Putting Some Weakly Context-Free Formalisms in Order

David Chiang
University of Pennsylvania

1. Introduction

A number of formalisms have been proposed in order to restrict tree adjoining grammar (TAG) to be weakly equivalent to context free grammar (CFG): for example, tree substitution grammar (TSG), tree insertion grammar (TIG), and regular-form TAG (RF-TAG); in the other direction, tree-local multicomponent TAG (TL-MCTAG) has been proposed as an extension to TAG which is weakly equivalent to TAG. These formalisms have been put to use in various applications. For example, Kroch and Joshi (1987) and others use TL-MCTAG for linguistic description; Bod (1992) uses TSG, and Chiang (2000) uses TIG, for statistical parsing; Shieber (1994) proposes to use TSG for generating synchronous TAG derivations for translation, and Dras (1999) uses RF-TAG for the same purpose. Although it is understood that these formalisms are useful because they have greater strong generative capacity (henceforth SGC) than their weakly-equivalent relatives, it is not always made clear what this actually means: sometimes it is understood in terms of phrase structures, sometimes in terms of a broader notion of structural descriptions (so Chomsky (1963)).

We take the latter view, and follow Miller (1999) in seeing phrase structures as just one of many possible interpretations of structural descriptions (alongside, for example, dependency structures). For Miller, structural descriptions themselves should never be compared (since they vary widely across formalisms), but only their interpretations. Thus SGC in the phrase-structure sense is one of several ways of testing SGC in the broader sense. However, not much effort has been made to demonstrate precisely how formalisms compare in these other ways.

In this paper we examine four formalisms—CFG, TIG, RF-TAG, and what we call *component-local scattered context grammar* (CL-SCG)—under four different interpretations, and find that TIG, RF-TAG, and CL-SCG all extend the expressivity of CFG in different ways (see Figure 1). These results show that it is possible to make formally precise statements about notions of generative capacity other than weak generative capacity (henceforth WGC), as a step towards articulating desiderata of formal grammars for various applications.

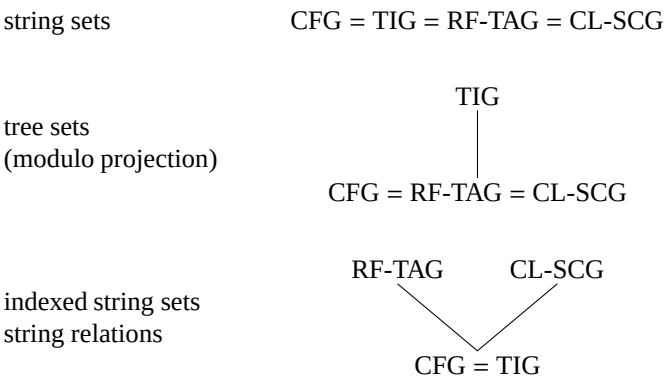


Figure 1: Summary of results. Edges denote strict inclusion (lower $\subset$ higher); = denotes equality.

2. Definitions

We assume familiarity with CFGs and TAGs, and proceed to define two restrictions on TAGs:

Definition 1. A *left* (or *right*) *auxiliary tree* is an auxiliary tree in which every frontier node to the right (resp., left) of the foot node is labeled with the empty string. A *tree insertion grammar* (Schabes and Waters, 1995) is a TAG in which all auxiliary trees are either left or right auxiliary trees, and adjunction is constrained so that:

- no left (right) auxiliary tree can be adjoined on any node that is on the spine of a right (left) auxiliary tree, and
- no adjunction is permitted on a node that is to the right (left) of the spine of a left (right) auxiliary tree.

 Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6), pp. 11–18. Universitá di Venezia.

Definition 2. We say that a TAG is in *regular form* (Rogers, 1994) if there exists some partial ordering $\preceq$ over nonterminal symbols such that if β is an auxiliary tree whose root and foot nodes are labeled X, and η is a node labeled Y on β's spine where adjunction is allowed, then $X \preceq Y$, and $X = Y$ only if η is a foot node. Thus adjunction at the foot node is allowed freely, adjunction at the middle of a spine is allowed only to a bounded depth, and adjunction at the root is not allowed at all.[1]

Next we define CL-SCG, first introducing the notion of an indexed string, which we will make use of in several places in this paper.

Definition 3. An *indexed string* is a pair $(w; I_w)$, where w is a string and I_w is an equivalence relation over string positions of w. An *indexed string n-tuple* is an $(n+1)$-tuple $(w_1, \ldots, w_n; I_w)$, where $w_1, \ldots, w_n$ are strings and I_w is an equivalence relation over string positions of the w_i. We notate these equivalence relations using boxed indices.

Definition 4. A *local scattered context grammar*[2] is a tuple $G = (N, T, P, S)$, where

- N and T are finite, disjoint sets of nonterminal symbols and terminal symbols, respectively,
- $S \in N$ is the start symbol, and
- P is a finite set of productions of the form

$$(A_1, \ldots, A_n) \rightarrow (\alpha_1, \ldots, \alpha_n; I_\alpha)$$

  where $n \geq 1$, $A_i \in N$ and $(\alpha_i, \ldots, \alpha_n; I_\alpha)$ is an indexed tuple of strings over $(N \cup T)^*$.

We write $(\gamma; I_\gamma) \Rightarrow_G (\delta, I_\delta)$, where $(\gamma; I_\gamma)$ and $(\delta; I_\delta)$ are indexed strings, if and only if there exists a production $(A_1, \ldots, A_n) \rightarrow (\alpha_1, \ldots, \alpha_n; I_\alpha) \in P$ such that

$$\gamma = \gamma_0 A_1 \gamma_1 \cdots A_n \gamma_n$$

where $A_1, \ldots, A_n$ comprise an equivalence class of I_γ, and

$$\delta = \gamma_0 \alpha_1 \gamma_1 \cdots \alpha_n \gamma_n,$$

where any nonterminal instances in the γ_i whose corresponding instances in γ are equivalent under I_γ are also equivalent under I_δ, as are any nonterminal instances in the α_i which are equivalent under I_α, and nothing else.

Let I_S be the equivalence relation on string positions of S that relates S to itself. Then

$$L(G) = \{w \mid (S; I_S) \overset{*}{\Rightarrow}_G (w; I_w) \text{ for some } I_w\}.$$

We say that a local scattered context grammar is *component-local* if for each production $(A_1, \ldots, A_n) \rightarrow (\alpha_1, \ldots, \alpha_n; I_\alpha) \in P$, a nonterminal instance in α_i and a nonterminal instance in α_j are equivalent under I_α only if $i = j$.

We call this restriction "component-local" by analogy with tree-local MCTAG (Weir, 1988), because a production simultaneously rewrites multiple nonterminals with multiple components, but all those nonterminals must have come from the same component.

3. The formalisms considered as string-rewriting systems

Proposition 1. *CFG, TIG, RF-TAG, and CL-SCG are weakly equivalent.*

Proof. The weak equivalence of TIG to CFG was shown by Schabes and Waters (1995). The basic idea is to flatten each elementary tree into a CFG production, discarding every foot node, but adding a nonterminal to the left (right) of every node at which left-adjunction (resp., right-adjunction) is possible.

1. Note that this definition is stricter than Rogers' original definition, which allows "redundant" elementary trees. His parsing algorithm does not produce all possible derivations under the original definition, but does under the stricter definition.
2. This definition is based on local unordered scattered context grammar (Rambow and Satta, 1999), but is simplified in two ways: first, our scattered contexts are ordered rather than unordered; second, our productions explicitly specify which sets of nonterminals may be rewritten. We do not believe either of these simplifications affects the results shown here.

The weak equivalence of RF-TAG to CFG was shown by Rogers (1994). The basic idea is to break each elementary tree into CFG productions, augmenting the nonterminal alphabet with a stack of bounded depth to keep track of non-foot adjunctions.

For any CL-SCG, a weakly equivalent CFG can be obtained by a construction analogous to that for tree-local multicomponent TAG (Weir, 1988). Given a CL-SCG $G = (N, T, P, S)$, let f be the maximum number of components of (the left- or right-hand side of) any production in P, and let $N' = (\bigcup_{i=1}^{f} N^i \times \{1, \ldots f\})$. We want to break each production of P into its component CFG productions, using this augmented nonterminal alphabet to ensure that all the component productions are used together. To do this, construct P' from P as follows:

- For every nonterminal occurrence A_i on a left-hand side $(A_1, \ldots A_n)$, replace A_i with $(A_1, \ldots, A_n, i)$.
- For every nonterminal occurrence A_i on a right-hand side $(\alpha_1, \ldots, \alpha_n; I_\alpha)$, if A_i is the ith member of an equivalence class of I_α whose members are $A_1, \ldots, A_n$, in that order, replace A_i with $(A_1, \ldots, A_n, i)$.

Then construct P'' from P' as follows: if $(A'_1, \ldots, A'_n) \to (\alpha'_1, \ldots, \alpha'_n) \in P'$, then $A'_i \to \alpha'_i \in P''$ for all i between 1 and n. Then the CFG $(N', T, P'', (S, 1))$ is weakly equivalent to G. □

4. The formalisms considered as tree-rewriting systems

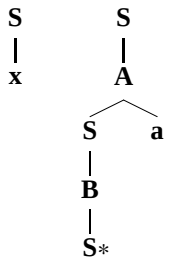
CFG generates only local sets, whereas CL-SCG and RF-TAG generate some recognizable sets which are not local. However, any recognizable set can be made from a local set by projection of labels (Thatcher, 1967). If we factor out this distinction, then CL-SCG and RF-TAG are no more powerful than CFG:

Proposition 2. *For any RF-TAG (or CL-SCG) G, there is a CFG G' and a projection of labels π such that $T(G) = \{\pi(t) \mid t \in T(G')\}$.*

Proof. The constructions used to prove the weak equivalence of these formalisms to CFG also preserve trees, modulo projection of labels. □

Proposition 3. *TIG can generate a tree set which is not recognizable.*

Proof. As was observed by Schabes and Waters (1995), the following TIG generates a non-recognizable set:



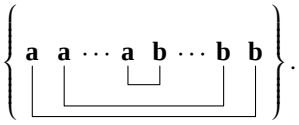
When the path set is intersected with $\{\mathbf{SA}\}^*\mathbf{S}\{\mathbf{BS}\}^*\mathbf{x}$, the result is $\{(\mathbf{SA})^n\mathbf{S}(\mathbf{BS})^n\mathbf{x} \mid n \geq 0\}$, a non-regular set. □

5. The formalisms considered as linking systems

We now define derivational generative capacity (henceforth DGC, introduced by Becker et al. (1992)), which measures the generative capacity of what Miller (1999) calls "linking systems."

Definition 5. We say that a grammar G *index-generates* an indexed string $(a_1 \cdots a_n; I_w)$ (see Definition 3) if G generates $a_1 \cdots a_n$ such that a_i and a_j are equivalent under I_w if and only if a_i and a_j are contributed by the same derivation step. The *derivational generative capacity* of a grammar G is the set of all indexed string sets index-generated by G.

In this section we notate indexed strings by drawing links (following Joshi (1985)) between all the positions of each equivalence class. Thus the CFG $\mathbf{X} \to \mathbf{aXb} \mid \epsilon$ index-generates the familiar-looking indexed string set



We saw in the previous section that with respect to SGC (in the sense of phrase structures), TIG was more powerful than CFG, whereas RF-TAG and CL-SCG were no more powerful than CFG. With respect to DGC, the opposite obtains.

Proposition 4. *CFG and TIG are derivationally equivalent.*

Proof. The construction given by Schabes and Waters (1995) preserves derivations, and therefore preserves indexed strings. □

On the other hand, RF-TAG and CL-SCG both have greater DGC than CFG (and TIG). Moreover, they extend CFG in different ways, because each is able to generate an indexed string set that the other is not.

Lemma 5 (indexed pumping lemma). *Let L be an indexed string set generated by a CFG (or CL-SCG). Then there is a constant n such that if $(z; I_z)$ is in L and $|z| \geq n$, then z may be rewritten as $uvwxy$, with $|vx| > 0$ and $|vwx| \leq n$, such that for all $i \geq 1$, there is an equivalence relation I_z^i such that $(uv^iwx^iy; I_z^i)$ is in L and I_z^i does not relate any positions in w to any positions in u or y.*

Proof. The proof is analogous to that of the standard pumping lemma (Hopcroft and Ullman, 1979). However, since the grammar cannot be put into Chomsky normal form, we let $n = m^k$ instead of 2^k, where k is the size of the nonterminal alphabet and m is the maximum number of symbols on any right-hand side. The key difference from the standard proof is the observation that since, for each i, the derivation of uv^iwx^iy can be written as

$$S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uv^iAx^iy \overset{*}{\Rightarrow} uv^iwx^iy$$

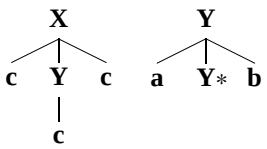
for some nonterminal A, no position in w can be contributed by the same derivation step as any position in u or y.

The generalization to CL-SCG is straightforward, since a CL-SCG G can be converted into a CFG G' which generates the same trees. G' will not generate the same indexed strings as G; nevertheless, the equivalence relations index-generated by G can only relate terminal instances which are first cousins in the derived tree, so for $i \geq 1$, it remains the case that no position in w is related to any position in u or y. □

Proposition 6. *The following indexed string set is index-generable by an RF-TAG but not by any CL-SCG:*

$$L_1 = \left\{ \mathbf{c}\ \mathbf{a}\ \mathbf{a} \cdots \mathbf{a}\ \mathbf{c}\ \mathbf{b} \cdots \mathbf{b}\ \mathbf{b}\ \mathbf{c} \right\}$$

Proof. The following RF-TAG generates L_1:



But suppose L_1 is index-generated by some CFG or CL-SCG G. For any n given by the indexed pumping lemma, let $z = \mathbf{ca}^n\mathbf{cb}^n\mathbf{c}$ satisfy the conditions of the pumping lemma. It must be the case that v and x contain only **a**'s and **b**'s, respectively, or else $uv^iwx^iy \notin L_1$. But then u, w, and y would each have to contain one of the **c**'s, and since the **c**'s are all related, this contradicts the pumping lemma. □

Proposition 7. *The following indexed string set (consisting of a single string) is generable by a CL-SCG but not by any RF-TAG, nor indeed by any TAG:*

$$L_2 = \left\{ \mathbf{a}\ \mathbf{b}\ \mathbf{a}\ \mathbf{b}\ \mathbf{a}\ \mathbf{b} \right\}$$

Proof. The following CL-SCG generates L_2:

$$(\mathbf{S}) \rightarrow (\mathbf{aB}_{\boxed{1}}\mathbf{aB}_{\boxed{1}}\mathbf{aB}_{\boxed{1}})$$

$$(\mathbf{B}, \mathbf{B}, \mathbf{B}) \rightarrow (\mathbf{b}, \mathbf{b}, \mathbf{b})$$

But L_2 cannot be generated by any TAG. In general, if $a_1 \cdots b \cdots a_2$ is index-generable by a TAG such that a_1 and a_2 are related, then either the tree that contributes a_1 and a_2 (call it β_a) adjoins into the tree that contributes b (call it β_b) such that its foot node dominates b, or else β_b adjoins into β_a. In the case of L_2, suppose that the **a**'s are contributed by β_a, and the **b**'s are contributed by β_b. If β_a adjoins into β_b, its foot node must dominate both of the **b**'s, which is a contradiction; similarly if β_b adjoins into β_a. □

6. The formalisms considered as local synchronous systems

In a *local synchronous system* (Aho and Ullman, 1969; Shieber, 1994; Rambow and Satta, 1996), two grammars are constrained to generate pairs of strings via isomorphic derivations (up to relabeling and reordering of sisters). Although translation via a synchronous grammar is not really an "interpretation" in Miller's sense, nevertheless, because a synchronous derivation in effect realizes a single derivation structure in two different ways, we expect the set of string relations generated by a local synchronous system to reveal something about the relationship between derivations and strings that weak generative capacity does not.

Definition 6. A *synchronous CFG* is a tuple $G = (N, T, P, S)$, where N, T, and S are as in ordinary CFG, and P is a set of productions of the form

$$(A : A') \rightarrow (\alpha : \alpha'; I_\alpha)$$

where $A, A' \in N$, $\alpha, \alpha' \in (N \cup T)^*$, and I_α is a bijection between nonterminal instances in α and nonterminal instances in α'.

We write $(\gamma : \gamma'; I_\gamma) \Rightarrow_G (\delta : \delta'; I_\delta)$, where $(\gamma : \gamma'; I_\gamma)$ and $(\delta : \delta'; I_\delta)$ are indexed string pairs, if and only if there exists a production $(A : A') \rightarrow (\alpha : \alpha'; I_\alpha) \in P$ such that

$$\gamma = \gamma_0 A \gamma_1 \qquad \gamma' = \gamma'_0 A' \gamma'_1$$

where A and A' are related under I_γ, and

$$\delta = \gamma_0 \alpha \gamma_1 \qquad \delta' = \gamma'_0 \alpha' \gamma'_1$$

where any nonterminal instances in γ_0, γ_1, γ'_0, or γ'_1 whose corresponding instances in γ are equivalent under I_γ are also equivalent under I_δ, as are any nonterminal instances in α and α' which are equivalent under I_α, and nothing else.

Let I_S be the equivalence relation on string positions of $(S : S)$ which relates both instances of S to each other. Then the *weak generative capacity* of G is the string relation

$$L(G) = \{(w : w') \mid (S : S; I_S) \stackrel{*}{\Rightarrow}_G (w : w'; I_w)\}.$$

The definition of synchronous TAG (Shieber, 1994) is analogous, but with bijections between adjunction sites instead of bijections between nonterminal instances; synchronous TIG and synchronous RF-TAG are just restrictions of synchronous TAG. The definition of synchronous CL-SCG is also analogous, but with bijections between equivalence classes of nonterminal instances instead of bijections between nonterminal instances. These four synchronous formalisms relate to each other in the same way as the linking systems of the previous section.

Proposition 8. *Synchronous CFG and synchronous TIG are weakly equivalent.*

Proof. The construction given by Schabes and Waters (1995) preserves derivations, and therefore preserves string pairs. □

Lemma 9 (synchronous pumping lemma). *Let L be a string relation generated by a synchronous CFG (or synchronous CL-SCG). Then there is a constant n such that if $(z : z')$ is in L and $|z| \geq n$ and $|z'| \geq n$, then $(z : z')$ may be written as $(uwy : u'w'y')$, and there exist strings v, x, v', x', such that $|vxv'x'| > 0$, $|vwx| \leq n$, $|v'w'x'| \leq n$, and for all $i \geq 0$, $(uv^iwx^iy : u'v'^iw'x'^iy')$ is in L.*

Proof. The proof is again analogous to that of the standard pumping lemma: let $G = (N, T, P, S)$ be a synchronous CFG generating L. We choose n as the proof of the standard lemma would if the nonterminal alphabet were $N \times N$. This guarantees the existence of a pair of corresponding paths in the derivation of $(z : z')$ such that the same pair of nonterminals $(A : A')$ occurs twice:

$$(S : S) \stackrel{*}{\Rightarrow} (uAy : u'A'y') \stackrel{*}{\Rightarrow} (u_1vAxy_1 : u'_1v'A'x'y'_1) \stackrel{*}{\Rightarrow} (u_1vwxy_1 : u'_1v'w'x'y'_1)$$

If we let $u = u_1v$ and $y = xy_1$, and likewise $u' = u'_1v'$ and $y' = x'y'_1$, then $(z : z') = (uwy : u'w'y')$, and for all $i \geq 0$, $(uv^iwx^iy : u'v'^iw'x'^iy') \in L$.

The CL-SCG case is similar but quite messy. A CL-SCG derivation tree has the same height as its derived tree, minus one. Therefore we can choose n such that any derivation of $(z : z')$ where $|z| \geq n$ and $|z'| \geq n$ must have a pair of corresponding paths such that the same set of nonterminals occurs twice (new material underlined for clarity):

$$\begin{aligned}
(S : S) &\stackrel{*}{\Rightarrow} (u_0A_1u_1 \cdots A_ku_k \cdots A_nu_n : u'_0A'_1u'_1 \cdots A'_{k'}u_{k'} \cdots A'_{n'}u'_{n'}) \\
&\stackrel{*}{\Rightarrow} (u_0\underline{v_1}u_1 \cdots \underline{v_{k0}A_1v_{k1} \cdots A_nv_{kn}}u_k \cdots \underline{v_n}u_n : u'_0\underline{v'_1}u'_1 \cdots \underline{v'_{k'0}A'_1v'_{k'1} \cdots A'_{n'}v'_{k'n'}}u'_{k'} \cdots \underline{v'_{n'}}u'_{n'}) \\
&\stackrel{*}{\Rightarrow} (u_0v_1u_1 \cdots v_{k0}\underline{w_1}v_{k1} \cdots \underline{w_n}v_{kn}u_k \cdots v_nu_n : u'_0v'_1u'_1 \cdots v'_{k'0}\underline{w'_1}v'_{k'1} \cdots \underline{w'_{n'}}v'_{k'n'}u'_{k'} \cdots v'_{n'}u'_{n'})
\end{aligned}$$

If we let

$$\begin{aligned}
u &= u_0v_1u_1 \cdots v_{k-1}u_{k-1} & u' &= u'_0v'_1u'_1 \cdots v'_{k'-1}u'_{k'-1} \\
v &= v_{k0}v_1v_{k1} \cdots v_{k-1}v_{k,k-1} & v' &= v'_{k'0}v'_1v'_{k'1} \cdots v'_{k'-1}v'_{k',k'-1} \\
w &= v_{k0}w_1v_{k1} \cdots w_nv_{kn} & w' &= v'_{k'0}w'_1v'_{k'1} \cdots w'_{n'}v'_{k'n'} \\
x &= v_{k,k+1}v_{k+1} \cdots v_nv_{kn} & x' &= v'_{k',k'+1}v'_{k'+1} \cdots v'_{n'}v'_{k'n'} \\
y &= u_kv_{k+1}u_{k+1} \cdots v_nu_n & y' &= u'_{k'}v'_{k'+1}u'_{k'+1} \cdots v'_{n'}u'_{n'}
\end{aligned}$$

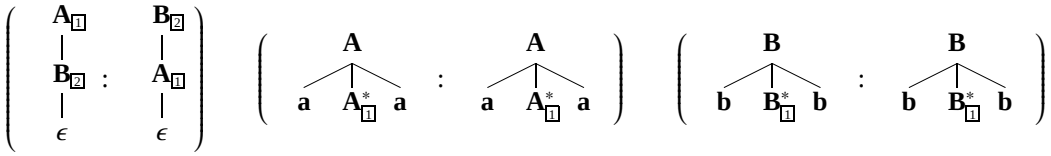
then $(z : z') = (uwy : u'w'y')$, and for all $i \geq 0$, $(uv^iwx^iy : u'v'^iw'x'^iy') \in L$. □

Proposition 10. *The string relation*

$$L_3 = \{(\mathbf{a}^m\mathbf{b}^n\mathbf{c}^n\mathbf{d}^m : \mathbf{b}^n\mathbf{a}^m\mathbf{d}^m\mathbf{c}^n)\}$$

is generable by a synchronous RF-TAG but not by any synchronous CL-SCG.

Proof. The following synchronous RF-TAG generates L_3:



But suppose L_3 can be generated by some CL-SCG G. For any n given by the pumping lemma, let $(z : z') = (\mathbf{a}^n\mathbf{b}^n\mathbf{c}^n\mathbf{d}^n : \mathbf{b}^n\mathbf{a}^n\mathbf{c}^n\mathbf{d}^n)$ satisfy the conditions of the pumping lemma. Then $vxv'x'$ must contain only **a**'s and **d**'s, or only **b**'s and **c**'s, otherwise $(uv^iwx^iy : u'v'^iw'x'^iy')$ will not be in L. But in the former case, $|vwx| > n$, and in the latter case, $|v'w'x'| > n$, which is a contradiction.[3] □

Proposition 11. *There is a string relation which is generable by a synchronous CL-SCG but not by any synchronous RF-TAG, nor indeed by any synchronous TAG.*

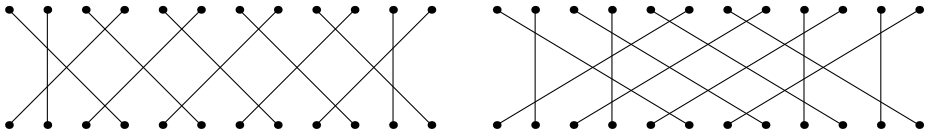
Proof. Define $L_4(k)$ to be

$$\{(w_1 \cdots w_{2n} : w_{\pi_n^k(1)} \cdots w_{\pi_n^k(2n)}) \mid w_i \in \{\sigma(i)\}^*, n \geq 1\},$$

where

$$\sigma(i) = \begin{cases} \mathbf{a} & \text{if } i \text{ is odd,} \\ \mathbf{b} & \text{if } i \text{ is even,} \end{cases}$$

3. An analogous result for synchronous regular-form two-level TAG was shown by Chiang et al. (2000).



π_6^3 π_6^5

Figure 2: Example diagrams of π_n^k.

and

$$\pi_n^k(i) = \begin{cases} i+k & \text{if } i \text{ is odd and } i+k \leq 2n, \\ i-k & \text{if } i \text{ is even and } i-k \geq 1, \\ i & \text{otherwise.} \end{cases}$$

See Figure 2 for examples of π_n^k. The following synchronous CL-SCGs generate $L_4(3)$ and $L_4(5)$, respectively:

$$\begin{aligned} (\mathbf{S} : \mathbf{S}) &\rightarrow (\mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}} : \mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}}) \\ (\mathbf{X}, \mathbf{Y} : \mathbf{X}, \mathbf{Y}) &\rightarrow (\mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}}, \mathbf{B}_{\boxed{3}} : \mathbf{X}_{\boxed{1}}\mathbf{B}_{\boxed{3}}\mathbf{Y}_{\boxed{1}}, \mathbf{A}_{\boxed{2}}) \\ &\mid (\epsilon, \mathbf{B}_{\boxed{1}} : \epsilon, \mathbf{B}_{\boxed{1}}) \\ (\mathbf{A} : \mathbf{A}) &\rightarrow (\mathbf{a}\mathbf{A}_{\boxed{1}} : \mathbf{a}\mathbf{A}_{\boxed{1}}) \mid (\epsilon : \epsilon) \\ (\mathbf{B} : \mathbf{B}) &\rightarrow (\mathbf{b}\mathbf{B}_{\boxed{1}} : \mathbf{b}\mathbf{B}_{\boxed{1}}) \mid (\epsilon : \epsilon) \end{aligned}$$

$$\begin{aligned} (\mathbf{S} : \mathbf{S}) &\rightarrow (\mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}}\mathbf{A}_{\boxed{3}}\mathbf{Z}_{\boxed{1}} : \mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}}\mathbf{A}_{\boxed{3}}\mathbf{Z}_{\boxed{1}}) \\ (\mathbf{X}, \mathbf{Y}, \mathbf{Z} : \mathbf{X}, \mathbf{Y}, \mathbf{Z}) &\rightarrow (\mathbf{X}_{\boxed{1}}\mathbf{A}_{\boxed{2}}\mathbf{Y}_{\boxed{1}}\mathbf{A}_{\boxed{3}}\mathbf{Z}_{\boxed{1}}, \mathbf{B}_{\boxed{4}}, \mathbf{B}_{\boxed{5}} : \mathbf{X}_{\boxed{1}}\mathbf{B}_{\boxed{4}}\mathbf{Y}_{\boxed{1}}\mathbf{B}_{\boxed{5}}\mathbf{Z}_{\boxed{1}}, \mathbf{A}_{\boxed{2}}, \mathbf{A}_{\boxed{3}}) \\ &\mid (\epsilon, \mathbf{B}_{\boxed{1}}, \mathbf{B}_{\boxed{2}} : \epsilon, \mathbf{B}_{\boxed{1}}, \mathbf{B}_{\boxed{2}}) \\ (\mathbf{A} : \mathbf{A}) &\rightarrow (\mathbf{a}\mathbf{A}_{\boxed{1}} : \mathbf{a}\mathbf{A}_{\boxed{1}}) \mid (\epsilon : \epsilon) \\ (\mathbf{B} : \mathbf{B}) &\rightarrow (\mathbf{b}\mathbf{B}_{\boxed{1}} : \mathbf{b}\mathbf{B}_{\boxed{1}}) \mid (\epsilon : \epsilon) \end{aligned}$$

But suppose $L_4(3)$ is generated by a synchronous CFG G. Let r be the order of G, that is, the maximum number of nonterminals in either half of the right-hand side of any production. By means of closure properties it can be shown that there is a synchronous CFG G' of order r which generates the language

$$\{(w_1 \cdots w_{2n} : w_{\pi_n^3(1)} \cdots w_{\pi_n^3(2n)}) \mid w_i \in \{\mathbf{c}_i\}^*\},$$

where $n = \max\{3, \lceil \frac{r+1}{2} \rceil\}$ and π_n^k is as above. But this leads to a contradiction by means of the same argument used by Aho and Ullman (1969) to show that the synchronous CFGs of order $r+1$ properly include the synchronous CFGs of order r. The basic idea is as follows: we say that a production *covers* $\mathbf{c}_i$ if its right-hand side derives an unbounded number of $\mathbf{c}_i$'s. Then it can be shown that any production which covers two of the $\mathbf{c}_i$ must cover all of them. So there would have to be a production covering all $2n$ of the $\mathbf{c}_i$, with $2n$ nonterminals on its right-hand side, each of which gets rewritten with a production covering only one of the $\mathbf{c}_i$. But since $2n > r$, this is a contradiction.

For the TAG case, suppose $L_4(5)$ is generated by a synchronous TAG G, and again let r be the order of G. By means of closure properties it can be shown that there is a synchronous TAG G' of order r which generates the language

$$\{(w_1 \cdots w_{2n} : w_{\pi_n^5(1)} \cdots w_{\pi_n^5(2n)}) \mid w_i \in \{\mathbf{c}_i\}^*\},$$

where $n = 4r+1$ and π_n^k is as above. This case is more difficult because unlike a CFG nonterminal, a TAG auxiliary tree has a hole in its span created by its foot node. But it can be shown that any elementary tree pair which covers three of the $\mathbf{c}_i$, such that none of them are separated by a foot node in either the domain or the range, must cover all the $\mathbf{c}_i$ except perhaps one. Given our choice of n, this suffices to show that G' cannot exist.

□

7. Conclusion

Joshi (2000) poses the question, "How much strong generative power can be squeezed out of a formal system without increasing its weak generative power?" The shifting relations between these four formalisms under different interpretations (see Figure 1) show that there is more than one way to answer this question.

First, there is more than one way to measure strong generative power. That TIG generates more tree sets than RF-TAG or CL-SCG but fewer indexed string sets demonstrates a point noted by Becker et al. (1992): that SGC (in the sense of phrase structures) and DGC are orthogonal notions. This is because SGC (in the sense of phrase structures) is based on the tree yield function, which can be chosen somewhat independently of the string yield function. On the other hand, DGC and synchronous WGC, which are both based on the string yield function, order our four formalisms in the same way.

Second, there is more than one way to squeeze a formal system. RF-TAG and CL-SCG are incommensurate with respect to both DGC and synchronous WGC; that is, each is able to do something the other is not. Thus, even under a particular interpretation of strong generative power, the question is not only, how much strong generative power can be squeezed, but also, in what ways? Characterizing the different ways in which strong generative power can be both measured and squeezed is a task for future research.

Acknowledgements

This research was supported in part by NSF grant SBR-89-20230. I would like to thank Anoop Sarkar, William Schuler, Aravind Joshi, and the three anonymous reviewers for their valuable comments. *S. D. G.*

References

Aho, A. V. and J. D. Ullman. 1969. Syntax Directed Translations and the Pushdown Assembler. *J. Comp. Sys. Sci.*, 3:37–56.

Becker, Tilman, Owen Rambow and Michael Niv. 1992. The derivational generative power of formal systems, or, Scrambling is beyond LCFRS. Technical Report IRCS-92-38, Institute for Research in Cognitive Science, University of Pennsylvania. Presented at MOL3.

Bod, Rens. 1992. Data-oriented parsing. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, Nantes.

Chiang, David. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting of the Assocation for Computational Linguistics*, pages 456–463, Hong Kong.

Chiang, David, William Schuler and Mark Dras. 2000. Some remarks on an extension of synchronous TAG. In *Proceedings of the Fifth International Workshop on TAG and Related Formalisms (TAG+5)*, pages 61–66.

Chomsky, Noam. 1963. Formal properties of grammars. In R. Duncan Luce, Robert R. Bush and Eugene Galanter, editors, *Handbook of Mathematical Psychology*. Wiley, New York, pages 323–418.

Dras, Mark. 1999. A meta-level grammar: redefining synchronous TAG for translation and paraphrase. In *Proceedings of the 37th Annual Meeting of the Assocation for Computational Linguistics*, pages 80–87, College Park, MD.

Hopcroft, John E. and Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages, and computation*. Reading, MA: Addison-Wesley.

Joshi, Aravind K. 1985. Tree adjoining grammars: How much context-sensitivity is necessary for assigning structural descriptions? In David Dowty, Lauri Karttunen and Arnold Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, pages 206–250.

Joshi, Aravind K. 2000. Relationship between strong and weak generative power of formal systems. In *Proceedings of the Fifth International Workshop on TAG and Related Formalisms (TAG+5)*, pages 107–113.

Kroch, Anthony and Aravind K. Joshi. 1987. Analyzing extraposition in a tree adjoining grammar. In Geoffrey J. Huck and Almerindo E. Ojeda, editors, *Discontinuous Constituency*. Academic Press, Orlando.

Miller, Philip H. 1999. *Strong Generative Capacity: The Semantics of Linguistic Formalism*. CSLI lecture notes, number 103. Stanford: CSLI Publications.

Rambow, Owen and Giorgio Satta. 1996. Synchonous Models of Language. In *Proceedings of the 34th Annual Meeting of the Assocation for Computational Linguistics*, pages 116–123, Santa Cruz, CA.

Rambow, Owen and Giorgio Satta. 1999. Independent parallelism in finite copying parallel rewriting systems. *Theoretical Computer Science*, 223:887–120.

Rogers, James. 1994. Capturing CFLs with tree adjoining grammars. In *Proceedings of the 32nd Annual Meeting of the Assocation for Computational Linguistics*, pages 155–162, Las Cruces, NM.

Schabes, Yves and Richard C. Waters. 1995. Tree insertion grammar: a cubic-time parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Computational Linguistics*, 21:479–513.

Shieber, Stuart M. 1994. Restricting the weak generative capacity of synchronous tree-adjoining grammars. *Computational Intelligence*, 10(4):371–385, November. Special Issue on Tree Adjoining Grammars.

Thatcher, J. W. 1967. Characterizing Derivation Trees of Context-Free Grammars through a Generalization of Finite Automata Theory. *J. Comp. Sys. Sci.*, 1:317–322.

Weir, David J. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. Ph.D. thesis, Univ. of Pennsylvania.