the grammars mentioned above. In section 3, these grammars are characterized as subclasses of mcfg. Generative power and closure property of these grammars are discussed in section 4. Section 5 concludes the paper.

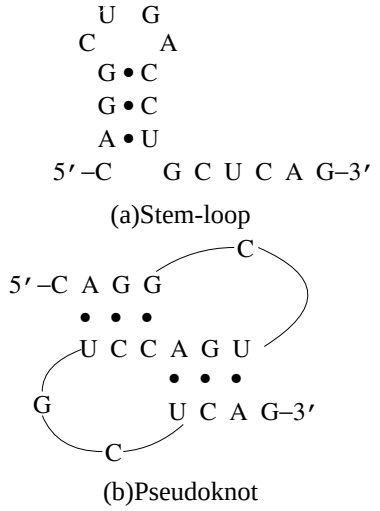


Figure 1: Example of RNA secondary structure

TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms.
May 20-22, 2004, Vancouver, BC, CA.
Pages 48-55.

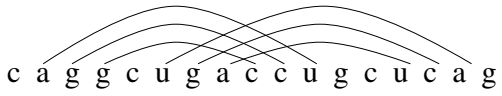


Figure 2: Arc depiction of Figure 1 (b)

2 Preliminaries

2.1 Tree Adjoining Grammar

We will use standard notations for tree adjoining grammar (Joshi and Schabes, 1997). The empty sequence is denoted by ε. For a sequence $\alpha \in S^*$, let $|\alpha|$ denote the length of α.

A *tree adjoining grammar* (*tag*) is a 5-tuple $G = (N, T, S, \mathcal{I}, \mathcal{A})$ where N and T are finite sets of nonterminals and terminals respectively, S the start symbol, $\mathcal{I}$ a finite set of *initial trees* (*center trees*) and $\mathcal{A}$ a finite set of *adjunct trees* (*auxiliary trees*). The path of an adjunct tree from the root node to the foot node is called the *backbone*. *Selective adjoining* (SA), *null adjoining* (NA) and *obligatory adjoining* (OA) are defined in the standard way. For trees s and t, if t' is obtained by adjoining s into t, we write $t \vdash_s t'$ (or simply $t \vdash t'$). We write the reflective and transitive closure of $\vdash$ as $\vdash^*$. We call t' a *derived tree* (or a tree derived from t) if $t \vdash^* t'$ for some $t \in \mathcal{I} \cup \mathcal{A}$. A node n is *inactive* if the constraint for the node is NA, otherwise *active*. If no active node in a tree t has OA constraint, then t is called *mature*. The tree set of a tag G is defined as $T(G) = \{t \mid s \vdash^* t,\ s \in \mathcal{I}$ and t is mature$\}$. $T(G)$ can be alternatively characterized in a bottom up way as follows. Let us define a series of tree sets $T_0(G),\ T_1(G), \ldots$.

(T1) $T_0(G) = \{t \in \mathcal{I} \cup \mathcal{A} \mid t$ is mature$\}$.

(T2) $T_{n+1}(G) = T_n(G) \cup \{t \mid t_0 \vdash_{s_1} t_1 \vdash_{s_2} \cdots \vdash_{s_k} t_k = t,\ t_0 \in \mathcal{I} \cup \mathcal{A},\ s_i \in T_n(G)\ (1 \le i \le k),\ p_1, \ldots, p_k$ are different addresses of t_0, s_i is adjoinable to t_0 at $p_i\ (1 \le i \le k)$ and t is mature$\}$.

It is not difficult to show that $T(G) = \{t \mid t \in T_n(G)$ for some $n \ge 0$ and yield$(t) \in T^*\}$. This characterization of $T(G)$ by (T1) and (T2) is frequently used in proofs in section 3.

The language generated by G is defined as $L(G) = \{w \mid w =$ yield$(t),\ t \in T(G)\}$, which is called a *tree adjoining language* (*tal*). Let TAG denote the class of tags and TAL denote the class of tals. We use the same notational convention, i.e., a language generated by an xxg is called an xxl, the class of xxgs is denoted by XXG and the class of xxls is denoted by XXL.

We now define *simple linear tag* (*sl-tag*) and *extended simple linear tag* (*esl-tag*) introduced in (Uemura et al., 1999). Let $G = (N, T, S, \mathcal{I}, \mathcal{A})$ be a tag. An elementary tree is *simple linear* if it has exactly one active node, and for an adjunct tree, the active node is on the backbone of the tree. A tag G is a *simple linear tag* (*sl-tag*) if and only if all elementary trees in G are simple linear. An adjunct tree is *semi-simple linear* if it has two active nodes, where one is on the backbone and the other is elsewhere. A tag G is an *extended simple linear tag* (*esl-tag*) if and only if all initial trees in G are simple linear and all adjunct trees in G are either simple linear or semi-simple linear.

Example 1 (Uemura et al., 1999). Let $G = (N, T, S, \mathcal{I}, \mathcal{A})$ be an sl-tag where $N = \{S\}$, $T = \{a, c, g, u\}$ and elementary trees in $\mathcal{I}$ and $\mathcal{A}$ are shown in Figure 3. In the figure, $z \in \{a, c, g, u\}$, $(x, y) \in \{(a, u), (u, a), (c, g), (g, c)\}$ and an active node is denoted by S^*. Figure 4 shows a derivation of a pseudoknot. □

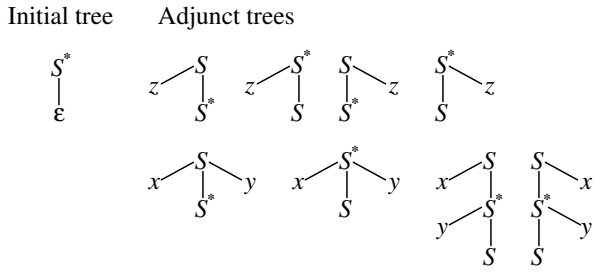


Figure 3: Elementary trees in Example 1

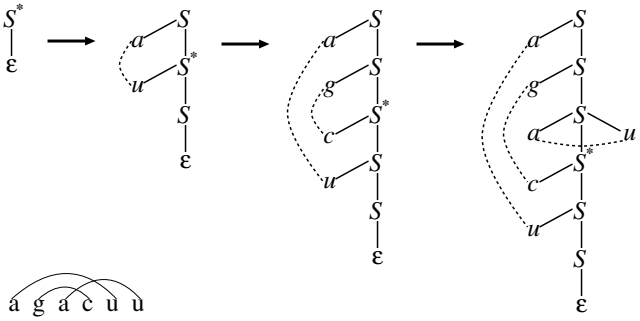


Figure 4: A derivation of a pseudoknot in Example 1

By definition,

$$\text{SL-TAL} \subseteq \text{ESL-TAL} \subseteq \text{TAL}. \qquad (*1)$$

On the inclusion relation among CFL, SL-TAL and ESL-TAL, the following has been shown in Propositions 1 to 3 of (Uemura et al., 1999):

$$L_2 = \{\sharp a_1^k b_1^k \sharp a_2^l b_2^l \sharp a_3^m b_3^m \sharp a_4^n b_4^n \sharp \mid k, l, m, n \ge 1\} \in \text{CFL} \setminus \text{SL-TAL}, \qquad (*2)$$

$$\{a^n b^n c^n \mid n \ge 0\} \in \text{SL-TAL} \setminus \text{CFL}, \qquad (*3)$$

$$\text{CFL} \subseteq \text{ESL-TAL}. \qquad (*4)$$

2.2 Multiple Context-Free Grammar

A *multiple context-free grammar* (*mcfg*) or *linear context-free rewriting system* (Vijay-Shanker et al., 1987) is a 5-tuple $G = (N, T, F, P, S)$ where N is a finite set of nonterminals, T a finite set of terminals, F a finite set of functions, P a finite set of (production) rules and S the start symbol. For each $A \in N$, a positive integer denoted as $\dim(A)$ is given and A derives $\dim(A)$-tuples of terminal sequences. For the start symbol S, $\dim(S) = 1$. For each $f \in F$, positive integers d_i $(0 \leq i \leq k)$ are given and f is a total function from $(T^*)^{d_1} \times \cdots \times (T^*)^{d_k}$ to $(T^*)^{d_0}$ which satisfies the following condition (F):

(F) Let $\overline{x_i} = (x_{i1}, \ldots, x_{id_i})$ denote the ith argument of f for $1 \leq i \leq k$. The hth component of function value for $1 \leq h \leq d_0$, denoted by $f^{[h]}$, is defined as

$$f^{[h]}[\overline{x_1}, \ldots, \overline{x_k}] = \beta_{h0} z_{h1} \beta_{h1} z_{h2} \cdots z_{hv_h} \beta_{hv_h} \quad (*)$$

where $\beta_{hl} \in T^*$ $(0 \leq l \leq v_h)$ and $z_{hl} \in \{x_{ij} \mid 1 \leq i \leq k,\ 1 \leq j \leq d_i\}$ $(1 \leq l \leq v_h)$. The total number of occurrences of x_{ij} in the right hand sides of $(*)$ from $h = 1$ through d_0 is at most one.

Each rule in P has the form of $A_0 \to f[A_1, \ldots, A_k]$ where $A_i \in N$ $(0 \leq i \leq k)$ and $f : (T^*)^{\dim(A_1)} \times \cdots \times (T^*)^{\dim(A_k)} \to (T^*)^{\dim(A_0)} \in F$. If $k \geq 1$, then the rule is called a *nonterminating rule*, and if $k = 0$, then it is called a *terminating rule*.

We define the relation $\stackrel{*}{\Rightarrow}$ and derivation trees (refer to Figure 5) recursively by the following (L1) and (L2):

(L1) If $A \to \alpha \in P$ $(\alpha \in T^*)$, then $A \stackrel{*}{\Rightarrow} \alpha$ and a tree with the single node labeled $A : \alpha$ is a derivation tree for α.

(L2) If $A \to f[A_1, \ldots, A_k] \in P$, $A_i \stackrel{*}{\Rightarrow} \alpha_i = (\alpha_{i1}, \ldots, \alpha_{i\dim(A_i)})$ $(1 \leq i \leq k)$ and $t_1, \ldots, t_k$ are derivation trees for $\alpha_1, \ldots, \alpha_k$, then $A \stackrel{*}{\Rightarrow} f[\alpha_1, \ldots, \alpha_k]$ where $f[\alpha_1, \ldots, \alpha_k]$ denotes the $\dim(A)$-tuple of terminal sequences obtained from the right hand sides of $(*)$ in condition (F) by substituting α_{ij} $(1 \leq i \leq k,\ 1 \leq j \leq \dim(A_i))$ into x_{ij}, and a tree with the root labeled $A : f$ which has $t_1, \ldots, t_k$ as (immediate) subtrees from left to right is a derivation tree for $f[\alpha_1, \ldots, \alpha_k]$.

The language generated by an mcfg G is defined as $L(G) = \{w \in T^* \mid S \stackrel{*}{\Rightarrow} w\}$.

To introduce subclasses of MCFG, we define a few terminologies. Let $G = (N, T, F, P, S)$ be an arbitrary mcfg. The *dimension* of G is defined as $\dim(G) = \max\{\dim(A) \mid A \in N\}$. For a function $f \in F$, let $\mathrm{rank}(f)$ denote the number of arguments of f. The *rank* of G is defined as $\mathrm{rank}(G) = \max\{\mathrm{rank}(f) \mid f \in F\}$. For a function $f : (T^*)^{d_1} \times \cdots \times (T^*)^{d_k} \to (T^*)^{d_0}$, let $\deg(f) = \Sigma_{j=0}^{k} d_j$, which is called the *degree* of f. Finally, let us define the degree of G as $\deg(G) = \max\{\deg(f) \mid f \in F\}$. By definition, $\deg(G) \leq \dim(G)(\mathrm{rank}(G) + 1)$. With these parameters, we define subclasses of MCFG. An mcfg G with $\dim(G) \leq m$ and $\mathrm{rank}(G) \leq r$ is called an (m, r)-mcfg. Likewise, an mcfg G with $\dim(G) \leq m$ is called an m-mcfg.

It has been proved that

$$\text{TAL} \subset \text{(2,2)-MCFL} \subset \text{2-MCFL} \subset \text{MCFL}, \quad (*5)$$

where the proper inclusion relation from left to right in $(*5)$ were given by Lemma 4.15 of (Seki et al., 1991), Theorem 1 of (Rambow and Satta, 1994) and Lemma 5 of (Kasami et al., 1988a), respectively.

Example 2. Consider the (2,2)-mcfg $G_3 = (\{S, A\}, \{a, c, g, u\}, F_3, P_3, S)$ for generating RNA sequences, where P_3 and F_3 are as follows:

$$\begin{aligned}
&S \to J[A],\\
&A \to XS_1[A, A] \mid XS_2[A, A] \mid XS_3[A, A],\\
&A \to BF_1[A, A] \mid BF_2[A, A] \mid BF_3[A, A],\\
&A \to BP_{\alpha\beta}[A]\\
&\quad ((\alpha, \beta) \in \{(a, u), (u, a), (c, g), (g, c)\}),\\
&A \to UP_{\alpha}^{1,L}[A] \mid UP_{\alpha}^{1,R}[A] \mid UP_{\alpha}^{2,L}[A] \mid UP_{\alpha}^{2,R}[A]\\
&\quad (\alpha \in \{a, c, g, u\}),\\
&A \to (\varepsilon, \varepsilon),\\
&J[(x_1, x_2)] = x_1 x_2,\\
&XS_1[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11}, x_{21} x_{12} x_{22}),\\
&XS_2[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11} x_{21}, x_{12} x_{22}),\\
&XS_3[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11} x_{21} x_{12}, x_{22}),\\
&BF_1[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11}, x_{12} x_{21} x_{22}),\\
&BF_2[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11} x_{12}, x_{21} x_{22}),\\
&BF_3[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11} x_{12} x_{21}, x_{22}),\\
&BP_{\alpha\beta}[(x_1, x_2)] = (\alpha x_1, x_2 \beta),\\
&UP_{\alpha}^{1,L}[(x_1, x_2)] = (\alpha x_1, x_2),\\
&UP_{\alpha}^{1,R}[(x_1, x_2)] = (x_1 \alpha, x_2),\\
&UP_{\alpha}^{2,L}[(x_1, x_2)] = (x_1, \alpha x_2),\\
&UP_{\alpha}^{2,R}[(x_1, x_2)] = (x_1, x_2 \alpha).
\end{aligned}$$

Functions have mnemonic names where XS, BF, BP and UP stand for crossing, bifurcation, base pair and unpair, respectively. The RNA sequence `agacuu` in Figure 4 can be generated by the above rules as follows: $A \stackrel{*}{\Rightarrow} BP_{gc}[(\varepsilon, \varepsilon)] = (g, c)$, $A \stackrel{*}{\Rightarrow} BP_{au}[(g, c)] = (ag, cu)$, $A \stackrel{*}{\Rightarrow} BP_{au}[(\varepsilon, \varepsilon)] = (a, u)$, $A \stackrel{*}{\Rightarrow} XS_2[(ag, cu), (a, u)] = (aga, cuu)$ and $S \stackrel{*}{\Rightarrow}$

$J[(aga, cuu)] = agacuu$. G_3 has a derivation tree (Figure 5) for agacuu which represents the pseudoknot shown in Figure 4. □

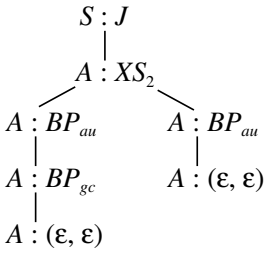


Figure 5: A derivation tree in G_3

Recognition problem for mcfg can be solved in polynomial time:

Proposition 1 (Kasami et al., 1988b; Seki et al., 1991). Let G be an mcfg with $\deg(G) = e$. For a given $w \in T^*$, whether $w \in L(G)$ or not can be decided in $O(n^e)$ time where $n = |w|$. □

3 Subclasses of MCFG

3.1 A Subclass of MCFG for SL-TAL

Grammars G and G' are called weakly equivalent if $L(G) = L(G')$. Remember that each elementary tree in an sl-tag contains exactly one active node as shown in Figure 6 (An inactive node and an active node are denoted like A^ϕ and B^*, respectively in the figure). By utilizing this restriction, we can define a translation from an sl-tag into a weakly equivalent (2,2)-mcfg simpler than that of (Vijay-Shanker et al., 1986). Namely, for an adjunct tree in Figure 6 (a), construct an mcfg rule $A \to f[B]$ where $f[(x_1, x_2)] = (u_1x_1v_1, v_2x_2u_2)$. This translation motivates us to define the following subclass of (2,1)-MCFG.

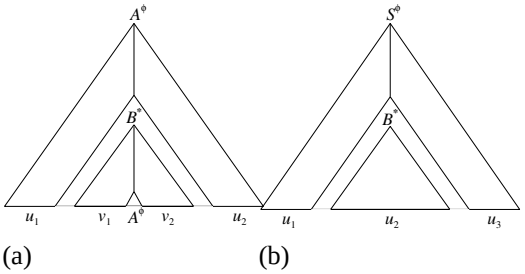


Figure 6: Elementary trees in sl-tag

Definition 1. A (2,1)-mcfg $G = (N, T, F, P, S_0)$ is an *sl-mcfg* if G satisfies the following conditions (1) and (2):

(1) For each nonterminal A other than S_0, $\dim(A) = 2$.

(2) Each nonterminating rule has the form of either $S_0 \to J[A]$ where $J[(x_1, x_2)] = x_1x_2$ or $A \to f[B]$ where $A, B \in N \setminus \{S_0\}$ and $f[(x_1, x_2)] = (u_1x_1v_1, v_2x_2u_2)$ for some $u_j, v_j \in T^*$ $(j = 1, 2)$. Such a function f is called a *simple linear function*. □

Lemma 2. SL-TAL = SL-MCFL.

Proof. (SL-TAL $\subseteq$ SL-MCFL) Let $G = (N, T, S, \mathcal{I}, \mathcal{A})$ be a given sl-tag. We will construct an sl-mcfg $G' = (N', T, F, P, S_0)$ as follows:

(1) $N' = N \cup \{S_0\}$ where $\dim(S_0) = 1$ and $\dim(A) = 2$ for each $A \in N$.

(2) P (and F) are the smallest sets which satisfy the following conditions (a) through (c):

(a) $S_0 \to J[S] \in P$ and $J \in F$.

(b) For each adjunct tree $t \in \mathcal{A}$ shown in Figure 6 (a),
- $A \to f[B] \in P$ and $f \in F$ where $f[(x_1, x_2)] = (u_1x_1v_1, v_2x_2u_2)$, and
- $A \to (u_1v_1, v_2u_2)$ if B in Figure 6 (a) does not have OA constraint (i.e., t is mature).

(c) For each initial tree $t \in \mathcal{I}$ shown in Figure 6 (b),
- $S \to g[B] \in P$ and $g \in F$ where $g[(x_1, x_2)] = (u_1x_1u_2, x_2u_3)$, and
- $S \to (u_1u_2, u_3)$ if t is mature.

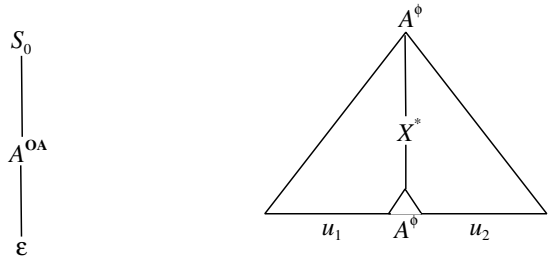
We can show that there exists a tree $t \in T_n(G)$ for some $n \geq 0$ such that yield$(t) = w_1Aw_2$ $(A \in N,\ w_1, w_2 \in T^*)$ if and only if $A \overset{*}{\Rightarrow}_{G'} (w_1, w_2)$.
(SL-MCFL $\subseteq$ SL-TAL) Let $G = (N, T, F, P, S_0)$ be a given sl-mcfg. Construct an sl-tag $G' = (N', T, S_0, \mathcal{I}, \mathcal{A})$ as follows:

(1) $N' = N \cup \{X\}$ where $X \notin N$.

(2) $\mathcal{I}$ consists of initial trees shown in Figure 7 (a) for $S_0 \to J[A] \in P$.

(3) $\mathcal{A}$ is the smallest set satisfying:
- For each $A \to f[B] \in P$ where $f[(x_1, x_2)] = (u_1x_1v_1, v_2x_2u_2)$, the adjunct tree shown in Figure 6 (a) belongs to $\mathcal{A}$.
- For each $A \to (u_1, u_2) \in P$, the adjunct tree in Figure 7 (b) belongs to $\mathcal{A}$.

Proof of $L(G) = L(G')$ can be done in a similar way to the converse direction. □



(a) (b)

Figure 7: Constructed elementary trees

3.2 A Subclass of MCFG for ESL-TAL

In this subsection, we will define a subclass of (2,2)-MCFG which exactly generates ESL-TAL. Let $G = (N, T, S, \mathcal{I}, \mathcal{A})$ be a given esl-tag. By virtue of Property 2 of (Uemura et al., 1999), we can assume that G is in normal form such that for every semi-simple linear adjunct tree $t \in \mathcal{A}$, yield$(t) \in N$. Thus, for each leaf v of t, either v is the foot node or the label of v is ε (see Figure 8). From this observation, we define a subclass of (2,2)-MCFG by adding rules corresponding to adjunct trees shown in Figure 8 to the definition of sl-mcfg.

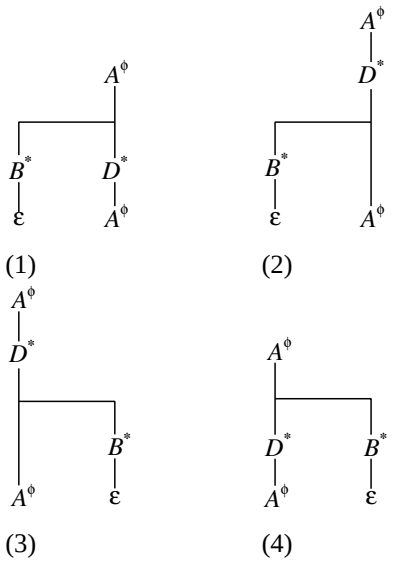


Figure 8: Semi-simple linear adjunct trees in normal form

Definition 2. A (2,2)-mcfg $G = (N, T, F, P, S_0)$ is an *esl-mcfg* if each nonterminating rule has one of the following forms (1) through (3):

(1) $A \to J[B]$ where $\dim(A) = 1$ and $\dim(B) = 2$.

(2) $A \to f[B]$ where f is a simple linear function.

(3) $A \to g[B, D]$ where $\dim(A) = \dim(D) = 2$, $\dim(B) = 1$, $g \in \{C_1, C_2, C_3, C_4\}$ and

$$C_1[x_1, (x_{21}, x_{22})] = (x_1 x_{21}, x_{22}),$$
$$C_2[x_1, (x_{21}, x_{22})] = (x_{21} x_1, x_{22}),$$
$$C_3[x_1, (x_{21}, x_{22})] = (x_{21}, x_1 x_{22}),$$
$$C_4[x_1, (x_{21}, x_{22})] = (x_{21}, x_{22} x_1).$$

□

Lemma 3. ESL-TAL = ESL-MCFL.

Proof. (ESL-TAL $\subseteq$ ESL-MCFL) Let $G = (N, T, S, \mathcal{I}, \mathcal{A})$ be a given esl-tag in normal form (Uemura et al., 1999). We construct an esl-mcfg $G' = (N', T, F, P, S_0)$ from G as follows:

(1) $N' = N \cup \{A' \mid A \in N\}$ where $\dim(A') = 1$ and $\dim(A) = 2$ for $A \in N$.

(2) P (and F) are the smallest sets which satisfy the following conditions (a) through (d):

(a) For each $A \in N$, $A' \to J[A] \in P$ and $J \in F$.

(b) Same as (2) (b) (c) in the proof of (SL-TAL $\subseteq$ SL-MCFL) in Lemma 2.

(c) For each semi-simple linear adjunct tree t shown in Figure 8 (1),
- $A \to C_1[B', D] \in P$ and $C_1 \in F$, and
- $A \to (\varepsilon, \varepsilon) \in P$ if t is mature.

(d) For each semi-simple linear adjunct tree (2) through (4) in Figure 8, the rules using C_2, C_3 and C_4, respectively, instead of C_1 belong to P.

We can show that there exists a tree $t \in T_n(G)$ for some $n \geq 0$ such that yield$(t) = w_1 A w_2$ ($A \in N$, $w_1, w_2 \in T^*$) if and only if $A \overset{*}{\Rightarrow}_{G'} (w_1, w_2)$.
Proof of (ESL-MCFL $\subseteq$ ESL-TAL) is similar and is omitted here. □

3.3 A Subclass of MCFG for RPL

Rivas and Eddy (2000) introduce *crossed-interaction grammar* (*cig*) which is similar to mcfg, and define *RNA pseudoknot grammar* (*rpg*) as a subclass of CIG to describe RNA secondary structure including pseudoknots. In this subsection, we reformulate RPG as a subclass of MCFG.

Definition 3. A (2,2)-mcfg $G = (N, T, F, P, S)$ is called an rpg if a nonterminating rule is one of the following forms (1) through (3):

(1) $A \to J[B]$.

(2) $A \to BF[E_1, E_2]$ where $\dim(A) = 2$, $\dim(E_1) = \dim(E_2) = 1$ and $BF[x_1, x_2] = (x_1, x_2)$.

(3) $A \to f[B,D]$ where $\dim(A) = \dim(B) = \dim(D) = 2$, $f \in \{XS_1, XS_2, XS_3, W\}$, XS_i $(i = 1,2,3)$ is defined in Example 2 and $W[(x_{11},x_{12}),(x_{21},x_{22})] = (x_{11}x_{21}, x_{22}x_{12})$. □

Proposition 4.

$$\text{RPL} \subseteq \text{(2,2)-MCFL}. \qquad (*6)$$

□

We obtain the following property on recognition complexity.

Proposition 5. For a given $w \in T^*$ $(n = |w|)$, whether $w \in L$ or not can be decided in $O(n^6)$ time if L is an rpl, $O(n^5)$ time if L is an esl-tal, and $O(n^4)$ time if L is an sl-tal.

Proof. For an rpg G, $\deg(G) \leq 6$, for an esl-mcfg G, $\deg(G) \leq 5$ and for an sl-mcfg G, $\deg(G) \leq 4$. The proposition follows from Proposition 1, Lemmas 2 and 3. □

The above complexity results were first shown in (Uemura et al., 1999) for ESL-TAL and SL-TAL and in (Rivas and Eddy, 2000) for RPL by providing an individual recognition algorithm for each class. On the other hand, by identifying these classes of languages as subclasses of MCFL, we can easily obtain the same results as stated in Proposition 5. Akutsu (2000) defines a structure called a simple pseudoknot and proposes an $O(n^4)$ time exact prediction algorithm and $O(n^{4-\delta})$ time approximation algorithm without using grammar. Note that the set of simple pseudoknots can be generated by an sl-tag.

4 Inclusion Relation

First, we summarize the inclusion relation among the classes of languages stated in $(*1)$ through $(*6)$.

Proposition 6. (1) $(\text{CFL} \cup \text{SL-TAL}) \subseteq \text{ESL-TAL} \subseteq \text{TAL} \subset \text{(2,2)-MCFL}$.

(2) $\text{RPL} \subseteq \text{(2,2)-MCFL} \subset \text{2-MCFL} \subset \text{MCFL}$. □

In the following, we refine the above proposition.

4.1 $(\text{CFL} \cup \text{SL-TAL}) \subset \text{ESL-TAL}$

First, we introduce a normal form of esl-mcfg and then show closure properties of SL-TAL and ESL-TAL. By using sl-mcfg and esl-mcfg, we can prove these properties in a simple way. Some of these properties will be used for proving inclusion relation between SL-TAL and ESL-TAL.

Definition 4. An esl-mcfg is in normal form if the following conditions (1) and (2) hold:

(1) For each $A \to f[B]$ where $f[(x_1,x_2)] = (u_1x_1v_1, v_2x_2u_2)$ is a linear function, $|u_1v_1v_2u_2| = 1$.

(2) For each $A \to (u_1,u_2)$ $(u_1,u_2 \in T^*)$, $u_1 = u_2 = \varepsilon$. □

Remark that a similar normal form is defined for esl-tag in (Uemura et al., 1999). It is easy to prove the following lemma.

Lemma 7. For a given esl-mcfg G, a normal form esl-mcfg G' can be constructed from G such that $L(G') = L(G)$. □

Theorem 8. SL-TAL and ESL-TAL have the following properties.

(1) SL-TAL contains every linear language.

(2) SL-TAL is closed under union, homomorphism, intersection with regular languages and regular substitution, but is not closed under concatenation, Kleene closure, positive closure or substitution.

(3) ESL-TAL is closed under intersection with regular languages and substitution.

Proof. (1) For linear cfg rules $A \to u_1Bv_1$ and $A \to u$, construct sl-mcfg rules $A \to f[B]$ where $f[(x_1,x_2)] = (u_1x_1v_1, x_2)$ and $A \to (u,\varepsilon)$, respectively.

(2) (regular substitution) Let $G = (N,T,F,P,S_0)$ be an sl-mcfg in normal form. We also assume that each rule $A \to f[B] \in P$ has a unique label, say r, and write $r : A \to f[B] \in P$. Let $s : T \to 2^{(T')^*}$ be a regular substitution and for each $\alpha \in T$, let $s(\alpha) = L(G_\alpha)$ where $G_\alpha = (N_\alpha, T', P_\alpha, S_\alpha)$ is a regular grammar. We now construct an sl-mcfg $G' = (N',T',F',P',S_0)$ such that $L(G') = s(L(G))$ as follows. G' will simulate G_α by a linear function instead of generating $\alpha \in T$. To do this, we introduce a nonterminal $X^{[r]}$ in G' where $X \in N_\alpha$ and $r : A \to f[B] \in P$ such that the definition of f contains $\alpha \in T$.

- $N' = N \cup \{X^{[r]} \mid X \in N_\alpha \setminus \{S_\alpha\},\ \alpha \in T,\ r : A \to f[B] \in P\}$.
- F' consists of J, $UP_\beta^{1,L}$, $UP_\beta^{1,R}$, $UP_\beta^{2,L}$, $UP_\beta^{2,R}$ $(\beta \in T')$ of Example 2 and $EPS[\,] = (\varepsilon,\varepsilon)$.
- P' is the smallest set satisfying:
  - If $S_0 \to J[A] \in P$, then $S_0 \to J[A] \in P'$.
  - Assume that $r : A \to f[B] \in P$ where $f[(x_1,x_2)] = (\alpha x_1, x_2)$ $(\alpha \in T)$. If $X \to \beta Y \in P_\alpha$ $(X,Y \in N_\alpha,\ \beta \in T')$,

then $X^{[r]} \to UP_\beta^{1,L}[Y^{[r]}] \in P'$, and if $X \to \beta \in P_\alpha$ ($X \in N_\alpha$, $\beta \in T'$), then $X^{[r]} \to UP_\beta^{1,L}[B] \in P'$ where $S_\alpha^{[r]}$ is identified with A for simplicity.

– For the other rules in P, similar construction can be defined. For example, if $f[(x_1, x_2)] = (x_1, x_2\alpha)$ ($\alpha \in T$), then we will use $UP_\beta^{2,R}$ instead of $UP_\beta^{1,L}$.

Proof of $L(G') = s(L(G))$ is easy.
The other closure properties can be easily proved.
(concatenation) Let $L = \{\sharp a_1^k b_1^k \sharp a_2^l b_2^l \mid k, l \geq 1\}$ and $L' = \{\sharp a_3^m b_3^m \sharp a_4^n b_4^n \sharp \mid m, n \geq 1\}$, both of which are sl-tals. An sl-mcfg which generates L is such that $S_0 \to J[S]$, $S \to \text{add}^{\sharp\sharp}[A]$ where $\text{add}^{\sharp\sharp}[(x_1, x_2)] = (\sharp x_1, \sharp x_2)$, $A \to f[A] \mid B$ where $f[(x_1, x_2)] = (a_1 x_1 b_1, x_2)$ and $B \to g[B] \mid (a_1 b_1, a_2 b_2)$ where $g[(x_1, x_2)] = (x_1, a_2 x_2 b_2)$. Construction of an sl-mcfg which generates L' is similar. The concatenation of them, i.e., $LL' = L_2$ defined in ($*2$) is not an sl-tal.
(Kleene closure, positive closure) By the next corollary, SL-TAL is a union closed full trio. If SL-TAL is closed under Kleene closure or positive closure, then by Theorem 3.1 of (Mateescu and Salomaa, 1997), SL-TAL is closed under concatenation, which is a contradiction.
(substitution) Let $L_1 = \{\sharp d_1 \sharp d_2 \sharp d_3 \sharp d_4 \sharp\}$, which is a finite language and thus an sl-tal, and let s be a substitution such that $s(d_i) = \{a_i^n b_i^n \mid n \geq 1\}$ ($1 \leq i \leq 4$), which is also an sl-tal by (1) of this theorem. Then $s(L_1) = L_2$ defined in ($*2$), which is not an sl-tal.

(3) (intersection with regular languages) Same as the proof of Theorem 3.9 (3) of (Seki et al., 1991).
(substitution) Easy. □

Corollary 9. SL-TAL is a full trio (or cone). (That is, SL-TAL is closed under homomorphism, inverse homomorphism and intersection with regular languages.) ESL-TAL is a substitution closed full abstract family of languages (full AFL). (That is, ESL-TAL is a full trio and closed under union, concatenation, Kleene closure and substitution.)

Proof. (full trio) By Theorem 3.2 of (Mateescu and Salomaa, 1997) and (2) of Theorem 8. (full AFL) By Theorem 3.3 of (Mateescu and Salomaa, 1997) and (1), (3) of Theorem 8. □

Now we show inclusion relation between SL-TAL and ESL-TAL.

Theorem 10. Let $L_3 = \{\sharp a_1^k b_1^k c_1^k \sharp a_2^l b_2^l c_2^l \sharp a_3^m b_3^m c_3^m \sharp a_4^n b_4^n c_4^n \sharp \mid k, l, m, n \geq 1\}$. Then, $L_3 \in$ ESL-TAL $\setminus$ (CFL $\cup$ SL-TAL).

Proof. Let h_1 be a homomorphism such that $h_1(a_1) = a_1$, $h_1(b_1) = b_1$, $h_1(c_1) = c_1$ and $h_1(x) = \varepsilon$ for $x \in \{a_i, b_i, c_i \mid i = 2, 3, 4\} \cup \{\sharp\}$. Then $h_1(L_3) = \{a_1^k b_1^k c_1^k \mid k \geq 1\}$, which is not a cfl. Since CFL is closed under homomorphism, L_3 is not a cfl. Similarly, let h_2 be a homomorphism such that $h_2(c_i) = \varepsilon$ for $i = 1, 2, 3$ and identity on the other symbols. Then $h_2(L_3) = L_2$ defined in ($*2$), which is not an sl-tal. By Theorem 8 (2), L_3 is not an sl-tal. We can easily give an esl-mcfg which generates L_3. □

4.2 RPL = (2,2)-MCFL

We introduce a condition (S) which states that for each argument (x_{i1}, x_{i2}) of a function of an mcfg, the order of the occurrences of its components x_{i1} and x_{i2} is not interchanged in the function value.

(S) Let $G = (N, T, F, P, S)$ be a 2-mcfg and f be an arbitrary function in F such that

$$f[(x_{11}, x_{12}), \ldots, (x_{n1}, x_{n2})] = (\alpha_1, \alpha_2).$$

For each i ($1 \leq i \leq n$), if both of x_{i1} and x_{i2} occur in $\alpha_1\alpha_2$, then x_{i1} occurs to the left of the occurrence of x_{i2}, i.e., $\alpha_1\alpha_2 = \beta_1 x_{i1} \beta_2 x_{i2} \beta_3$ for some $\beta_j \in (N \cup T)^*$ ($1 \leq j \leq 3$).

Lemma 11. For a given 2-mcfg G, we can construct a 2-mcfg G' satisfying condition (S) and $L(G') = L(G)$. □

Lemma 12. Let $G = (N, T, F, P, S)$ be a (2,2)-mcfg satisfying condition (S). Then we can construct an rpg G' such that $L(G') = L(G)$.

Proof. Let $G = (N, T, F, P, S)$ be an arbitrary (2,2)-mcfg satisfying condition (S). We construct an rpg G' weakly equivalent to G as follows. The number of functions $f : (T^*)^2 \times (T^*)^2 \to (T^*)^2$ satisfying condition (S) is 18. A half of them can be obtained from the other half of them by interchanging the first and second arguments. Among the remaining nine functions, four are rpg functions. The others are $f_1 = (x_{11}, x_{12}x_{21}x_{22})$, $f_2 = (x_{11}x_{12}, x_{21}x_{22})$, $f_3 = (x_{11}x_{12}x_{21}, x_{22})$, $f_4 = (x_{11}, x_{21}x_{22}x_{12})$, $f_5 = (x_{11}x_{21}x_{22}, x_{12})$. (We omit variables in the left hand sides.) For example, $A \to f_1[B, D]$ can be simulated by $A \to XS_2[B, Y_1]$, $Y_1 \to BF[Y_2, Y_3]$, $Y_2 \to \varepsilon$ and $Y_3 \to J[D]$. The other four functions can be simulated by rpg functions in a similar way. □

By Proposition 6 (2), Lemmas 11 and 12, we obtain the following theorem.

Theorem 13. RPL = (2,2)-MCFL. □

The following corollary follows from Proposition 6, Theorems 10 and 13.

Corollary 14. $(\text{CFL} \cup \text{SL-TAL}) \subset \text{ESL-TAL} \subseteq \text{TAL} \subset \text{RPL} = \text{(2,2)-MCFL}$. □

Whether the inclusion ESL-TAL $\subseteq$ TAL is proper or not is an open problem.

5 Conclusions

In this paper, some formal grammars for RNA secondary structure have been identified as subclasses of MCFG and their generative powers have been compared. To the authors' knowledge, the exact definition of pseudoknot in a biological or geometrical sense is not known and then it is difficult to answer which class of grammars is the minimum to represent pseudoknots. However, SL-TAG cannot generate RNA sequences obtained by repeating a simple pseudoknot shown in Figure 2 by ($*2$), and ESL-TAG (or ESL-MCFG) can be the minimum grammars which can represent such a class of pseudoknots.

Meanwhile, Satta and Schuler (1998) introduce a subclass of TAG (, which we will call *SS-TAG*) and show that ss-tals are recognizable in $O(n^5)$ time. The definition of ss-tag is slightly more general than that of esl-tag while keeping the constraint such that there exists (at most) one active node in the backbone. We conjecture that the generative power of ESL-TAG, SS-TAG and (2,2)-MCFG with $\deg(G) \leq 5$ are all the same.

Secondary structure is represented by a derivation (or derived) tree (see Figures 4 and 5). Comparison of the tree generative power of esl-tag and rpg is an interesting problem. To apply these grammars to RNA structure prediction, a probabilistic model should be introduced by extending these grammars such as stochastic cfg (Durbin et al., 1998), which is left as future work.

Acknowledgments

The authors would like to express their thanks to Professor S. Kanaya of Nara Institute of Science and Technology for his valuable discussions. The authors also thank Professor K. Asai of the University of Tokyo, Professor Y. Sakakibara of Keio University and members of his laboratory for their helpful comments.

References

Tatsuya Akutsu. 2000. *Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots*. Discrete Applied Mathematics, 104:45–62.

Anne Condon. 2003. *Problems on RNA secondary structure prediction and design*. ICALP2003, LNCS 2719:22–32.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis*. Cambridge University Press.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. *Tree adjunct grammars*. J. Computer & System Sciences, 10(1):136–163.

Aravind K. Joshi and Yves Schabes. 1997. *Tree adjoining grammars* in Grzegorz Rozenberg and Arto Salomaa, Eds., Handbook of Formal Languages, volume 3 (Beyond Words):69–123. Springer.

Aravind K. Joshi, K. Vijay-Shanker, and David J. Weir. 1988. *The convergence of mildly context-sensitive grammar formalisms*. Institute for Research in Cognitive Science, University of Pennsylvania.

Tadao Kasami, Hiroyuki Seki, and Mamoru Fujii. 1988. *Generalized context-free grammar and multiple context-free grammar*. IEICE Trans., J71-D(5):758–765 (in Japanese).

Tadao Kasami, Hiroyuki Seki, and Mamoru Fujii. 1988. *On the membership problem for head languages and multiple context-free languages*. IEICE Trans., J71-D(6):935–941 (in Japanese).

Yuki Kato, Hiroyuki Seki, and Tadao Kasami. 2004. *On the generative power of grammars for RNA secondary structure*. IEICE Technical Report, COMP-2003-75.

Alexandru Mateescu and Arto Salomaa. 1997. *Aspects of classical language theory* in Grzegorz Rozenberg and Arto Salomaa, Eds., Handbook of Formal Languages, volume 1 (Word, Language, Grammar):175–251. Springer.

Owen Rambow and Giorgio Satta. 1994. *A two-dimensional hierarchy for parallel rewriting systems*. IRCS Report 94-02, Institute for Research in Cognitive Science, University of Pennsylvania.

Elena Rivas and Sean Eddy. 2000. *The language of RNA: A formal grammar that includes pseudoknots*. Bioinformatics, 16(4):334–340.

Giorgio Satta and William Schuler. 1998. *Restrictions on tree adjoining languages*. Proc. 17th Int'l. Conf. on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING/ACL98).

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. *On multiple context-free grammars*. Theoretical Computer Science, 88:191–229.

Yasuo Uemura, Aki Hasegawa, Satoshi Kobayashi, and Takashi Yokomori. 1999. *Tree adjoining grammars for RNA structure prediction*. Theoretical Computer Science, 210:277–303.

K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. 1986. *Tree adjoining and head wrapping*. Proc. 11th Intl. Conf. on Computational Linguistics (COLING86):202–207.

K. Vijay-Shanker, David J. Weir and Aravind K. Joshi. 1987. *Characterizing structural descriptions produced by various grammatical formalisms*. Proc. 25th Annual Meeting of the Association for Computational Linguistics (ACL87).