ISSN 1346-5597

National Institute of Informatics

NII Technical Report

Lexicalization of Second-Order ACGs

Makoto Kanazawa and Ryo Yoshinaka

NII-2005-012E
Aug. 2005

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

Lexicalization of Second-Order ACGs

Makoto Kanazawa[1] and Ryo Yoshinaka[2]

[1,2] *National Institute of Informatics*
[2] *Graduate School of Interdisciplinary Information Studies, University of Tokyo*

Abstract

We use techniques familiar from the theory of context-free grammars to show that, given a second-order abstract categorial grammar (ACG), one can effectively find a lexicalized second-order ACG whose object language is the object language of the original ACG minus combinators.

1 Introduction

An abstract categorial grammar (ACG) (de Groote 2001) is *second-order* if every abstract constant has a type of the form $q_1 \to \cdots \to q_k \to p$ ($k \geq 0$), where $q_1, \ldots, q_k, p$ are atomic types. An ACG is *lexicalized* if the object image of every abstract constant contains a constant. In this note, we use techniques familiar from the theory of context-free grammars to show that, given a second-order ACG, one can effectively find a lexicalized second-order ACG whose object language is the object language of the original ACG minus combinators. This result was stated in Yoshinaka and Kanazawa 2005, but the proof in that paper was in error. The purpose of this note is to correct this mistake and at the same time relate the result to familiar properties of context-free grammars.

Let $\mathscr{G} = \langle \Sigma, \Sigma_{\text{obj}}, \mathscr{L}, s \rangle$ be a second-order ACG with $\Sigma = \langle A, C, \tau \rangle$. The set C is partitioned into the set D of *lexical* constants (abstract constants whose image under $\mathscr{L}$ contains at least one constant) and the set E of *non-lexical* constants. That $\mathscr{G}$ is second-order means that there is a *regular tree grammar* $G_{\mathscr{G}}$ generating the abstract language of $\mathscr{G}$ whose productions are of the form $p \to cq_1 \ldots q_k$, where $p, q_1, \ldots, q_k \in A$ and $c \in C$. Our procedure for eliminating non-lexical constants from $\mathscr{G}$ corresponds to eliminating from $G_{\mathscr{G}}$ all productions of the form $p \to eq_1 \ldots q_k$, where $e \in E$. We break down the procedure into three steps, each corresponding to a modification of a familiar transformation on CFGs, as summarized in Table 1. Overall, the procedure roughly corresponds to a procedure for converting a context-free grammar into *Greibach normal form*.

	non-lexical constants eliminated from ACG	productions eliminated from RTG	technique applied
1.	nullary	$p \to e$	elimination of ϵ-productions
2.	unary	$p \to eq$	elimination of unit productions
3.	k-ary ($k \geq 2$)	$p \to eq_1 \ldots q_k$ ($k \geq 2$)	left-corner transform

Table 1: Three steps for eliminating non-lexical constants

Formally, we describe each of the first two steps of the procedure in terms of a simple transformation of a regular tree grammar into another one generating a certain subset of the original language. These transformations induce transformations on ACGs because second-order ACGs can be translated into regular tree grammars and regular tree grammars can be translated back into second-order ACGs. The third step of our procedure is described in terms of a transformation of a regular tree grammar into a *linear context-free tree grammar* generating the same language, relying on a translation from linear context-free tree grammars into second-order ACGs (de Groote and Pogodalla 2004).

2 The left-corner transform

The last step of our procedure for lexicalizing second-order ACGs is based on a transformation on context-free grammars known as the *left-corner transform* (Rosenkrantz and Lewis 1970), which converts a possibly left-recursive context-free grammar into an equivalent non-left-recursive one. Top-down parsing with the left-corner transformed grammar simulates left-corner parsing with the original grammar. The left-corner transform can be used as an intermediate step in converting a context-free grammar to *Greibach normal form*.

A *context-free grammar* is a quadruple $G = \langle N, T, P, S\rangle$, where N is a finite set of *nonterminals*, T is a finite set of *terminals* (disjoint from N), S is a member of N (called the *start symbol*), and P is a finite set of *productions* of the form $A \to \beta$, where $A \in N$ and $\beta \in (N \cup T)^*$. An *ϵ-production* is a production of the form $A \to \epsilon$. A *unit production* is a production of the form $A \to B$, where B is a nonterminal. For $\alpha, \alpha' \in (N \cup T)^*$, the relation $\alpha \Rightarrow_G \alpha'$ holds if there is a production $A \to \beta$ such that $\alpha = \alpha_1 A \alpha_2$ and $\alpha' = \alpha_1 \beta \alpha_2$ for some $\alpha_1, \alpha_2 \in (N \cup T)^*$. A context-free grammar $G = \langle N, T, P, S\rangle$ is *left-recursive* if $A \Rightarrow_G^* A\alpha$ for some $A \in N$ and $\alpha \in (N \cup T)^*$.

We use the following variant of Rosenkrantz and Lewis's (1970) original construction:

Definition 1. Let $G = \langle N, T, P, S\rangle$ be a context-free grammar with no ϵ-production or unit production. The *left-corner transform* of G is $G' = \langle N', T, P', S\rangle$, where

$$N' = N \cup \{\, B{-}A \mid A, B \in N \,\}$$

and P' is defined as follows:

- For each production in P of the form $A \to b\ \beta$ where $b \in T$ and $\beta \in (N \cup T)^*$, P' will have the production

$$A \to b\ \beta \tag{I}$$

and the production of the form

$$D \to b\ \beta\ D{-}A \tag{II}$$

for each $D \in N$.

- For each production in P of the form $A \rightarrow B\ \beta$ where $B \in N$ and $\beta \in (N \cup T)^*$, P' will have the production

$$A{-}B \rightarrow \beta \tag{III}$$

and the production of the form

$$D{-}B \rightarrow \beta\ D{-}A \tag{IV}$$

for each $D \in N$.

The grammar G' in Definition 1 corresponds to the result of eliminating ϵ-productions from Rosenkrantz and Lewis's (1970) transform.[1] It is easy to see that G' is non-left-recursive.

Proposition 2 (Rosenkrantz and Lewis). *Let G be a context-free grammar without ϵ- or unit productions, and let G' be the left-corner transform of G. Then $L(G) = L(G')$.*

For the sake of completeness, we provide a proof of Proposition 2.

Proof. First, we prove by simultaneous induction on $n \geq 1$ that the following hold for every $D, B \in N$ and $w \in T^+$:

(i) $D \Rightarrow_G^n w$ implies $D \Rightarrow_{G'}^n w$;

(ii) $D \Rightarrow_G^n Bw$ implies $D{-}B \Rightarrow_{G'}^n w$.

To prove (i), suppose $D \Rightarrow_G^n w$. Then we must have

$$\begin{gathered} D \Rightarrow_G^{n_1} A\alpha \Rightarrow_G b\beta\alpha \Rightarrow_G^{n_2+n_3} bw_1w_2 = w, \\ \beta \Rightarrow_G^{n_2} w_1, \\ \alpha \Rightarrow_G^{n_3} w_2, \end{gathered}$$

where $n = n_1 + n_2 + n_3 + 1$ and $A \rightarrow b\beta \in P$. The induction hypothesis about (i) implies $\beta \Rightarrow_{G'}^{n_2} w_1$. If $n_1 = 0$, then $D = A$, $\alpha = w_2 = \epsilon$, $n_3 = 0$, and schema (I) gives $D \rightarrow b\beta \in P'$. Thus,

$$D \Rightarrow_{G'} b\beta \Rightarrow_{G'}^{n_2} bw_1 = w.$$

If $n_1 \geq 1$, since $D \Rightarrow_G^{n_1+n_3} Aw_2$, the induction hypothesis about (ii) gives $D{-}A \Rightarrow_{G'}^{n_1+n_3} w_2$. We have $D \rightarrow b\ \beta\ D{-}A \in P'$ by schema (II). Thus,

$$D \Rightarrow_{G'} b\ \beta\ D{-}A \Rightarrow_{G'}^{n_2} b\ w_1\ D{-}A \Rightarrow_{G'}^{n_1+n_3} bw_1w_2 = w.$$

To prove (ii), suppose $D \Rightarrow_G^n Bw$. Then we must have

$$\begin{gathered} D \Rightarrow_G^{n_1} A\alpha \Rightarrow_G B\beta\alpha \Rightarrow_G^{n_2+n_3} Bw_1w_2 = w, \\ \beta \Rightarrow_G^{n_2} w_1, \\ \alpha \Rightarrow_G^{n_3} w_2, \end{gathered}$$

[1] As Rosenkrantz and Lewis (1970) note, G' in general has many useless nonterminals. One can modify Definition 1 to avoid creating useless nonterminals (Moore 2000, Johnson and Roark 2000), but we choose not to do so here.

where $n = n_1 + n_2 + n_3 + 1$ and $A \to B\beta \in P$. The induction hypothesis about (i) implies $\beta \Rightarrow_{G'}^{n_2} w_1$ If $n_1 = 0$, then $D = A$, $\alpha = w_2 = \epsilon$, $n_3 = 0$, and schema (III) gives $D{-}B \to \beta \in P'$. Thus,

$$D{-}B \Rightarrow_{G'} \beta \Rightarrow_{G'}^{n_2} w_1 = w.$$

If $n_1 \geq 1$, since $D \Rightarrow_G^{n_1+n_3} A\, w_2$, the induction hypothesis about (ii) gives $D{-}A \Rightarrow_{G'}^{n_1+n_3} w_2$. We have $D{-}B \to \beta\ D{-}A \in P'$ by schema (IV). Thus,

$$D{-}B \Rightarrow_{G'} \beta\ D{-}A \Rightarrow_{G'}^{n_2} w_1\ D{-}A \Rightarrow_{G'}^{n_1+n_3} w_1 w_2 = w.$$

Next we prove by simultaneous induction on $n \geq 1$ that the following hold for every $D, B \in N$ and $w \in T^+$:

(iii) $D \Rightarrow_{G'}^{n} w$ implies $D \Rightarrow_G^n w$;

(iv) $D{-}B \Rightarrow_{G'}^{n} w$ implies $D \Rightarrow_G^n Bw$.

To prove (iii), suppose $D \Rightarrow_{G'}^{n} w$. There are two cases to consider.

Case 1. The first step of the derivation is by a production of type (I). Then

$$\begin{gathered} D \Rightarrow_{G'} b\beta \Rightarrow_{G'}^{n_1} bw_1 = w, \\ \beta \Rightarrow_{G'}^{n_1} w_1, \end{gathered}$$

where $n = n_1 + 1$ and $D \to b\beta \in P$. The induction hypothesis about (iii) implies $\beta \Rightarrow_G^{n_1} w_1$. Thus,

$$D \Rightarrow_G b\beta \Rightarrow_G^{n_1} bw_1 = w.$$

Case 2. The first step of the derivation is by a production of type (II). Then

$$\begin{gathered} D \Rightarrow_{G'} b\ \beta\ D{-}A \Rightarrow_{G'}^{n_1+n_2} bw_1 w_2 = w, \\ \beta \Rightarrow_{G'}^{n_1} w_1, \\ D{-}A \Rightarrow_{G'}^{n_2} w_2, \end{gathered}$$

where $n = n_1 + n_2 + 1$ and $A \to b\beta \in P$. The induction hypothesis about (iii) implies $\beta \Rightarrow_G^{n_1} w_1$. By the induction hypothesis about (iv), $D \Rightarrow_G^{n_2} Aw_2$. Thus,

$$D \Rightarrow_G^{n_2} Aw_2 \Rightarrow_G b\beta w_2 \Rightarrow_G^{n_1} bw_1 w_2 = w.$$

To prove (iv), suppose $D{-}B \Rightarrow_{G'}^{n} w$. Again, there are two cases to consider.

Case 1. The first step of the derivation is by a production of type (III). Then

$$D{-}B \Rightarrow_{G'} \beta \Rightarrow_{G'}^{n_1} w,$$

where $n = n_1 + 1$ and $D \to B\beta \in P$. The induction hypothesis about (iii) implies $\beta \Rightarrow_G^{n_1} w$. Thus,

$$D \Rightarrow_G B\beta \Rightarrow_G^{n_1} Bw.$$

Case 2. The first step of the derivation is by a production of type (IV). Then

$$D{-}B \Rightarrow_{G'} \beta\ D{-}A \Rightarrow_{G'}^{n_1+n_2} w_1w_2 = w,$$
$$\beta \Rightarrow_{G'}^{n_1} w_1,$$
$$D{-}A \Rightarrow_{G'}^{n_2} w_2,$$

where $n = n_1 + n_2 + 1$ and $A \to B\beta \in P$. The induction hypothesis about (iii) implies $\beta \Rightarrow_G^{n_1} w_1$. By the induction hypothesis about (iv), $D \Rightarrow_G^{n_2} Aw_2$. Thus,

$$D \Rightarrow_G^{n_2} Aw_2 \Rightarrow_G B\beta w_2 \Rightarrow_G^{n_1} Bw_1w_2 = Bw.$$

The proposition follows from (i) and (iii). □

One further simple transformation converts G' into what is sometimes called *extended Greibach normal form*. A context-free grammar is in extended Greibach normal form if the right-hand side of each production starts with a terminal. Productions of G' of types (I) and (II) are already of the required form. Consider a production of G' of type (III) or (IV). Since the original context-free grammar G has no unit production, β cannot be empty. If the first symbol of β is a terminal, there is nothing to do. If $\beta = C\gamma$ with $C \in N$, then we replace this production by all productions that we obtain by expanding C with a production of type (I) or (II). This way we obtain a grammar G'' in extended Greibach normal form such that $L(G'') = L(G)$.

3 Second-order ACGs and tree grammars

3.1 ACGs

Our notations and terminology for ACGs follow de Groote (2001) and de Groote and Pogodalla (2004), except that we speak of *higher-order signatures* instead of higher-order *linear* signatures, we write $\to$ instead of $\multimap$, and we use σ (for type substitution) and θ (for term homomorphism) to denote the two components of a lexicon.

Given a finite set of *atomic types*, the set $\mathscr{T}(A)$ of *types* built upon A is the smallest superset of A satisfying the condition

$$\alpha, \beta \in \mathscr{T}(A) \quad \text{implies} \quad (\alpha \to \beta) \in \mathscr{T}(A).$$

We omit the outermost parentheses when we write types. The connective $\to$ is assumed to be right-associative, so we write $\alpha_1 \to \alpha_2 \to \alpha_3$ instead of $\alpha_1 \to (\alpha_2 \to \alpha_3)$.

A *higher-order signature* is a triple $\Sigma = \langle A, C, \tau \rangle$, where A is a finite set of atomic types, C is a finite set of *constants*, and τ is a mapping from C to $\mathscr{T}(A)$. Let X be a countably infinite set of variables. The set $\Lambda(\Sigma)$ of (untyped) *linear λ-terms* built upon a higher-order signature $\Sigma = \langle A, C, \tau \rangle$ is the smallest superset of $X \cup C$ satisfying the following conditions:

1. If $t, u \in \Lambda(\Sigma)$ and t and u do not share any free variables, then $(tu) \in \Lambda(\Sigma)$;

2. If $t \in \Lambda(\Sigma)$ and x is a variable that occurs free in t, then $(\lambda x.t) \in \Lambda(\Sigma)$. ($x$ is no longer free in $\lambda x.t$.)

We omit the outermost parentheses when we write λ-terms. We write $t_1t_2t_3$ for $(t_1t_2)t_3$, $\lambda x.t_1t_2$ for $\lambda x.(t_1t_2)$, and $\lambda x_1 \dots x_n.t$ for $\lambda x_1.(\lambda x_2.\dots.(\lambda x_n.t)\dots)$. We take for granted the notion of $\beta\eta$-equality, and treat $\beta\eta$-equal λ-terms as equal.

Given a higher-order signature $\Sigma = \langle A, C, \tau \rangle$, λ-terms in $\Lambda(\Sigma)$ may be assigned types in the usual way. A *context* is a finite set Γ of typing declarations of the form $x : \alpha$ (where $x \in X, \alpha \in \mathscr{T}(A)$) in which no variable is declared more than once. The following inference system derives *typing judgments* of the form $\Gamma \vdash_\Sigma t : \alpha$, where Γ is a context, $t \in \Lambda(\Sigma)$, and $\alpha \in \mathscr{T}(A)$:

$$\vdash_\Sigma c : \tau(c) \quad \text{for } c \in C \qquad\qquad x : \alpha \vdash_\Sigma x : \alpha \quad \text{for } x \in X \text{ and } \alpha \in \mathscr{T}(A)$$

$$\frac{\Gamma, x : \alpha \vdash_\Sigma t : \beta}{\alpha \vdash_\Sigma \lambda x.t} \qquad\qquad \frac{\Gamma \vdash_\Sigma t : \alpha \to \beta \quad \Delta \vdash_\Sigma u : \alpha}{\Gamma, \Delta \vdash_\Sigma tu : \beta}$$

We write $\Gamma \vdash t : \alpha$ when $\Gamma \vdash_\Sigma t : \alpha$ for some $\Sigma = \langle A, \varnothing, \varnothing \rangle$.

When we write $\Sigma, \Sigma', \Sigma_1$, etc., to denote higher-order signatures, we assume $\Sigma = \langle A, C, \tau \rangle$, $\Sigma' = \langle A', C', \tau' \rangle$, $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$, etc., unless otherwise noted. Given higher-order signatures Σ and Σ', a *lexicon* from Σ to Σ' is a pair $\mathscr{L} = \langle \sigma, \theta \rangle$ such that

1. σ is a type substitution that maps elements of A to elements of $\mathscr{T}(A')$;
2. θ is a mapping from C to $\Lambda(\Sigma')$;
3. $\vdash_{\Sigma'} \theta(c) : \sigma(\tau(c))$ for all $c \in C$.

θ is extended to a mapping from $\Lambda(\Sigma)$ to $\Lambda(\Sigma')$ as follows:

$$\begin{aligned} \theta(x) &= x \quad \text{for } x \in X, \\ \theta(tu) &= \theta(t)\theta(u), \\ \theta(\lambda x.t) &= \lambda x.\theta(t). \end{aligned}$$

When $\mathscr{L} = \langle \sigma, \theta \rangle$ is a lexicon, we write $\mathscr{L}(\alpha)$ and $\mathscr{L}(t)$ for $\sigma(\alpha)$ and $\theta(t)$, respectively.

An *abstract categorial grammar* (ACG) is a quadruple $\mathscr{G} = \langle \Sigma, \Sigma', \mathscr{L}, s \rangle$, where

1. Σ is a higher-order signature called the *abstract vocabulary*;
2. Σ' is a higher-order signature called the *object vocabulary*;
3. $\mathscr{L}$ is a lexicon from Σ to Σ';
4. s is an atomic type of the abstract vocabulary ($s \in A$).

The *abstract language* of $\mathscr{G}$, denoted by $\mathcal{A}(\mathscr{G})$, is defined as follows:

$$\mathcal{A}(\mathscr{G}) = \{\, t \in \Lambda(\Sigma) \mid \vdash_\Sigma t : s \,\}.$$

The *object language* of $\mathscr{G}$, denoted by $\mathcal{O}(\mathscr{G})$, is defined as follows:

$$\mathcal{O}(\mathscr{G}) = \{\, u \in \Lambda(\Sigma') \mid u = \mathscr{L}(t) \text{ for some } t \in \mathcal{A}(\mathscr{G}) \,\}.$$

A constant c of the abstract vocabulary of an ACG $\mathscr{G}$ is *lexical* if at least one constant occurs in $\mathscr{L}(c)$. We say that $\mathscr{G}$ is *lexicalized* if all its abstract constants are lexical.

The *order* of a type α, denoted by $\mathrm{ord}(\alpha)$, is defined as follows:

$$\mathrm{ord}(p) = 1 \quad \text{if } p \text{ is atomic,}$$
$$\mathrm{ord}(\alpha \to \beta) = \max(\mathrm{ord}(\alpha) + 1, \mathrm{ord}(\beta)).$$

The *order* of a higher-order signature Σ is $\max\{\, \mathrm{ord}(\tau(c)) \mid c \in C \,\}$. The *order* of a lexicon $\mathscr{L}$ from Σ to Σ' is $\max\{\, \mathrm{ord}(\mathscr{L}(p)) \mid p \in A \,\}$. $\mathbf{G}(n, m)$ denotes the class of ACGs $\mathscr{G} = \langle \Sigma, \Sigma', \mathscr{L}, s \rangle$ such that the order of Σ is $\leq n$ and the order of $\mathscr{L}$ is $\leq m$. An ACG $\mathscr{G}$ is n-th order if $\mathscr{G} \in \mathbf{G}(n, m)$ for some m.

Note that the set $\Lambda(\Sigma)$ of linear λ-terms built upon a higher-order signature $\Sigma = \langle A, C, \tau \rangle$ does not depend on A and τ. The following lemma is obvious from the definition of the object language, but it will be convenient to be able to refer to it.

Lemma 3. *Let Σ be a higher-order signature. Let Σ_1 and Σ_2 be two higher-order signatures such that $C_1 = C_2$, and let $\theta \colon C \to \Lambda(\Sigma_1) = \Lambda(\Sigma_2)$. Suppose that $\sigma_1 \colon A \to \mathscr{T}(A_1)$ and $\sigma_2 \colon A \to \mathscr{T}(A_2)$ are such that $\mathscr{L}_1 = \langle \sigma_1, \theta \rangle$ is a lexicon from Σ to Σ_1 and $\mathscr{L}_2 = \langle \sigma_2, \theta \rangle$ is a lexicon from Σ to Σ_2. Let $s \in A$ and define*

$$\mathscr{G}_1 = \langle \Sigma, \Sigma_1, \mathscr{L}_1, s \rangle,$$
$$\mathscr{G}_2 = \langle \Sigma, \Sigma_2, \mathscr{L}_2, s \rangle.$$

Then $\mathcal{O}(\mathscr{G}_1) = \mathcal{O}(\mathscr{G}_2)$.

3.2 From second-order ACGs to regular tree grammars

A *ranked alphabet* is a pair $\langle F, \rho \rangle$, where F is an alphabet and ρ is a mapping $\rho \colon F \to \mathbb{N}$. We often write F for the ranked alphabet $\langle F, \rho \rangle$, suppressing reference to ρ. The set of trees over a ranked alphabet F is denoted $\mathbb{T}_F$. We write trees in parenthesis-free prefix notation so that $\mathbb{T}_F \subseteq F^+$.

Let $\Sigma = \langle A, C, \tau \rangle$ be a second-order signature. We call a constant $c \in C$ k-ary if $\tau(c)$ is of the form $q_1 \to \cdots \to q_k \to p$, where $q_1, \ldots, q_k, p$ are atomic types. We define the ranked alphabet $\langle C, \rho_\Sigma \rangle$ by letting $\rho_\Sigma(c) = k$ if c is k-ary. If t is a λ-term in $\Lambda(\Sigma)$ such that $\vdash_\Sigma t : p$ for some $p \in A$, then t can be identified with a member of $\mathbb{T}_{\langle C, \rho_\Sigma \rangle}$. We say that Σ is *compatible* with a ranked alphabet $\langle F, \rho \rangle$ if $F = C$ and $\rho = \rho_\Sigma$.

Any ranked alphabet F can be identified with a second-order signature $\Sigma_F = \langle \{o\}, F, \tau_F \rangle$, where for each $f \in F$, $\tau_F(f) = o^k \to o$ if f has rank k. If $\Sigma = \langle A, C, \tau \rangle$ is a second-order signature, there is a unique first-order lexicon $\mathscr{L}_\Sigma$ from Σ to $\Sigma_{\langle C, \rho_\Sigma \rangle}$ such that $\mathscr{L}_\Sigma(c) = c$ for every $c \in C$.

Let $F = F_1 \cup F_2$ be a ranked alphabet ($F_1 \cap F_2 = \varnothing$). We can write any $t \in \mathbb{T}_F$ uniquely in the following form:

$$w_0 a_1 w_1 \ldots a_k w_k,$$

where $a_i \in F_1$ for $1 \leq i \leq k$ and $w_i \in F_2^*$ for $0 \leq i \leq k$. We let $t|F_1$ denote the string $a_1 \ldots a_k$, and write $t_{y_1, \ldots, y_k}$ for the linear λ-term t' in $\Lambda(\Sigma_{F_2})$ such that $\mathrm{FV}(t') = \{y_1, \ldots, y_k\}$ and $t'[a_1/y_1, \ldots, a_k/y_k] = t$.

A *regular tree grammar* is a context-free grammar such that the nonterminal and terminal alphabets N and T are ranked, all nonterminals have rank 0, and the right-hand side of every production is a tree over $N \cup T$. If G is a regular tree grammar and p is a nonterminal of G, then $\{\, \alpha \in (N \cup T)^* \mid p \Rightarrow_G^* \alpha \,\}$ is a subset of $\mathbb{T}_{N \cup T}$. A set $L \subseteq \mathbb{T}_T$ is a *regular tree language* if $L = L(G)$ for some regular tree grammar G.

It is easy to see that the abstract language of a second-order ACG is a regular tree language.

Definition 4. Let $\mathscr{G} = \langle \Sigma, \Sigma_{\text{obj}}, \mathscr{L}, s \rangle$ be a second-order ACG, with $\Sigma = \langle A, C, \tau \rangle$. The regular tree grammar $G_{\mathscr{G}}$ determined by $\mathscr{G}$ is

$$G_{\mathscr{G}} = \langle A, C, P_{\mathscr{G}}, s \rangle,$$

where the rank for $c \in C$ is given by $\rho_\Sigma(c)$ and

$$P_{\mathscr{G}} = \{\, p \to c q_1 \dots q_k \mid c \in C \wedge \tau(c) = q_1 \to \dots \to q_k \to p \,\}.$$

We can easily prove the following lemma:

Lemma 5. *Let $\mathscr{G} = \langle \Sigma, \Sigma_{obj}, \mathscr{L}, s \rangle$ be a second-order ACG with $\Sigma = \langle A, C, \tau \rangle$, and let $G_{\mathscr{G}}$ be the regular tree grammar determined by $\mathscr{G}$. For every $p \in A$ and $t \in \mathbb{T}_{A \cup C}$,*

$$p \Rightarrow_{G_{\mathscr{G}}}^* t \quad \textit{if and only if} \quad y_1 : q_1, \dots, y_k : q_k \vdash_\Sigma t_{y_1,\dots,y_k} : p,$$

where $t|A = q_1 \dots q_k$.

Lemma 5 implies

Proposition 6. *For every second-order ACG $\mathscr{G}$, it holds that $L(G_{\mathscr{G}}) = \mathcal{A}(\mathscr{G})$.*

3.3 From linear context-free tree grammars to second-order ACGs

We let $X_n = \{x_1, \dots, x_n\}$ be a ranked alphabet of n variables, all of rank 0. Given a ranked alphabet F, an *n-context* over F is a tree $t[x_1, \dots, x_n]$ in $\mathbb{T}_{F \cup X_n}$ in which each x_i $(1 \leq i \leq n)$ occurs exactly once. If $u_1, \dots, u_n \in \mathbb{T}_F$, $t[u_1, \dots, u_n]$ denotes the result of substituting $u_1, \dots, u_n$ for $x_1, \dots, x_n$ in $t[x_1, \dots, x_n]$. The set of n-contexts over F is denoted by $\mathbb{C}_F(n)$. Note that $\mathbb{C}_F(0) = \mathbb{T}_F$.

Since a context $t[x_1, \dots, x_n] \in \mathbb{C}_{F_1 \cup F_2}(n)$ is a special kind of tree in $\mathbb{T}_{F_1 \cup F_2 \cup X_n}$, the vertical bar notation extends to contexts. If

$$t[x_1, \dots, x_n] = w_0 a_1 w_1 \dots a_k w_k,$$

where $a_i \in F_1$ for $1 \leq i \leq k$ and $w_i \in (F_2 \cup X_n)^*$ for $0 \leq i \leq k$, we let $t[x_1, \dots, x_n]|F_1$ denote the string $a_1 \dots a_n$, and write $t[x_1, \dots, x_n]_{y_1,\dots,y_k}$ for the linear λ-term t' in $\Lambda(\Sigma_{F_2})$ such that $\mathrm{FV}(t') = \{x_1, \dots, x_n, y_1, \dots, y_k\}$ and $t'[q_1/y_1, \dots, q_k/y_k] = t$.

A *linear context-free tree grammar*[2] is a quadruple $G = \langle N, T, P, s \rangle$, where N is a ranked alphabet of nonterminals, T is a ranked alphabet of

[2]We follow de Groote and Pogodalla (2004) in deviating from standard terminology and mean *linear and non-deleting* by 'linear'.

terminals, s is a nonterminal of rank 0, and P is a finite set of productions of the form

$$ax_1 \dots x_n \to t[x_1, \dots, x_n]$$

where a is a member of N of rank n and $t[x_1, \dots, x_n]$ is an n-context over $N \cup T$.

For every $u, v \in \mathbb{T}_{N \cup T}$, $u \Rightarrow_G v$ is defined to hold if and only if there is a $c[x_1] \in \mathbb{C}_{N \cup T}(1)$ and a production $ax_1 \dots x_n \to t[x_1, \dots, x_n]$ in P such that

$$\begin{aligned} u &= c[au_1 \dots u_n], \\ v &= c[t[u_1, \dots, u_n]]. \end{aligned}$$

The language $L(G)$ of a linear context-free tree grammar G is defined by $L(G) = \{\, t \in \mathbb{T}_T \mid s \Rightarrow_G^* t \,\}$.

Note that a regular tree grammar $G = \langle N, T, P, s \rangle$ is just a special kind of linear context-free tree grammar such that all elements of N have rank 0.

Let $G = \langle N, T, P, s \rangle$ be a linear context-free tree grammar. We use the method of de Groote and Pogodalla (2004) to construct a second-order ACG $\mathscr{G}_G$ such that $\mathcal{O}(\mathscr{G}_G) = L(G)$.

Definition 7. Let $G = \langle N, T, P, s \rangle$ be a linear context-free tree grammar.

1. Define a second-order signature $\Sigma_G = \langle A_G, C_G, \tau_G \rangle$ as follows:

$$\begin{aligned} A_G &= N, \\ C_G &= \{\, c_\pi \mid \pi \in P \,\}, \\ \tau_G(c_{ax_1 \dots x_n \to t[x_1, \dots, x_n]}) &= a_1 \to \dots \to a_k \to a, \\ &\qquad \text{where } t[x_1, \dots, x_n] | N = a_1 \dots a_k. \end{aligned}$$

2. Define a second-order lexicon $\mathscr{L}_G = \langle \sigma_G, \theta_G \rangle$ from Σ_G to Σ_T by

$$\begin{aligned} \sigma_G(a) &= o^n \to o \quad \text{if } a \text{ has rank } n, \\ \theta_G(c_{ax_1 \dots x_n \to t[x_1, \dots, x_n]}) &= \lambda y_1 \dots y_k x_1 \dots x_n . t[x_1, \dots, x_n]_{y_1, \dots, y_k}, \\ &\qquad \text{where } t[x_1, \dots, x_n] | N = a_1 \dots a_k. \end{aligned}$$

3. Define a second-order ACG $\mathscr{G}_G$ by

$$\mathscr{G}_G = \langle \Sigma_G, \Sigma_T, \mathscr{L}_G, s \rangle.$$

Note that $\mathscr{G}_G \in \mathbf{G}(2, 2)$, and in the special case where G is a regular tree grammar, $\mathscr{G}_G \in \mathbf{G}(2, 1)$.

Proposition 8. *For every linear context-free tree grammar G, it holds that $\mathcal{O}(\mathscr{G}_G) = L(G)$.*

Proof. It suffices to prove the following claim:

Claim. *For every $t \in \mathbb{T}_{N \cup T}$, the following are equivalent:*

(i) $s \Rightarrow_G^* t$.

(ii) *There is a $u \in \Lambda(\Sigma_G)$ such that*

$$y_1 : a_1, \dots, y_k : a_k \vdash_{\Sigma_G} u : s \quad \textit{and} \quad \mathscr{L}_G(u)[a_1/y_1, \dots, a_k/y_k] = t.$$

First we prove (i) $\Rightarrow$ (ii) by induction on n such that $s \Rightarrow_G^n t$. If $n = 0$, then $t = s$. Let $u = y_1$. We have $y_1 : s \vdash_{\Sigma_G} u : s$ and $\mathscr{L}_G(u)[s/y_1] = y_1[s/y_1] = s$, so (ii) holds. If $n = n_1 + 1$, then $s \Rightarrow_G^{n_1} c[bt_1 \dots t_m] \Rightarrow_G c[t_0[t_1, \dots, t_m]] = t$ for some context $c[x_1] \in \mathbb{C}_{N \cup T}$ and some production $\pi = bx_1 \dots x_m \to t_0[x_1, \dots, x_m]$ in P. By the induction hypothesis, there is a $u' \in \Lambda(\Sigma_G)$ such that

$$y_1 : a_1, \dots, y_l : a_l, z : b \vdash_{\Sigma_G} u' : s \tag{1}$$

and

$$\mathscr{L}_G(u')[a_1/y_1, \dots, a_l/y_l, b/z] = c[bt_1 \dots t_m]. \tag{2}$$

Let $t_0[x_1, \dots, x_m]|N = b_1 \dots b_j$. Since $\tau_G(\pi) = b_1 \to \dots \to b_j \to b$, we have

$$z_1 : b_1, \dots, z_j : b_j \vdash_{\Sigma_G} c_\pi z_1 \dots z_j : b. \tag{3}$$

By the definition of $\mathscr{L}_G$,

$$\mathscr{L}_G(c_\pi) = \lambda z_1 \dots z_j x_1 \dots x_m . t_0[x_1, \dots, x_m]_{z_1, \dots, z_j}. \tag{4}$$

By (1) and (3),

$$y_1 : a_1, \dots, y_l : a_l, z_1 : b_1, \dots, z_j : b_j \vdash_{\Sigma_G} u'[c_\pi z_1 \dots z_j/z] : s.$$

By (2) and (4),

$$\begin{aligned}
&\mathscr{L}_G(u'[c_\pi z_1 \dots z_j/z])[a_1/y_1, \dots, a_l/y_l, b_1/z_1, \dots, b_j/z_j] \\
&= \mathscr{L}_G(u')[\mathscr{L}_G(c_\pi) z_1 \dots z_j/z][a_1/y_1, \dots, a_l/y_l, b_1/z_1, \dots, b_j/z_j] \\
&= \mathscr{L}_G(u')[a_1/y_1, \dots, a_l/y_l][\mathscr{L}_G(c_\pi) b_1 \dots b_j/z] \\
&= c[zt_1 \dots t_m][(\lambda z_1 \dots z_j x_1 \dots x_m . t_0[x_1, \dots, x_m]_{z_1, \dots, z_j}) b_1 \dots b_j/z] \\
&= c[zt_1 \dots t_m][\lambda x_1 \dots x_m . t_0[x_1, \dots, x_m]/z] \\
&= c[t_0[t_1, \dots, t_m]] \\
&= t.
\end{aligned}$$

Therefore, (ii) holds with $u = u'[c_\pi z_1 \dots z_j]$.

Next we prove (ii) $\Rightarrow$ (i) by induction on the number n of occurrences of constants in u. If $n = 0$, then $k = 1$, $u = y_1$, and $s = a_1$. Since $\mathscr{L}_G(u) = y_1$, it follows that $t = s$ and (i) obviously holds. If $n \geq 1$, then there must be a context $c[x_1] \in \mathbb{C}_{C_G}(1)$ and some production $\pi = bx_1 \dots x_m \to t_0[x_1, \dots, x_m]$ such that

$$u = c[c_\pi y_{i_1} \dots y_{i_l}], \tag{5}$$

$$t_0[x_1, \dots, x_m]|N = a_{i_1} \dots a_{i_l} \tag{6}$$

for some $i_1, \dots, i_l \in \{1, \dots, k\}$. Let $\{j_1, \dots, j_{k-l}\} = \{1, \dots, k\} - \{i_1, \dots, i_l\}$. Since $y_1 : a_1, \dots, y_k : a_k \vdash_{\Sigma_G} u : s$ and $y_{i_1} : a_{i_1}, \dots, y_{i_l} : a_{i_l} \vdash_{\Sigma_G} c_\pi y_{i_1} \dots y_{i_l} : b$, we must have

$$y_{j_1} : a_{j_1}, \dots, y_{j_{k-l}} : a_{j_{k-l}}, z : b \vdash c[z] : s.$$

Let

$$t' = \mathscr{L}_G(c[z])[a_{j_1}/y_{j_1}, \dots, a_{j_{k-l}}/y_{j_{k-l}}, b/z]. \tag{7}$$

By the induction hypothesis,

$$s \Rightarrow_G^* t'. \tag{8}$$

Since b has rank m,

$$t' = c'[bt_1 \dots t_m] \tag{9}$$

for some $c' \in C_{N \cup T}(1)$ and $t_1, \dots, t_m \in \mathbb{T}_{N \cup T}$. Let

$$t = \mathscr{L}_G(u)[a_1/y_1, \dots, a_k/y_k].$$

Then

$$\begin{aligned}
t &= \mathscr{L}_G(c[c_\pi y_{i_1} \dots y_{i_l}])[a_1/y_1, \dots, a_k/y_k] \quad \text{by (5)} \\
&= \mathscr{L}_G(c[z])[\mathscr{L}_G(c_\pi) y_{i_1} \dots y_{i_l}/z][a_1/y_1, \dots, a_k/y_k] \\
&= \mathscr{L}_G(c[z])[\lambda x_1 \dots x_m.t_0[x_1, \dots, x_m]_{y_{i_1}, \dots, y_{i_l}}/z][a_1/y_1, \dots, a_k/y_k] \\
&= \mathscr{L}_G(c[z])[a_{j_1}/y_{j_1}, \dots, a_{j_{k-l}}/y_{j_{k-l}}] \\
&\qquad [\lambda x_1 \dots x_m.t_0[x_1, \dots, x_m]_{y_{i_1}, \dots, y_{i_l}}[a_{i_1}/y_{i_1}, \dots, a_{i_l}/y_{i_l}]/z] \\
&= \mathscr{L}_G(c[z])[a_{j_1}/y_{j_1}, \dots, a_{j_{k-l}}/y_{j_{k-l}}][\lambda x_1 \dots x_m.t_0[x_1, \dots, x_m]/z] \quad \text{by (6)} \\
&= c'[zt_1 \dots t_m][\lambda x_1 \dots x_m.t_0[x_1, \dots, x_m]/z] \quad \text{by (7) and (9)} \\
&= c'[t_0[t_1, \dots, t_m]].
\end{aligned}$$

So $t' \Rightarrow_G t$. From this and (8), we conclude (i). □

More generally, $L(G)$ may be generated by a second-order ACG whose object vocabulary is compatible with T.

Corollary 9. *Let $G = \langle N, T, P, s \rangle$ be a linear context-free tree grammar, and let $\Sigma = \langle A, T, \tau \rangle$ be a higher-order signature compatible with T. Suppose that $\sigma \colon N \to \mathscr{T}(A)$ satisfies the following condition:*

(†) *For every production $ax_1 \dots x_n \to t[x_1, \dots, x_n]$ in P,*

$$y_1 : \sigma(a_1), \dots, y_k : \sigma(a_k) \vdash_\Sigma \lambda x_1 \dots x_n.t[x_1, \dots, x_n]_{y_1, \dots, y_k} : \sigma(a)$$

where $t[x_1, \dots, x_n]|N = a_1 \dots a_k$.

Then

$$\mathscr{L} = \langle \sigma, \theta_G \rangle,$$

where θ_G is defined in part 2 of Definition 7, is a lexicon from Σ_G to Σ. Define a second-order ACG $\mathscr{G}$ by

$$\mathscr{G} = \langle \Sigma_G, \Sigma, \mathscr{L}, s \rangle.$$

Then $\mathcal{O}(\mathscr{G}) = L(G)$.

Proof. By Proposition 8 and Lemma 3. □

The following is an important special case of the above construction. Note that when G is a regular tree grammar, the definition of θ_G takes the form

$$\theta_G(c_{a \to t}) = \lambda y_1 \dots y_k.t_{y_1, \dots, y_k}.$$

Corollary 10. *Let $G = \langle N, T, P, s \rangle$ be a regular tree grammar. Suppose that $\tau \colon T \to \mathscr{T}(N)$ is such that $\Sigma = \langle N, T, \tau \rangle$ is a second-order signature compatible with T and the following condition is satisfied:*

(‡) *For every* $a \to t \in P$ *with* $t|N = a_1 \dots a_k$,

$$y_1 : a_1, \dots, y_k : a_k \vdash_\Sigma t_{y_1,\dots,y_k} : a.$$

Then

$$\widetilde{\mathscr{L}_G} = \langle \mathrm{id}_N, \theta_G \rangle,$$

where id_N *is the identity function on* N*, is a first-order lexicon from* Σ_G *to* Σ*. Define a second-order ACG* $\widetilde{\mathscr{G}_G}$ *by*

$$\widetilde{\mathscr{G}_G} = \langle \Sigma_G, \Sigma, \widetilde{\mathscr{L}_G}, s \rangle.$$

Then $\widetilde{\mathscr{G}_G} \in \mathbf{G}(2,1)$ *and* $\mathcal{O}(\widetilde{\mathscr{G}_G}) = L(G)$.

4 Elimination of non-lexical constants

At each step of our procedure, we start with a second-order ACG $\mathscr{G} = \langle \Sigma, \Sigma_{\mathrm{obj}}, \mathscr{L}, s \rangle$ with $\Sigma = \langle A, C, \tau \rangle$. We assume $C = D \cup E$, where D is the set of lexical constants and E is the set of non-lexical constants. We let $d, d_1, d_2, \dots$ stand for elements of D, and $e, e_1, e_2, \dots$ for elements of E.

In Section 4.1, we make no further assumption on $\mathscr{G}$ and produce a second-order ACG $\mathscr{G}_1$ without nullary non-lexical constants such that $\mathcal{O}(\mathscr{G}_1)$ equals $\mathcal{O}(\mathscr{G})$ minus combinators. In Section 4.2, we assume that $\mathscr{G}$ has no nullary non-lexical constant and produce a second-order ACG $\mathscr{G}_2$ without nullary or unary non-lexical constants such that $\mathcal{O}(\mathscr{G}_2) = \mathcal{O}(\mathscr{G})$. Finally, in Section 4.3, we assume that $\mathscr{G}$ has no nullary or unary non-lexical constant and produce a second-order ACG $\mathscr{G}_3$ without non-lexical constants such that $\mathcal{O}(\mathscr{G}_3) = \mathcal{O}(\mathscr{G})$. Combining the three steps, we obtain a procedure for lexicalizing an arbitrary second-order ACG. We will see that the order of the lexicon stays the same in the first two steps, but increases by one in the last step.

Theorem 1. *Given an ACG* $\mathscr{G} \in \mathbf{G}(2, m)$*, one can effectively find a lexicalized ACG* $\mathscr{G}' \in \mathbf{G}(2, m{+}1)$ *such that* $\mathcal{O}(\mathscr{G}')$ *equals* $\mathcal{O}(\mathscr{G})$ *minus combinators.*

4.1 Elimination of nullary non-lexical constants

To eliminate nullary non-lexical constants from $\mathscr{G}$, we eliminate from $G_{\mathscr{G}}$ all productions of the form $p \to e$, obtaining a regular tree grammar G_1 such that the right-hand side of every production contains some $d \in D$ or some $q \in A$. Clearly, this is in general impossible without affecting the generated language, since there may be infinitely many trees $t \in \mathbb{T}_E$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t$. However, what we are after is an ACG whose object language is equal to the object language of the original ACG minus combinators, and fortunately this allows us to discard from $L(G_{\mathscr{G}})$ all trees which contain as a subtree some tree in $\mathbb{T}_E$ whose height exceeds a certain bound. Our method of doing this is loosely based on the standard method of eliminating ϵ-productions from context-free grammars.

Let

$$\begin{aligned} \Gamma_0 &= \{\, \langle p, \mathscr{L}(e) \rangle \mid e \in E \wedge p \to e \in P_{\mathscr{G}} \,\}, \\ \Gamma_{h+1} &= \Gamma_h \cup \{\, \langle p, \mathscr{L}(e) u_1 \dots u_k \rangle \mid e \in E \wedge p \to e q_1 \dots q_k \in P_{\mathscr{G}} \wedge \\ &\qquad \langle q_i, u_i \rangle \in \Gamma_h \text{ for } 1 \le i \le k \,\}. \end{aligned}$$

Γ_h contains $\langle p, u\rangle$ if and only if there is a tree $t \in \mathbb{T}_E$ of height $\leq h$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t$ and $u = \mathscr{L}(t)$. Then if we let $\Gamma = \bigcup_{h\in\mathbb{N}} \Gamma_h$,

$$\Gamma = \{\, \langle p, \mathscr{L}(t)\rangle \mid p \in A \wedge t \in \mathbb{T}_E \wedge p \Rightarrow^*_{G_{\mathscr{G}}} t \,\}.$$

By Lemma 5, $\langle p, u\rangle \in \Gamma$ implies $\vdash u : \mathscr{L}(p)$. Since $\{\, u \mid \vdash u : \mathscr{L}(p) \,\}$ is finite, it follows that Γ is finite. Therefore, there must be a number h such that $\Gamma_h = \Gamma_{h+1} = \Gamma$. Let h be the least such number.

Now let

$$\mathbb{T}_E^{p,h} = \{\, t \in \mathbb{T}_E \mid p \Rightarrow^*_{G_{\mathscr{G}}} t \wedge t \text{ has height} \leq h \,\},$$

and define

$$\begin{aligned} P_1 = \{\, p \to ct_1 \dots t_k \mid\ & p \to cq_1 \dots q_k \in P_{\mathscr{G}} \wedge I \subseteq \{1, \dots, k\} \wedge \\ & t_i \in \mathbb{T}_E^{q_i,h} \text{ if } i \in I \wedge t_i = q_i \text{ if } i \notin I \wedge \\ & I \neq \{1, \dots, k\} \text{ if } c \in E \,\}, \\ G_1 = \langle C, A, s, P_1\rangle. \end{aligned}$$

Clearly, P_1 does not contain any productions of the form $p \to t$ for $t \in \mathbb{T}_E$.

Lemma 11. (i) $L(G_1) \subseteq L(G_{\mathscr{G}}) - \mathbb{T}_E$.

(ii) $\{\, \mathscr{L}(t) \mid t \in L(G_1) \,\} = \{\, \mathscr{L}(t) \mid t \in L(G_{\mathscr{G}}) - \mathbb{T}_E \,\}$.

Proof. Since P_1 does not contain any productions $p \to t$ with $t \in \mathbb{T}_E$, we see that $L(G_1) \cap \mathbb{T}_E = \varnothing$. From the definition of P_1, $p \to t \in P_1$ implies $p \Rightarrow^*_G t$. Therefore, $s \Rightarrow^*_{G_1} t$ implies $s \Rightarrow^*_{G_{\mathscr{G}}} t$, which means $L(G_1) \subseteq L(G_{\mathscr{G}})$. We have proved (i).

The $\subseteq$ direction of (ii) follows from (i), so it remains to prove the $\supseteq$ direction. This follows from the following claim:

Claim. *If $p \Rightarrow^*_{G_{\mathscr{G}}} t \in \mathbb{T}_C - \mathbb{T}_E$, then there is a $t' \in \mathbb{T}_C - \mathbb{T}_E$ such that $p \Rightarrow^*_{G_1} t'$ and $\mathscr{L}(t) = \mathscr{L}(t')$.*

We prove the claim by induction on $t \in \mathbb{T}_C - \mathbb{T}_E$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t$. If $t = ct_1 \dots t_k$, then there must be a production $p \to cq_1 \dots q_k$ in $P_{\mathscr{G}}$ and $q_i \Rightarrow^*_{G_{\mathscr{G}}} t_i$ $(1 \leq i \leq k)$. Let $I = \{\, i \mid 1 \leq i \leq k \wedge t_i \in \mathbb{T}_E \,\}$. By induction hypothesis, for every $i \in \{1, \dots, k\} - I$, there is a t'_i such that $q_i \Rightarrow^*_{G_1} t'_i$ and $\mathscr{L}(t'_i) = \mathscr{L}(t_i)$. By the definition of h, for every $i \in I$, there is a $t'_i \in \mathbb{T}_E^{q_i,h}$ such that $\mathscr{L}(t'_i) = \mathscr{L}(t_i)$. Since $t \notin \mathbb{T}_E$, if $c \in E$, $I \neq \{1, \dots, k\}$. By the definition of P_1,

$$p \to ct''_1 \dots t''_k \in P_1,$$

where

$$t''_i = \begin{cases} t'_i & \text{if } i \in I, \\ q_i & \text{otherwise.} \end{cases}$$

Therefore, $p \Rightarrow_{G_1} ct''_1 \dots t''_k \Rightarrow^*_{G_1} ct'_1 \dots t'_k$. We have

$$\begin{aligned} \mathscr{L}(ct'_1 \dots t'_k) &= \mathscr{L}(c)\mathscr{L}(t'_1) \dots \mathscr{L}(t'_k) \\ &= \mathscr{L}(c)\mathscr{L}(t_1) \dots \mathscr{L}(t_k) \\ &= \mathscr{L}(ct_1 \dots t_k) \\ &= \mathscr{L}(t). \end{aligned}$$

□

By the definition of P_1 and Lemma 5, condition (‡) holds of Σ and P_1. Hence, the ACG $\widetilde{\mathscr{G}_{G_1}}$ defined by Definition 7 satisfies

$$\mathcal{O}(\widetilde{\mathscr{G}_{G_1}}) = L(G_1) \tag{10}$$

by Corollary 10. By the definition of $\widetilde{\mathscr{L}_{G_1}}$, if an abstract constant $c_{p\to t}$ of $\widetilde{\mathscr{G}_{G_1}}$ is nullary, some $d \in D$ must occur in $\widetilde{\mathscr{L}_{G_1}}(c_{p\to t})$. We compose the original lexicon $\mathscr{L}$ with $\widetilde{\mathscr{L}_{G_1}}$ and obtain

$$\mathscr{G}_1 = \langle \Sigma_{G_1}, \Sigma_{\mathrm{obj}}, \mathscr{L} \circ \widetilde{\mathscr{L}_{G_1}}, s \rangle.$$

Then $\mathscr{G}_1$ has no nullary non-lexical constant. By Proposition 6, part (ii) of Lemma 11, and (10), we get

$$\mathcal{O}(\mathscr{G}_1) = \mathcal{O}(\mathscr{G}) - \{\, u \mid \vdash u : \mathscr{L}(s) \,\}.$$

Since $\widetilde{\mathscr{G}_{G_1}} \in \mathbf{G}(2,1)$, if $\mathscr{G} \in \mathbf{G}(2,m)$, then $\mathscr{G}_1 \in \mathbf{G}(2,m)$ as well.

4.2 Elimination of unary non-lexical constants

We now work with a second-order ACG $\mathscr{G}$ that has no nullary non-lexical constant. To eliminate unary non-lexical constants from $\mathscr{G}$, we eliminate from $G_{\mathscr{G}}$ all productions of the form $p \to eq$, obtaining a regular tree grammar G_2 such that the right-hand side of every production contains some $d \in D$ or at least two nonterminals $q_1, q_2 \in A$. Our method of doing this is loosely based on the standard method of eliminating unit productions from context-free grammars, and the necessary modification is parallel to what we have done in the previous subsection.

Note first that, since every $e \in E$ has rank ≥ 1, every element of $\mathbb{C}_E(1)$ is of the form $e_1 \dots e_n x_1$ where $e_1, \dots, e_n$ all have rank 1. Let

$$\begin{aligned}\Delta_0 &= \{\, \langle p, p, x_1 \rangle \mid p \in A \,\},\\ \Delta_{h+1} &= \Delta_h \cup \{\, \langle p, q, \mathscr{L}(e)u \rangle \mid e \in E \wedge p \to eq_1 \in P_{\mathscr{G}} \wedge \langle q_1, q, u \rangle \in \Delta_h \,\}.\end{aligned}$$

Δ_h contains $\langle p, q, u \rangle$ if and only if there is a context $t[x_1] \in \mathbb{C}_E(1)$ of height $\leq h$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t[q]$ and $u = \mathscr{L}(t[x_1])$. Then if we let $\Delta = \bigcup_{h\in\mathbb{N}} \Delta_h$,

$$\Delta = \{\, \langle p, q, \mathscr{L}(t[x_1]) \rangle \mid p, q \in A \wedge t[x_1] \in \mathbb{C}_E(1) \wedge p \Rightarrow_{G_{\mathscr{G}}} t[q] \,\}.$$

By Lemma 5, $\langle p, q, u \rangle \in \Delta$ implies $x_1{:}\mathscr{L}(q) \vdash u{:}\mathscr{L}(p)$. Since $\{\, u \mid x_1{:}\mathscr{L}(q) \vdash u : \mathscr{L}(p) \,\}$ is finite, it follows that Δ is finite. Therefore, there must be a number h such that $\Delta_h = \Delta_{h+1} = \Delta$. Let h be the least such number.

Now let

$$\mathbb{C}_E(1)^{p,q,h} = \{\, t[x_1] \in \mathbb{C}_E(1) \mid p \Rightarrow^*_{G_{\mathscr{G}}} t[q] \wedge t[x_1] \text{ has height} \leq h \,\},$$

and define

$$\begin{aligned}P_2 = \{\, p \to t[cq_1 \dots q_k] \mid (c \in D \vee k \geq 2) \wedge q \to cq_1 \dots q_k \in P_{\mathscr{G}} \wedge \\ t[x_1] \in \mathbb{C}_E(1)^{p,q,h} \,\},\end{aligned}$$

$$G_2 = \langle C, A, s, P_2 \rangle.$$

Clearly, P_2 does not contain any productions of the form $p \to t$ ($t \in \mathbb{T}_E$) or $p \to t[q]$ ($t[x_1] \in \mathbb{C}_E(1)$).

Lemma 12. (i) $L(G_2) \subseteq L(G_{\mathscr{G}})$.

(ii) $\{\, \mathscr{L}(t) \mid t \in L(G_2) \,\} = \{\, \mathscr{L}(t) \mid t \in L(G_{\mathscr{G}}) \,\}$.

Proof. By the definition of P_2, $p \to t \in P_2$ implies $p \Rightarrow^*_{G_{\mathscr{G}}} t$. Therefore, $s \Rightarrow^*_{G_2} t$ implies $s \Rightarrow^*_{G_{\mathscr{G}}} t$, which means $L(G_2) \subseteq L(G_{\mathscr{G}})$. We have proved (i).

The $\subseteq$ direction of (ii) follows from (i), so it remains to prove the $\supseteq$ direction. This follows from the following claim:

Claim. *If $p \Rightarrow^*_{G_{\mathscr{G}}} t \in \mathbb{T}_C$, then there is a $t' \in \mathbb{T}_C$ such that $p \Rightarrow^*_{G_2} t'$ and $\mathscr{L}(t) = \mathscr{L}(t')$.*

We prove the claim by induction on $t \in \mathbb{T}_C$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t$. We can write t uniquely as $t = t_0[ct_1 \dots t_k]$, where $t_0[x_1] \in \mathbb{C}_E(1)$ and either $c \in E$ or $k \geq 2$. Since $p \Rightarrow^*_{G_{\mathscr{G}}} t$, there must be $q, q_1, \dots, q_k \in A$ such that $p \Rightarrow^*_{G_{\mathscr{G}}} t_0[q]$, $q \to cq_1 \dots q_k \in P_{\mathscr{G}}$, and $q_i \Rightarrow^*_{G_{\mathscr{G}}} t_i$ for $1 \leq i \leq k$. By induction hypothesis, there are t'_i such that $q_i \Rightarrow^*_{G_2} t'_i$ and $\mathscr{L}(t_i) = \mathscr{L}(t'_i)$ for $1 \leq i \leq k$. By the definition of h, there must be a $t'_0[x_1] \in \mathbb{C}_E(1)^{p,q,h}$ such that $\mathscr{L}(t_0[x_1]) = \mathscr{L}(t'_0[x_1])$. By the definition of P_2, then, $p \to t'_0[cq_1 \dots q_k] \in P_2$. Therefore, $p \Rightarrow^*_{G_2} t'_0[cq_1 \dots q_k] \Rightarrow^*_{G_2} t'_0[ct'_1 \dots t'_k]$ and

$$\begin{aligned}
\mathscr{L}(t'_0[ct'_1 \dots t'_k]) &= \mathscr{L}(t'_0[x_1])[\mathscr{L}(c)\mathscr{L}(t'_1) \dots \mathscr{L}(t'_k)/x_1] \\
&= \mathscr{L}(t_0[x_1])[\mathscr{L}(c)\mathscr{L}(t_1) \dots \mathscr{L}(t_k)/x_1] \\
&= \mathscr{L}(t_0[ct_1 \dots t_k]) \\
&= \mathscr{L}(t).
\end{aligned}$$

□

By the definition of P_2 and Lemma 5, condition (‡) holds of Σ and P_2. Hence the ACG $\widetilde{\mathscr{G}}_{G_2}$ defined by Definition 7 satisfies

$$\mathcal{O}(\widetilde{\mathscr{G}}_{G_2}) = L(G_2) \tag{11}$$

by Corollary 10. By the definition of $\widetilde{\mathscr{L}_{G_2}}$, if an abstract constant $c_{p \to t}$ of $\widetilde{\mathscr{G}}_{G_2}$ is nullary or unary, some $d \in D$ must occur in $\widetilde{\mathscr{L}_{G_2}}(c_{p \to t})$. We compose the original lexicon $\mathscr{L}$ with $\widetilde{\mathscr{L}_{G_2}}$ and obtain

$$\mathscr{G}_2 = \langle \Sigma_{G_2}, \Sigma_{\text{obj}}, \mathscr{L} \circ \widetilde{\mathscr{L}_{G_2}}, s \rangle.$$

Then $\mathscr{G}_2$ has no nullary or unary non-lexical constant. By Proposition 6, part (ii) of Lemma 12, and (11), we get

$$\mathcal{O}(\mathscr{G}_2) = \mathcal{O}(\mathscr{G}).$$

Since $\widetilde{\mathscr{G}}_{G_2} \in \mathbf{G}(2,1)$, if $\mathscr{G} \in \mathbf{G}(2,m)$, then $\mathscr{G}_2 \in \mathbf{G}(2,m)$ as well.

4.3 Elimination of remaining non-lexical constants

We now work with a second-order ACG $\mathscr{G}$ which has no nullary or unary non-lexical constant. A production of $G_{\mathscr{G}}$ has one of the following forms:

$$p \to dq_1 \dots q_k,$$
$$p \to eq_1 \dots q_{k+2}.$$

We will eliminate all productions of the second type and obtain a grammar G_3 such that the right-hand side of every production contains some $d \in D$. Our method of doing this is based on the left-corner transform of Section 2.

A simple-minded application of the left-corner transform to $G_{\mathscr{G}}$ obviously does not yield a desirable result; for the purpose of our conversion, we have to treat e in the second schema above as if it is ϵ. As it turns out, we can easily overcome this problem by taking the mirror image of the left-corner transform—we apply the *right-corner transform* to $G_{\mathscr{G}}$ and convert it to a grammar G_3 in *reverse extended Greibach normal form.*

We use nonterminals of the form p/q instead of $p{-}q$. For every $w \in C^+$, we will have $p/q \Rightarrow^*_{G_3} w$ if and only if $p \Rightarrow^*_{G_{\mathscr{G}}} wq$. Define $G_3 = \langle A', C, P'', s \rangle$, where

$$A' = A \cup \{\, p/q \mid p, q \in A \,\},$$

and P'' is defined as follows:

- For each production in $P_{\mathscr{G}}$ of the form $p \to d$, P'' will have the production
$$p \to d \tag{I}$$
and the production of the form
$$r \to r/p\ d \tag{II}$$
for each $r \in A$.

- For each production in $P_{\mathscr{G}}$ of the form $p \to dq_1$, P'' will have the production
$$p/q_1 \to d \tag{III}$$
and the production of the form
$$r/q_1 \to r/p\ d \tag{IV}$$
for each $r \in A$.

- For each pair of productions in $P_{\mathscr{G}}$ of the forms $p \to cq_1 \dots q_{k+2}$ and $q_{k+1} \to d$, P'' will have the production
$$p/q_{k+2} \to c\ q_1\ \dots\ q_k\ d \tag{III+I}$$
and the production of the form
$$r/q_{k+2} \to r/p\ c\ q_1\ \dots\ q_k\ d \tag{IV+I}$$
for each $r \in A$.

- For each pair of productions in $P_{\mathscr{G}}$ of the forms $p \to cq_1 \dots q_{k+2}$ and $q \to d$, P'' will have the production
$$p/q_{k+2} \to c\ q_1\ \dots\ q_k\ q_{k+1}/q\ d \tag{III+II}$$
and the production
$$r/q_{k+2} \to r/p\ c\ q_1\ \dots\ q_k\ q_{k+1}/q\ d \tag{IV+II}$$
for each $r \in A$.

The reader can verify that G_3 is indeed the result of applying Definition 1 and the procedure following it in Section 2, except that the left-to-right order of the right-hand side of productions is reversed. Every production in G_3 has some $d \in D$ as the last symbol of its right-hand side. By Proposition 2, we have

Lemma 13. $L(G_3) = L(G_{\mathscr{G}})$.

Notice that G_3 is no longer a regular tree grammar. However, we can turn it into a linear context-free tree grammar G'_3 by assigning nonterminals of the form p/q rank 1, and rewriting productions of the form $p/q \to \alpha$ as $p/q\ x_1 \to \alpha\ x_1$:

$$P''' = \{\, p \to \alpha \mid p \to \alpha \in P'' \,\} \cup \{\, p/q\ x_1 \to \alpha\ x_1 \mid p/q \to \alpha \in P'' \,\},$$
$$G'_3 = \langle A', C, P''', s \rangle.$$

One can easily check that the right-hand side of a production in P''' is either a tree or a 1-context over $A' \cup C$, depending on whether the left-hand side is of the form p or of the form p/q. It is also easy to see that derivations in G_3 and derivations in G'_3 are isomorphic to each other, so we have

Lemma 14. $L(G'_3) = L(G_{\mathscr{G}})$.

All that remains to be done is to translate G'_3 back into a second-order ACG. Let $\Sigma_{G'_3}$ be the second-order signature defined by Definition 7. If we define a type substitution $\sigma\colon A' \to \mathscr{T}(A)$ as follows:

$$\sigma(p) = p \quad \text{for } p \in A,$$
$$\sigma(p/q) = q \to p \quad \text{for } p, q \in A,$$

then condition (†) holds of σ, Σ, and P''', as the reader can check easily. Let θ_{G_3} be as defined by part 2 of Definition 7, i.e.,

$$\theta_{G'_3}(c_{p \to t}) = \lambda y_1 \ldots y_k.t_{y_1,\ldots,y_k}, \quad \text{where } t|A' = q_1 \ldots q_k,$$
$$\theta_{G'_3}(c_{p/q\ x_1 \to t[x_1]}) = \lambda y_1 \ldots y_k x_1.t[x_1]_{y_1,\ldots,y_k} \quad \text{where } t[x_1]|A' = q_1 \ldots q_k.$$

Then

$$\widehat{\mathscr{L}_{G'_3}} = \langle \sigma, \theta_{G'_3} \rangle$$

is a second-order lexicon from $\Sigma_{G'_3}$ to Σ, and the second-order ACG

$$\widehat{\mathscr{G}_{G'_3}} = \langle \Sigma_{G'_3}, \Sigma, \widehat{\mathscr{L}_{G'_3}}, s \rangle$$

satisfies

$$\mathcal{O}(\widehat{\mathscr{G}_{G'_3}}) = L(G'_3) \tag{12}$$

by Corollary 9. By the definition of $\widehat{\mathscr{L}_{G'_3}}$, for every abstract constant c of $\widehat{\mathscr{G}_{G'_3}}$, $\widehat{L}_{G'_3}(c)$ must contain some $d \in D$. We compose the original lexicon $\mathscr{L}$ with $\widehat{\mathscr{L}_{G'_3}}$ and obtain

$$\mathscr{G}_3 = \langle \Sigma_{G'_3}, \Sigma_{\text{obj}}, \mathscr{L} \circ \widehat{\mathscr{L}_{G'_3}}, s \rangle.$$

Then $\mathscr{G}_3$ has no non-lexical constant. By Proposition 6, Lemma 14, and (12), we conclude

$$\mathcal{O}(\mathscr{G}_3) = \mathcal{O}(\mathscr{G}).$$

Since $\widehat{\mathscr{G}_{G'_3}} \in \mathbf{G}(2,2)$, if $\mathscr{G} \in \mathbf{G}(2,m)$, then $\mathscr{G}_3 \in \mathbf{G}(2,m+1)$.

References

de Groote, Philippe. 2001. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155.

de Groote, Philippe and Sylvain Pogodalla. 2004. On the expressive power of abstract categorial grammars: Representing context-free formalisms. *Journal of Logic, Language and Information* **13**, 421–438.

Johnson, Mark and Brian Roark. 2000. Compact non-left-recursive grammars using the selective left-corner transform and factoring. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING), 2000*, pages 355–361.

Moore, Robert C. 2000. Removing left recursion from context-free grammars. In *Proceedings, 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, Washington, pages 249–255.

Rosenkrantz, D.J. and P.M. Lewis II. 1970. Deterministic left corner parser. In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata*, pages 139–152.

Yoshinaka, Ryo and Makoto Kanazawa. 2005. The complexity and generative capacity of lexicalized abstract categorial grammars. In Philippe Blache, Edward Stabler, Joan Busquets, and Richard Moot, editors, *Logical Aspects of Computational Linguistics: 5th International Conference, LACL 2005*, pages 330–346. Berlin: Springer.