UCLA/Potsdam Working Papers in Linguistics, May 2003
Head Movement and Syntactic Theory-Mahajan (ed.)

COMPARING 3 PERSPECTIVES ON HEAD MOVEMENT

Edward Stabler
stabler@ucla.edu

In the attempt to understand the fundamental properties of human language, in virtue of which it can be acquired and used as it is, the most common strategy is to propose a grammar G that provides a reasonable account of some particular structures of expressions in particular languages. For example, there are proposals in this volume about verbal complexes in German, about agreement on nouns in Maasai, about A-binding in English, and so on. On the basis of these hypotheses, we can use poverty-of-stimulus arguments, cross-linguistic comparisons, etc. to support universal claims of the following form:

(U) G is in the (restricted) class of grammars $\mathcal{G}$

More than 5 decades of very active research shows that identifying a restrictive, illuminating, explanatory universal grammar $\mathcal{G}$ is not a trivial matter! There is little consensus about the structure of verbal complexes in German, nouns in Maasai, or A-binding in English. One practical, uncontroversial observation we can make is this: the objects of study, the particular grammars G, are complex, and our representations of these objects are complex too. Consequently, when the investigation is informal, it can be very difficult to separate the substantial, supported empirical claims about G from the consequences of mere notational conventions, and from consequences of assumptions that are merely programmatic.

Another strategy for getting to the universals (U) involves supporting weaker hypotheses than particular (parts of) grammars G. It still rests on hypotheses about properties of particular expressions, of course, but it does not begin with any complete account of them. Rather it identifies more abstract properties that any reasonable grammar should require. In effect, the strategy is to aim for instances of (U) more directly. Instead of attempting to specify any particular grammar G for some particular structures of a particular language, we can try to determine whether the actual grammar G is in some infinite class of grammars. Since the claims here are more abstract, they sometimes require more careful, formal formulations, which may be appropriate in any case given the complexity of most of the prominent proposals in the field.

There are many familiar hypotheses of this latter sort, which can be regarded as instances of (U). Some are well established, while others are still being actively studied. We can order our claims according the complexity of the patterns, the regularities, that can be enforced, from simple finite state languages, out to the sets of morpheme sequences that are not even recursively enumerable. (Remember that a simple counting argument shows that, by far, most sets of morpheme sequences are not recursively enumerable (RE), so even the hypothesis that human languages are RE is very strong). It is not a surprise that the most interesting and controversial claims about the complexity of human languages (here marked with ?) are in the middle of this spectrum:

(1.1) finite state grammars (**FSG**) do not provide a good description of the nested dependencies found in human languages: G$\notin$**FSG** (Chomsky 1956, et al)

(1.2) context free grammars (**CFG**) do not provide a good description of the crossing dependencies found in human languages: G$\notin$ **CFG** (Chomsky 1956, et al)

(?1.3) tree adjoining grammars (**TAG)** – and the similar, weakly equivalent grammars studied by Joshi, Vijay Shanker and Weir 1991 – do not provide a good description of the dependencies in German scrambling: G$\notin$ **TAG** (Rambow 1994, Steedman 2000, Kulick 2000, et al)

(?1.4) the sets of fully grammatical sequences of morphemes in human languages are semilinear (**semiLin**): G$\in$ **SemiLin** (Joshi et al, but cf Michaelis and Kracht 1996)

(1.5) the question of whether a sequence of morphemes is a fully grammatical sentence of a human languages is effectively decidable: G$\in$ **REC** (Putnam 1961, Matthews 1979, et al)

(1.6) the sequences of morphemes is a fully grammatical sentence of a human languages form a recursively enumerable set: G$\in$ **RE**

In the “controversial” region of this spectrum of claims, 3 approaches to head placement have been studied, 3 infinite classes of grammars:

MG: simple grammars with remnant movement,

MGH: simple grammars that allow head movement in certain configurations, and

MTTG: simple grammars for mirror-theory-like structures.

The question that will be considered here is whether these different kinds of grammar have different properties that would bear on the question of whether one of them is appropriate for human language syntax.

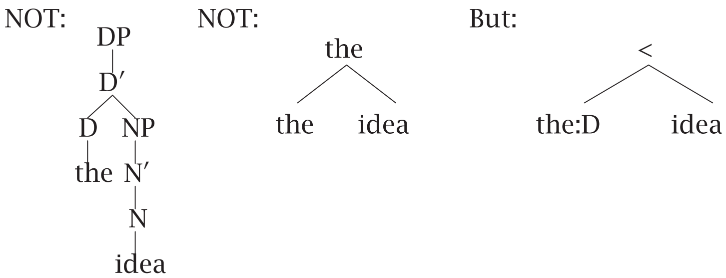
1 MGs: simple “minimalist” grammars

A simple formal model of grammars with remnant movement inspired by Chomsky (1995) and Koopman and Szabolcsi (2000) has been developed in (Stabler, 1999). The essential features are easily summarized. A grammar has two components: (i) a lexicon, and (ii) the structure building rules *merge* and *move* which are universal and invariant, as defined below. Each lexicon is a finite set of lexical items, and each lexical item is a finite sequence of features. There are 4 different kinds of syntactic features, and we use the standard written representations of words as an abbreviation for their phonetic and semantic features. The five kinds of features are the following, and some typical examples have been provided:

category	N,V,A,P,C,D,…
selector	=N,=V,=A,=P,=C,=D,…
licensor	+wh,+case,…
licensee	-wh,-case,…
phonetic+semantic	Romeo,Juliet,love,…

An expression with the feature N is a noun or noun phrase. An expression with the feature =N wants to select a noun phrase NP. An expression with the feature +wh wants to license a whprase in its specifier position, and an expression with the feature -wh must be licensed by moving to such a specifier position.

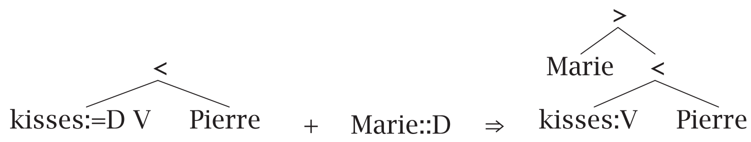
The structure building rules will be feature driven, and so it will be convenient to adopt a “bare” representation of syntactic structure:



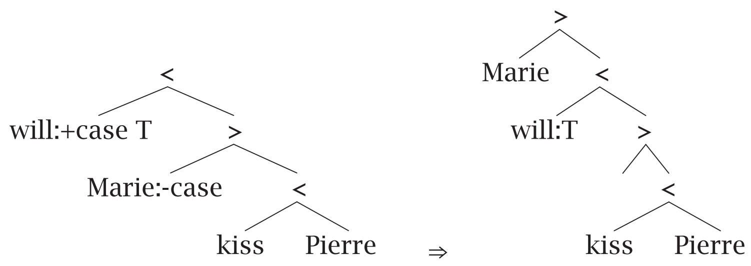
We adopt the “bare” ordered trees shown at the right, where the leaves have features and internal nodes are marked with an order symbol < or > which “points” toward the head of the complex. An expression with just 1 node is its own “head.” So then each leaf is a head with lexical features, and the maximal projection of a leaf is the largest subtree for which that leaf is the head.

It is not difficult to provide fully rigorous definitions of the structure building rules as in (Stabler, 1999), but the operations are quite simple and some examples will suffice to indicate what is intended. The operation *merge* is a function that applies to two trees, where the head of the first begins with a selection feature =f and the head of the second begins with a category feature f. The features =f and f are checked and deleted, but the structure of the result depends on whether the first tree, the selector, is complex or not – that is, whether it has already filled its complement position or not. When the first tree is a simple lexical item with an unfilled complement position, the second tree is attached on the right, in “complement” position:

When the first tree is not simple, the second tree is attached on the left, in “specifier” position: right, in “complement” position:



The operation *move* is unary, applying to a single expression whose head begins with a licensor feature +f, and where exactly one other node in the tree begins with -f. In this case, *move* applies to move the maximal projection of the -f head to the specifier of the +f head, again checking and deleting both features:

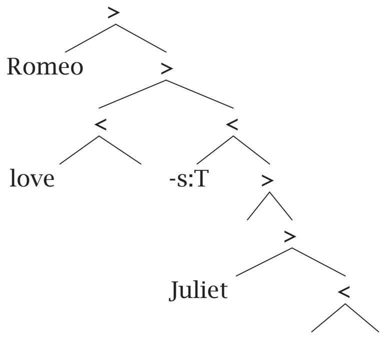


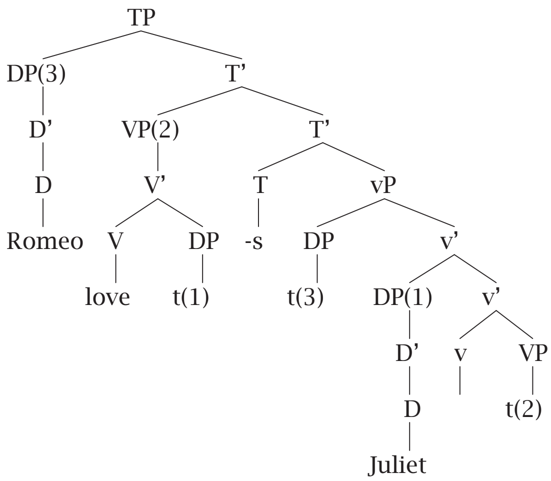
Since *merge, move* are invariant, each MG can be given simply by its finite lexicon. The language is the whole set of structures generated from the lexicon by the application of *merge* and *move*.

For example, consider the following grammar **MG1** with 5 lexical items, inspired by some suggestions of Mahajan (2000).

love::=D V -v	-s::=v +v +case T	ϵ::=V +case =D v
Romeo:: D -case	Juliet:: D -case	

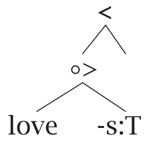
With this grammar we have a 7 step derivation of *Romeo love -s Juliet* (which the reader should be able to calculate by hand), with remnant movement at the penultimate step, when the VP from which the object was extracted moves to the specifier of IP. Here is the derived bare structure, and a more conventional depiction of the same result:





2 MGH: MGs with head movement

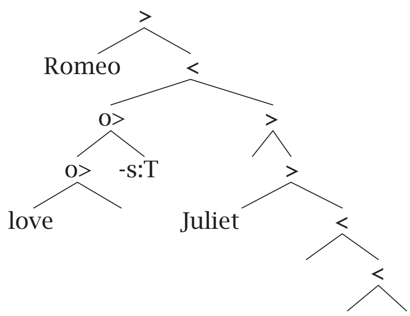
A more conventional treatment of the relation between inflectional affixes and the verbs they attach to involves head movement. A simple formal model of transformational grammar with head movement was proposed by (Stabler, 2001), based on earlier proposals (Stabler, 1997; Cornell, 1997; Cornell, 1998). In this model, head movement occurs only in the configuration of selection, and is triggered by a special selection feature on the affix =>, indicating its affixal status. To indicate the complex heads that result, we use the symbol o>:

-s::=>V T + love::V ⇒ 

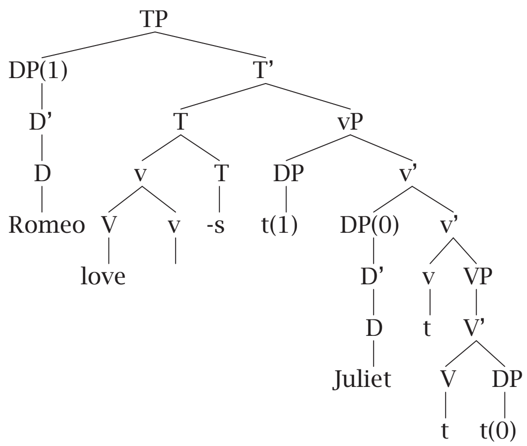
So for example, consider the following grammar **MGH1** with 5 lexical items:

love::=D V	-s::=>v +case T	ϵ::=>V +case =D v
Romeo::D -case	Juliet::D -case	

This grammar has an empty affix v which attracts the verb (when it selects the VP and attaches it in complement position), and the complex v head that results is then attracted to the tense affix *-s* (when the vP is selected and attached in complement position). So it is easy to calculate that in 6 steps we derive the following bare structure:



This bare phrase structure can also be represented in the more conventional form, as shown below:



3 MTTG: mirror theory tree grammars

A rather different mechanism accomplishes the affixation in the "mirror theory" proposed by Brody (2000) and Brody (1999). A simple formal, derivational grammar inspired by mirror theory is studied by Kobele (2002). In this grammar, a category can be selected to attach as a right or left daughter. Inflectional elements like *-s* attach V as a right daughter. In the resulting structure, a sequence of elements related by the complement relation is pronounced at highest strong category, and otherwise, left branch are pronounced before the root, and root before the right branch (if any). The lexical elements are lexically specified strong ::: or weak ::. To implement these ideas Kobele uses the following features:

category	N,V,A,P,C,D,…
select **as comp**	=N,=V,=A,=P,=C,=D,…
select **as spec**	N=,V=,A=,P=,C=,D=,…
licensor	+wh,+case,…
licensee	-wh,-case,…
phonetic+semantic	Romeo,Juliet,love,…

We have two two structure building operations, only slightly different from the more familiar ones above. There are again two cases of *merge*. Complement merge again checks and deletes the relevant features:

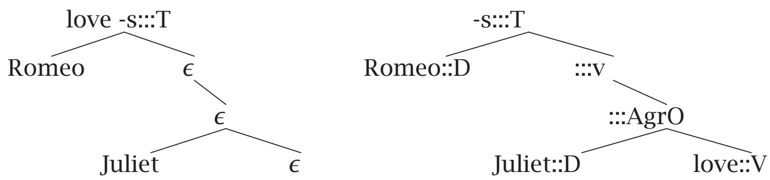
And merge specifier similarly checks and deletes features:

Move is triggered by a feature +f, as before, extending the structure by moving embedded maximal -f <u>specifier</u> to spec, checking and deleting features. This is just like movement in MG, MGH, except that we do not allow complements to move.

Consider for example the following grammar **MTTG1** with 6 lexical items:

-s:::=v +case T	ϵ:::=AgrO D= v	ϵ:::=V +case AgrO
love::D= V	Romeo::D -case	Juliet::D -case

With a simple calculation (which the reader should now be able to carry out), we derive 'bare structures' like those on the left, which can be annotated with categories as on the right:



Kobele (2002) considers a restriction (NRC) which blocks unboundedly complex morphological words. We will use MTTG to refer to that, restricted class of grammars. See Kobele (2002) for details.

4 First comparisons

Expressiveness: The 3 formalisms MG, MGH, MTTG all fall in the "controversial" region mentioned in the introduction, and they are equivalent (defining exactly the same languages, with closely comparable complexities). Writing CSG for the class of languages generated by context sensitive grammars, MCTAG for languages generated by (set-local) multiple component tree adjoining grammars, and LCFRS for languages generated by linear context free rewrite systems, we have the following relations among languages:

$$\text{CFG} \subset \text{TAG} \subset \boxed{\text{MG} \equiv \text{MGH} \equiv \text{MTTG}} \equiv \text{MCTAG} \equiv \text{LCFRS} \subset \text{CSG}$$

These results were established in a long tradition of work including, among others, (Harkema, 2001a; Michaelis, 2001b; Kobele, 2002; Michaelis, 1998; Weir, 1988; Seki et al., 1991; Vijay-Shanker and Weir, 1994).

Recognition: For the decidability of the languages defined by grammars near the mainstream of linguistic research, there has been a long line of discouraging complexity results, but the languages defined by MG, MGH and MTTG are all efficiently decidable:

– TG undecidable (Peters and Ritchie, 1973)

– LFG intractable (Berwick, 1981)

– UGs (hence HPSG,LFG) undecidable (Johnson, 1988; Torenvliet and Trautwein, 1995)

– GB intractable (Ristad, 1993)

\+ $\boxed{\text{MG}\equiv\text{MGH}\equiv\text{MTTG}}$≡MCTAG≡MCFG≡LCFRS tractable

\+ TAG≡CCG≡HG tractable (Joshi,Vijay-Shanker&Weir 1991)

\+ CFG tractable

The line of results involving MG, MGH and MTTG was established in the papers mentioned at the end of the **expressiveness** section just above.

Learnability: There are various formulations of the learning problem. In the tradition established by Gold's work, we have the following results for "identification of languages in the limit from positive text", where we say a grammar is k-valued iff it assigns at most k different syntactic categories to each lexical item:

- – TG, UGs (hence HPSG,LFG), GB, MG, MGH, MTTG, TAG, rCCG,…are not learnable from strings, even with learners capable of evaluating non-computable functions (Gold, 1967)
- + k-valued CGs ⊂ CFGs are learnable from strings (Kanazawa, 1996)
- + k-reversible regular languages are learnable from strings (Angluin, 1982)
- + finite sets learnable from strings (Gold, 1967)

We conjecture that k-valued MGs, MGHs and MTTGs are also learnable from strings (Kobele et al., 2002), but the proof has not yet been presented Stabler (2002). In any case, no differences in the learnability of the three kinds of grammars considered here, MGs, MGHs and MTTGs, have been discovered.

The very brief summary of formal research above does not reveal any significant differences among MG, MGH, MTTG. What kind of differences should we look for?

a. carefully exploring the details of particular constructions, we may find something that is appropriately handled only by (some elaboration of) one of these.
This is clearly an important strategy, but a result of this kind would be surprising because all 3 options very expressive.

b. expressively equivalent formalisms can differ in their **acquisition complexity**, so maybe only one of these can provide a reasonable acquisition theory

c. expressively equivalent formalisms can differ in the **succinctness** of their grammars for particular languages, and in the succinctness of their encodings of strings of those languages, so maybe one of these provides the "simplest" theory

Since appeals to the relative simplicity of one grammar over another are common in linguistic argumentation, let's briefly consider c.

5 Succinctness

Is there a meaningful way to compare the relative simplicity of grammars? We can of course observe:

MGH1 has	13 feature occurrences
MG1 has	15
MTTG1 has	15

But this comparison is not fair, because the grammars allow different numbers of feature types, and different operations.

We could try to provide a fairer measure for each framework $\mathcal{F} \in \{\text{MG,MGH,MTTG}\}$ by providing a grammar $G_{\mathcal{F}}$ that generates exactly the grammars in $\mathcal{F}$, then for any particular grammar G in any particular framework $\mathcal{F}$, we can let size(G)= the number of binary decisions required to specify the derivation of G from $G_{\mathcal{F}}$. One way of doing this is provided in Appendix A, and we find:

size(MGH1)=	68 bits
size(MG1)=	77 bits
size(MTTG1)=	84 bits

The size difference here are not dramatic. Experiments with larger grammars have not (yet) revealed anything more interesting than we see here, but this strategy does not really merit further investigation because there is still significant arbitrariness in these measures, coming from the choice of the particular coding scheme $G_{\mathcal{F}}$. Any particular choice in the coding scheme really requires justification.[1]

A different strategy for size comparison is to consider how simple these grammars make the data, e.g. the sentence *Romeo love -s Juliet.* Using the same measure as above, but this time in the particular derivation space of each framework, we find that they are all the same (2 bits), since the only choice is in

[1] It is true that the arbitrariness can be bounded (Li and Vitányi, 1997), but not tightly enough to make comparisons of particular small grammars like these meaningful. Berwick (1981), Clark (1994), Rissanen and Ristad (1994) and many others have proposed that measures of succinctness of roughly this kind should be relevant to acquisition complexity, but this will depend on the empirical motivation of the succinctness measure. Where can motivation come from? Stabler (1984) proposes (H): Choose a measure that makes learner's progression an increase in complexity.

the selection of the names. Is there any language L such that the smallest grammars for L in each of these frameworks differ significantly? We conjecture that the answer is no, but the question remains open.

6 Conclusions, open questions

We have see that with regard to expressive power,

MG≡MGH≡MTTG.

Consequently, data of the form

S, or *S,

for any expressions S, can never, by itself, be the basis for deciding among these approaches. The convergence of formalisms on the class of languages defined by these grammars provides some reason to believe that we are getting close to the natural class for natural languages, but does not provide any reason for preferring one of the equivalent formalisms over any other. With regard to recognition complexity, to the level of detail understood to date, again MG≡MGH≡MTTG, and all are tractable. We do not know how to establish any any relevant complexity differences. With regard to succinctness of smallest grammars for any *L*, to the level of detail understood to date, again MG≡MGH≡MTTG There is promising ongoing research on the learnability of k-valued MG, MGH, and MTTG from strings. It is reasonable to choose among equivalent formalisms those with simplest representations of child language and acquisition, but we have not yet discovered any reason for thinking that this favors one of the 3 kinds of grammar considered here. Many open questions remain.

References

Angluin, Dana. 1982. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29:741–765.

Berwick, Robert C. 1981. Computational complexity of lexical functional grammar. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics, ACL'81*, pages 7–12.

Brody, Michael. 1998. Projection and phrase structure. *Linguistic Inquiry*, 29:367–398.

Brody, Michael. 1999. Mirror theory: a brief sketch. In *A Celebration*. MIT Press, Cambridge, Massachusetts. Currently available at `http://mitpress.mit.edu/chomskydisc/brody.html`.

Brody, Michael. 2000. Mirror theory: syntactic representation in perfect syntax. *Linguistic Inquiry*, 31:29–56.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, IT-2:113–124.

Chomsky, Noam. 1995. *The Minimalist Program*. MIT Press, Cambridge, Massachusetts.

Clark, Robin. 1994. Kolmogorov complexity and the information content of parameters. Technical report, 94-17, Institute for Research in Cognitive Science, University of Pennsylvania.

Cornell, Thomas L. 1997. Representational minimalism. SFB 340 Technical Report #83, University of Tübingen. Revised version forthcoming in U. Mönnich and H.-P. Kolb, eds.

Cornell, Thomas L. 1998. Island effects in type logical approaches to the minimalist program. In *Proceedings of the Joint Conference on Formal Grammar, Head-Driven Phrase Structure Grammar, and Categorial Grammar, FHCG-98*, pages 279–288, Saarbrücken.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control*, 10:447–474.

Harkema, Henk. 2001a. A characterization of minimalist languages. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099, pages 193–211, NY. Springer.

Harkema, Henk. 2001b. *Parsing Minimalist Languages.* Ph.D. thesis, University of California, Los Angeles.

Johnson, Mark. 1988. *Attribute Value Logic and The Theory of Grammar.* Number 16 in CSLI Lecture Notes Series. Chicago University Press, Chicago.

Joshi, Aravind K., K. Vijay-Shanker, and David Weir. 1991. The convergence of mildly context sensitive grammar formalisms. In Peter Sells, Stuart Shieber, and Thomas Wasow, editors, *Foundational Issues in Natural Language Processing.* MIT Press, Cambridge, Massachusetts, pages 31–81.

Kanazawa, Makoto. 1996. Identification in the limit of categorial grammars. *Journal of Logic, Language, and Information*, 5:115–155.

Kobele, Gregory M. 2002. Formalizing mirror theory. *Grammars*, 5:177–221.

Kobele, Gregory M., Travis Collier, Charles Taylor, and Edward Stabler. 2002. Learning mirror theory. In *6th International Workshop on Tree Adjoining Grammars and Related Frameworks.*

Koopman, Hilda and Anna Szabolcsi. 2000. *Verbal Complexes.* MIT Press, Cambridge, Massachusetts.

Kulick, Seth. 2000. *Constraining non-local dependencies in tree adjoining grammar: computational and linguistic perspectives.* Ph.D. thesis, University of Pennsylvania.

Li, Ming and Paul Vitányi. 1997. *An Introduction to Kolmogorov Complexity.* Springer-Verlag, NY. Second edition.

Mahajan, Anoop. 2000. Eliminating head movement. In *The 23rd Generative Linguistics in the Old World Colloquium, GLOW '2000, Newsletter #44*, pages 44–45.

Matthews, Robert. 1979. Do the grammatical sentences of language form a recursive set? *Synthese*, 40:209–224.

Michaelis, Jens. 1998. Derivational minimalism is mildly context-sensitive. In *Proceedings, Logical Aspects of Computational Linguistics, LACL'98*, NY. Springer.

Michaelis, Jens. 2001a. *On Formal Properties of Minimalist Grammars*. Ph.D. thesis, Universität Potsdam. *Linguistics in Potsdam 13*, Universitätsbibliothek, Potsdam, Germany.

Michaelis, Jens. 2001b. Transforming linear context free rewriting systems into minimalist grammars. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099, pages 228–244, NY. Springer.

Michaelis, Jens and Marcus Kracht. 1997. Semilinearity as a syntactic invariant. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, pages 37–40, NY. Springer-Verlag (Lecture Notes in Computer Science 1328).

Peters, P. Stanley and R. W. Ritchie. 1973. On the generative power of transformational grammar. *Information Sciences*, 6:49–83.

Putnam, Hilary. 1961. Some issues in the theory of grammar. In *Proceedings of Symposia in Applied Mathematics. 12: 25-42*. Reprinted in *Mind, Language and Reality: Philosophical Papers, Volume 2*, by Hilary Putnam. New York: Cambridge University Press.

Rambow, Owen. 1994. *Formal and computational aspects of natural language syntax*. Ph.D. thesis, University of Pennsylvania. Computer and Information Science Technical report MS-CIS-94-52 (LINC LAB 278).

Rissanen, Jorma and Eric Ristad. 1994. Language acquistion in the MDL framework. In Eric Ristad, editor, *Language Computations*. American Mathematical Society, Philadelphia.

Ristad, Eric. 1993. *The Language Complexity Game*. MIT Press, Cambridge, Massachusetts.

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

Stabler, Edward P. 1984. Berwick and Weinberg on linguistics and computational psychology. *Cognition*, 17:155–179.

Stabler, Edward P. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*. Springer-Verlag (Lecture Notes in Computer Science 1328), NY, pages 68–95.

Stabler, Edward P. 1998. Remnant movement and complexity. In *Joint Conference on Formal Grammar, Head-Driven Phrase Structure Grammar, and Categorial Grammar, FHCG-98*, Saarbrücken, Germany. Universität des Saarlandes.

Stabler, Edward P. 1999. Remnant movement and complexity. In Gosse Bouma, Erhard Hinrichs, Geert-Jan Kruijff, and Dick Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*. CSLI, Stanford, California, pages 299–326.

Stabler, Edward P. 2001. Recognizing head movement. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099. Springer, NY, pages 254–260.

Stabler, Edward P. 2002. Identifying minimalist languages from dependency structures. http://www.clsp.jhu.edu/seminars/abstracts/S2002/stabler.shtml.

Steedman, Mark J. 2000. *The Syntactic Process*. MIT Press, Cambridge, Massachusetts.

Torenvliet, Leen and Marten Trautwein. 1995. A note on the complexity of restricted attribute-value grammars. In *Proceedings of Computational Linguistics In the Netherlands, CLIN5*, pages 145–164.

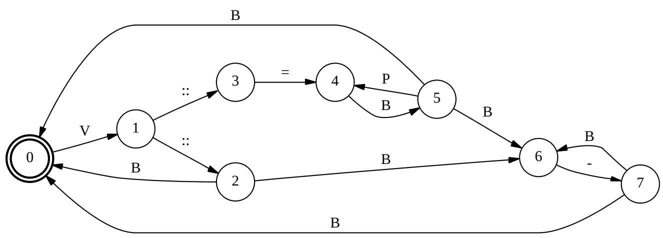
Vijay-Shanker, K. and David Weir. 1994. The equivalence of four extensions of context free grammar formalisms. *Mathematical Systems Theory*, 27:511–545.

Weir, David. 1988. *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

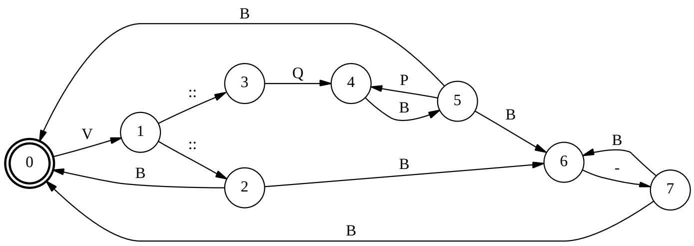
A Appendix: size measures for grammars

In each framework, the grammars over any vocabulary Σ and base features B form a regular set, so we provide finite state generators.

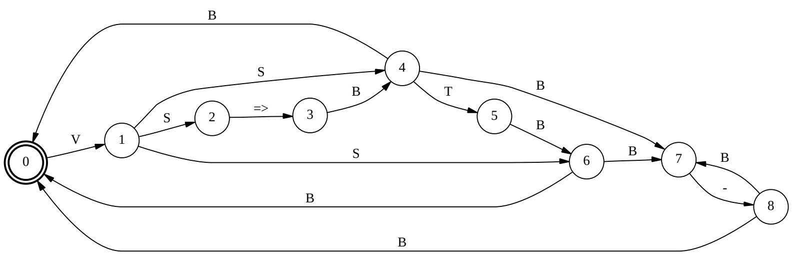
Generator for MGs, where B is the set of basic features, $V = \Sigma \cup \{\epsilon\}$, $P = \{=, +\}$. An arc labeled B is an abbreviation for $|B|$ arcs each labeled with a member of B – a choice among these arcs represents $\log_2(|B|)$ binary choices, and similarly for the other sets.



Generator for MGHs, where B, V, P as above, $Q = \{=, =>\}$



Generator for MTTGs, with B as above, $S = \{::, :::\}$, $T = \{<=, +\}$



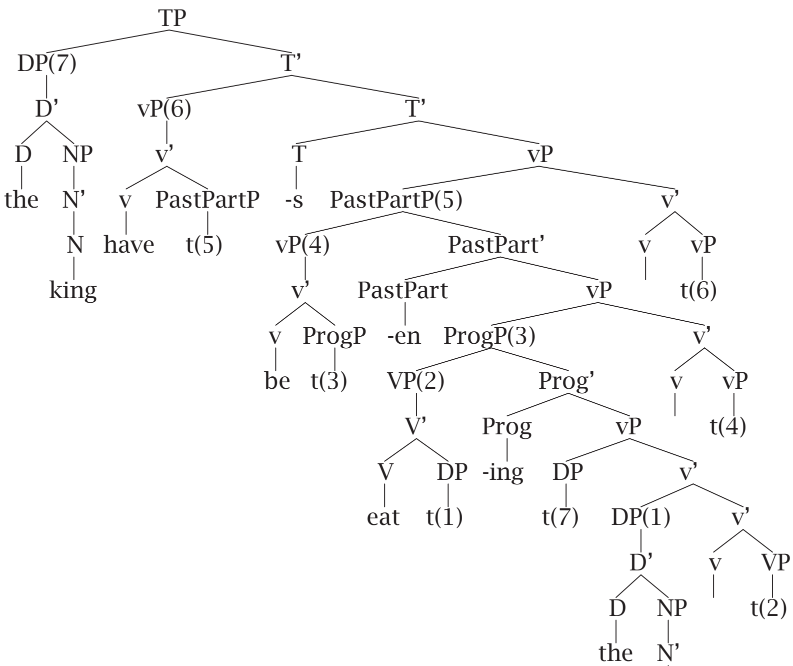
B Appendix: extending MG1, MGH1, MTTG1

Obviously there are many ways to extend the tiny grammars shown above. For estimates of succinctness, etc, it is useful to consider extensions like these. (These grammars are not presented as correct ones(!), but only as further examples to illustrate the different mechanisms of the respective frameworks)

Consider the slightly more elaborate MG1':

-s::=v +v +case T	ϵ::=V +case =D v	
the::=N D -case	king::N	pie::N
ϵ::=v +aux v	eat::=D V -v	-ing::=v +v Prog -aux
have::=PastPart v -v	be::=Prog v -v	-en::=v +v PastPart -aux

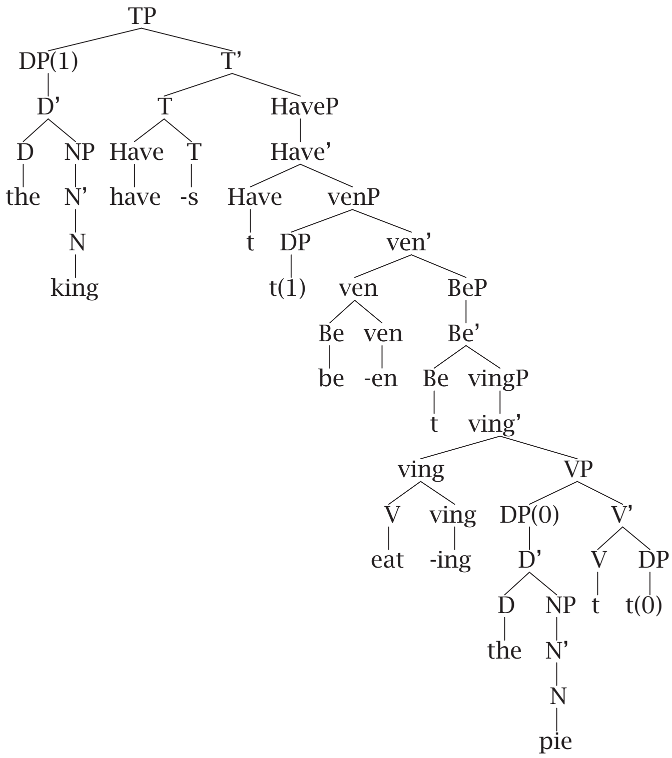
Then we can derive *the king have -s be -en eat -ing the pie*:



Consider the slightly more elaborate MGH1':

-s::=>Have +case T	have::=ven Have	-en::=>Be =D ven
eat::=D +case V	the::=N D -case	pie::N
be::=ving Be	-ing::=>V ving	king::N

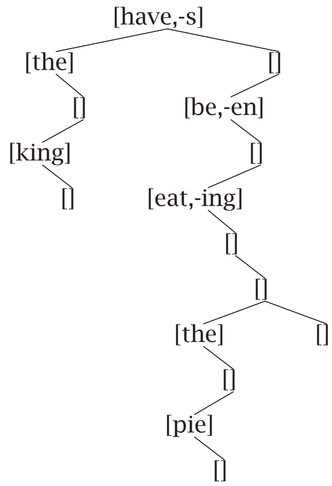
Then we can derive:



Consider the slightly more elaborate MTTG1':

the::AgrN= D	ϵ:::=D AgrD -case	king::N	
ϵ:::=N AgrN	eat::AgrD= V	ϵ:::=V +case AgrO	
ϵ:::=AgrO AgrD= v	be::ving= Be	-ing:::=v ving	
have::ven= Have	-en:::=Be ven	-s:::=Have +case T	pie::N

Then we can derive:



We have efficient parsers for all 3 frameworks (proven sound and complete for all grammars in each framework). I used simple implementations of these parsers to compute these derivations and format the trees, but of course it can be done by hand too. The simple computer implementations are available at

`http://www.linguistics.ucla.edu/people/stabler/`.