ISSN 0385-8855

英語英文学研究所
紀要

第 46 号

2022

東北学院大学

英語英文学研究所紀要 46 東北学院大学

Journal of Institute for Research in English Language and Literature

46

2022

東北学院大学英語英文学研究所

Simple Models: Computational and Linguistic Perspectives*

Sandiway Fong
University of Arizona

1. Introduction

The term *simplicity* is highly overloaded and means different things (to different people) in different disciplines when it comes to language, and in particular, the modeling of language. For rich and complex human language, model simplicity is motivated by explanatory depth, acquisition plausibility, and more recently, evolutionary plausibility. The last three have been highlighted as conditions for a *Genuine Explanation* (Chomsky 2021b).

Chomsky (1956) contrasts the empirical coverage of three formalisms, *viz*. (i) linear concatenation, (ii) (hierarchical) phrase structure, and (iii) phrase structure augmented with transformations, in order of expressive power and decreasing simplicity. Chomsky asks an important question, why is it that humans, from infancy, ignore linear order in the incoming signal? Instead, we reflexively use structures that we never hear (but the mind constructs) for all non-trivial computations.

Examples (1a-b) below are adapted from (Chomsky 2021b).

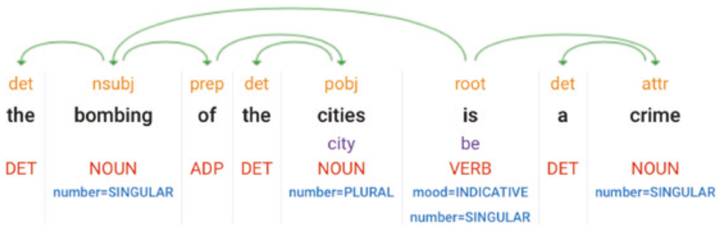
---

* This paper is based on a presentation given in October 2021 at *Tohoku Gakuin University*. The author is grateful to Robert Berwick, Noam Chomsky and Jason Ginsburg for comments on earlier drafts. All remaining errors and misconceptions belong solely to the author. This research was partially supported by a Japan Society for the Promotion of Science Grant-in-Aid for Scientific Research #20K00664 to Jason Ginsburg.

(1) a. the bombing$_{[sg]}$ of the cities$_{[pl]}$ is$_{[sg]}$ a crime
  b. *the bombing$_{[sg]}$ of the cities$_{[pl]}$ are$_{[pl]}$ a crime

A linear proximity metric for subject-verb agreement favors (1b); however, Agreement requires hierarchical proximity as illustrated below in (2) via *Google Natural Language*'s dependency representation.[1]

(2)



In (2), the main verb *be* agrees with the hierarchically adjacent noun *bombing*, measured in terms of the number of connecting links (as being 1 link away), rather than with the linearly adjacent noun *cities* (3 links distant in this representation).

These (and many more) examples demonstrate that the *Language Faculty* of the human brain ignores the simplest relation in its cognitive repertoire, linear adjacency, in favor of hierarchical adjacency. It is reasonable to assume that a toddler could not have converged on this machinery from scratch in short order. Chomsky also points out that there is experimental evidence that children as young as toddlers can distinguish between singular [sg] and plural [pl] agreement in the case of coordinated vs. non-coordinated nouns. Hence, the primitive operation of language is *Merge*, providing the ability to recursively build hierarchical structures starting with lexical items, and the selection of *Merge* over *Linear Concatenation* as the relevant primitive is part of our ge-

[1] See <https://cloud.google.com/natural-language>. Note the use of this representation is for illustrative purposes only. *Google Natural Language* does not compute the Agreement rule, *e.g.* it assigns an identical structural description to that of (1a) in the case of (1b).

netic endowment.

2. *N*-gram Models of Language

The idea of a simple *n*-gram concatenative model for human language, *e.g.* (3a-b), certainly predates Shannon (1948).

(3) a. unigram (1-gram), a single word
b. bigram (2-gram), 2 adjacent words (termed a *digram* by Shannon)
c. trigram (3-gram), 3 words in a row

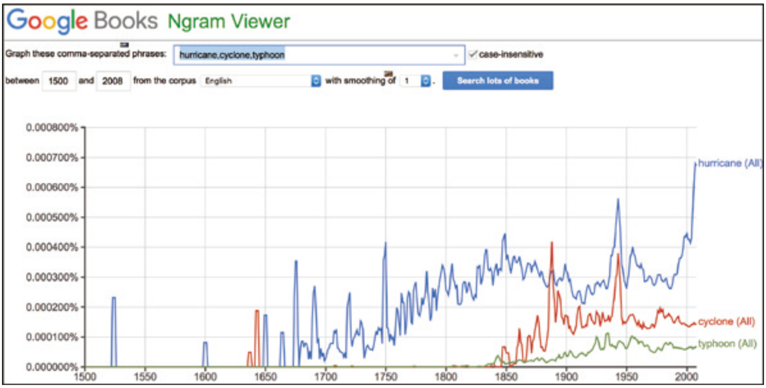
It has been well-known for a long time that linear regularities in language make ciphers vulnerable to attack using bigram and trigram probabilities; for example, Pratt (1939) cites *Francesco Simonetta*'s alphabetic frequency tables from 1474. The most frequent letter bigrams and trigrams across several languages, from Pratt (1939), are shown in (4) and (5) (in descending order of frequency).

(4) a. French: ES - EX - SE - ON - DE - TE - NT - LE - ET - AT - ON
b. German: EX - ER - OH - DE - GE - IE - El - ND - IN - TE - RE
c. Spanish: ES - EX - EL - DE - LA - OS - AR - UE - RA - RE - ER
d. Italian: ER - ES - OX - RE - EL - EN - DE - DI - TI - SI - AL

(5) a. French: EXT- QUE- LES - ION - AIT - TIO - ONT- ANS- ART - AIN - OUR - OUS
b. German: EIX - ICH - DEN - DER - TEN - CHT - SCH - CHE - DIE - UNG - GEN - UND
c. Spanish: QUE - EST - ARA - ADO - AQU - DEL - CIO - ETE - OSA - EDE - PER - IST
d. Italian: CHE- ERE - ZIO - DEL - ECO - QUE - ARI - ATO- EDI - IDE - ESI - IDI

With modern technology, letter regularities can scale up to word regularities, although obviously, the tables are much larger. Given any text corpus, or millions of machine-readable articles and books, we can straightforwardly record all associated *n*-gram statistics. This has always been simple to compute, provided sufficient resources are available. The presence of *Google Books N-gram Viewer* indicates that computer technology has solved both the scale and the memory storage problem.[2] Meta-data for *n*-grams, including a record of publication date, allow us to see when words (and bigrams) enter the language. For example, the diagram in (6) provides comparative frequency information for the English words *hurricane*, *cyclone* and *typhoon*, region-specific names for the same weather phenomenon.

(6)



According to the OED,[3] *hurricane* is the oldest term of the three, originating from Carib *huracan*, the current spelling became frequent after 1650, and was

---

[2] See <https://books.google.com/ngrams>. As of October 2019, about 40 million books have been "scanned". Note there are other sources of scanned books, *e.g. Project Gutenberg*, begun in 1971, contains about 60,000 books (taken only from public domain sources) predates Google's effort, but no *n*-gram statistics are provided.

[3] *Oxford English Dictionary*, online edition: <https://www.oed.com.>

established from 1688. The term *cyclone* is a neologism introduced by Henry Piddington in 1848, confirmed by the Google statistics, and as the umbrella term *tropical cyclone*, adopted by the scientific community. *Typhoon* comes from Cantonese *daai fung* (大風, *big wind*) according to the OED, with current spelling dating from around 1820. Again, this seems to correlate well with the Google data.

In the case of pop culture bigrams *Star Wars* and *Star Trek*, shown in (7), frequency peaks generally correlate around movie release and TV broadcast time periods.

(7)

Thus, these peaks receive simple explanations except for one notable exception; there is a huge (unexplained) blip in the bigram statistics for *Star Wars* peaking around 1987. The April 4th *Time* magazine cover from 1983, together with the time correlation with the trigram *Strategic Defense Initiative* in (8), provides a reasonable explanation for the unexpected popularity of this term.[4]

---

[4] This blip cannot be attributed to the *Star Wars* franchise because there was a large gap between the 1983 release of the third *Star Wars* movie and the fourth in 1999, a time period during which George Lucas (the creator), later revealed in interviews, did not plan to continue the series.

(8)

Despite such examples, probabilistic n-gram models, no matter how large the corpus, perform poorly when it comes to modeling human language. As Chomsky noted back in 1956, repeated below in (9), there need not be correlation between the statistical frequency of an n-gram sequence and its grammaticality.

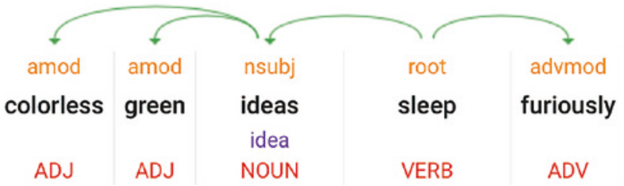
(9) [t]here is no general relation between the frequency of a string (or its component parts) and its grammaticalness. (§2.4: Chomsky 1956)

The now-famous constructed example Chomsky chose to illustrate the point is given in (10a) below, to be contrasted with (10b).

(10) a. colorless green ideas sleep furiously
b. *furiously sleep ideas green colorless

Note that although (10a) and (10b) contain the same five words, and both strings are unfamiliar to the reader, (10a) obeys the basic rules of English grammar, and, though difficult to interpret, is technically grammatical in the sense we can assign plausible syntactic structure, see (11) for instance, whereas (10b) does not obey the same rules, and is completely ungrammatical.

(11)



In (11), *Google Natural Language* correctly assigns the right parts-of-speech for each word and gives (10a) a parse as an unergative sentence headed by *sleep*.[5] Let us turn to consider the bigrams of (10a) and (10b), listed in (12a-b), respectively.

(12) a. i. colorless green, ii. green ideas, iii. ideas sleep,
iv. sleep furiously
b. i. furiously sleep, ii. sleep ideas, iii. ideas green,
iv. green colorless

Chomsky's observation is that the bigram frequencies for those listed in (12a) and (12b) are likely indistinguishable. Nevertheless, *n*-gram models are versatile and popular, engineers have applied *n*-gram models across many domains, as diverse as speech recognition and gene sequence analysis. Nearly half a century later, Pereira (2002) reported that an aggregate bigram language model (trained over newspaper text) achieved the relative probability estimate shown in (13) below, distinguishing between (10a) and (10b) by five orders of magnitude, and (in Pereira's words) "*a suitably constrained statistical model can meet Chomsky's particular challenge*."[6]

---

5 Note that *sleep* can be either a noun or a verb. In the case of strongly ungrammatical (10b), *Google Natural Language* analyses it as verbal *sleep* taking two objects, *ideas* as a direct object and an adjectival complement headed by *colorless*.

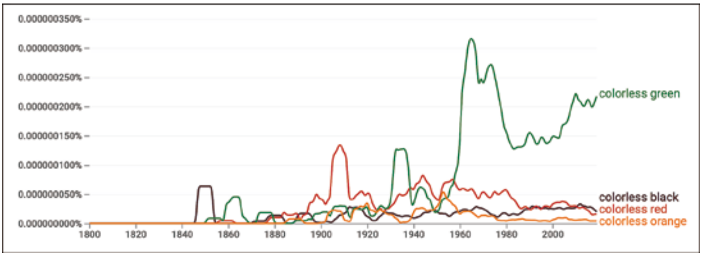
6 See Berwick (2018) also for extensive discussion of Pereira's claim. The aggregate bigram model is not a full word bigram mode (except when C = vocabulary size). Instead, it computes $P(w_2|w_1) = \Sigma_{c\in C} P(w_2|c)\ P(c|w_1)$, where C is a model parameter, representing the number of word classes. Although none of the bigrams listed in (12a-b) appear in the 3 million sentence training set, all the words do

(13) $$\frac{P(\textit{Colorless green ideas sleep furiously})}{P(\textit{Furiously sleep ideas green colorless})} \approx 2 \times 10^5$$

Postponing an expanded discussion of (13) for now, let us simply observe that (13) represents just one pairwise comparison in the space of possible ungrammatical sentences. Given 5 words, there are 120 (= 5!) possible permutations. A good language model should rank (10a) above all ungrammatical permutations (and perhaps above all possible permutations).[7]

Incidentally, Chomsky (p.c.) has also observed that (10a) is no longer an ideal example to use (as it has been well-discussed over the last half century or so), suggesting, instead, that substituting *orange* for *green* would reset the bigram statistics, as (14a-b) below indicates.

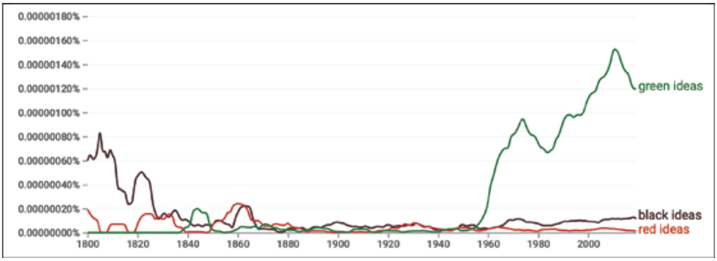
(14) a. bigram statistics for *colorless green/black/red/orange*



appear. The relevant word counts are *colorless*: 71, *green*: 4807, *ideas*: 2764, *sleep*: 1307, *furiously*: 84. The model can therefore compute distributional information about them with respect to the classes *C*.

7 (10a) is not the only grammatical sequence achievable with these five words. For example, we can permute adjectives *colorless* and *green*, and displace the adverb *furiously* from the post-verbal to the pre-verbal position without any appreciable change in grammaticality. See also Sprouse, *et al*. (2018) for discussion of trigram model performance with respect to grammaticality.

b. bigram statistics for *green/black/red/orange ideas*



The bigram *orange ideas* is missing entirely in (14b), this is because it is not found in Google's corpus.[8] However, it is unlikely that Pereira's result, given in (13) above, being based on newspaper text, would be subject to this effect. Nevertheless, validating that single datum against *orange ideas* (substituting for *green ideas*) would seem prudent, as the corresponding *Google Natural Language* parse in (15) is subtly different from the one in (11).

(15)

A *n*-gram model can be used to probabilistically generate (or predict) the next word based on the preceding *n* words, as outlined in Shannon (1948). (16a-b) are the unigram and bigram examples he chose to illustrate this pro-

[8] When a bigram does not exist in the training corpus, its frequency *has* to receive a non-zero estimate because the Chain Rule for calculating the probability of a sequence involves multiplication (as "zero by anything" is zero). Common *smoothing techniques* including setting its probability mass to a very small constant or giving those "zero" bigrams a probability mass based on the current probability mass for bigrams that occur once.

cess, respectively. The idea is that as n increases, the generated text will more closely approximate English, assuming that the n-gram frequencies are drawn from a large, representative sample of the English language. Today, this information is directly available through massive n-gram tables up to 5-grams, courtesy of *Google Books*.[9]

(16) a. representing and speedily is an good apt or come can different natural here he the a in came the to of to expert gray come to furnishes the line message had be these.

b. the head and in frontal attack on an English writer that the character of this point is therefore another method for the letters that the time of who ever told the problem for an unexpected.

However, Shannon used an interesting proxy method, practical in his day, to randomly select a bigram w_1w_2. The method is as follows: open a book at a random page, pick a (random) word and read until the word w_1 is found, then w_2 is the next word, and repeat all over again for the next word. Shannon remarked:

(17) The resemblance to ordinary English text increases quite noticeably at each of the above steps. … It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source. (Shannon 1948)

That is, the bigram-based (16b) can be informally judged as being closer to grammatical English than the unigram-based (16a). However, there is no reason to expect that any two (attested) n-grams will overlap to form a larger n-gram that is grammatical. (Linguists have observed that grammatical sequences are formed through hierarchical phrase structure.) Even in the less rigid language of poetry and songs, it is hard to make sense of generated bigram

9 See <https://storage.googleapis.com/books/ngrams/books/datasetsv2.html>.

lyrics such as those in (18a-e) below.[10]

(18) a. I me wrong I would upset the hippy hippy shake
b. I love her all belong . Try to , And
c. I beg you , baby , We 'll go ,
d. I write this letter for me down ! A little
e. I 'm gon na do . That boy would love

These lines, except for the 1st word *I*, were generated probabilistically using bi-grams from a corpus of about 300 songs, written by the *Beatles*, a prolific, popular music group from the 1960s and 70s. In total there are about 58,000 words, the mostly frequent ones are shown in visual word cloud form (taken from the *Guardian*), and with raw counts in (19b).

(19) a. Beatles lyrics word cloud[11] b. top 25 words

,	.	I	you	the
4246	2505	2487	2216	1255
it	do	and	my	me
1076	1071	878	803	765
to	's	a	n't	love
633	617	593	570	567
be	that	in	know	'm
492	485	462	434	404
You	your	And	of	on
385	383	369	363	356

Even when scaled up to *Google Book* dimensions, *n*-grams models turn out to be architecturally inappropriate models for language. (See also later discussion on the deep neural net system GPT-2.)

[10] Note that punctuation marks are counted as words.
[11] The word cloud was sourced from the November 16th 2010 online edition of the *Guardian*, <https://www.theguardian.com/music/datablog/2010/nov/16/beatles-lyrics-words-music-itunes>. Note that stop words, *e.g.* determiners *a* and *the*, and prepositions, are excluded from the word cloud, but they are present in the corpus. In this model, words include punctuation, and contractions such as *can't* or *wanna* each count as two separate words.

3. Phrase Structure Grammar-based Models of Language

Let us consider again the single data point represented by (13), and conduct a more extensive experiment using all 120 permutations of Chomsky's example from (10a). Using a statistical phrase structure-based parser, we train the parser on a treebank based on newspaper text, and compare (10a) with all its permutations, not just (10b), in terms of calculated probability.[12] (We emphasize that no bigram frequencies are used in the phrase structure model, *cf*. Pereira's experiment.[13]) Top-10 rankings for the permutations of (10a) are shown in (20a-j) below.

(20) a. #1: furiously ideas sleep colorless green
b. #2: furiously green ideas sleep colorless
c. #3: green ideas sleep furiously colorless
d. #4: colorless ideas sleep furiously green
e. #5: green ideas furiously sleep colorless
f. #6: furiously ideas sleep green colorless
g. #7: ideas sleep colorless furiously green
h. #8: furiously ideas colorless sleep green
i. #9: green ideas sleep colorless furiously
j. #10: furiously colorless ideas sleep green

Two observations come to mind immediately: (i) there does not appear to be any correlation between grammaticality and the rankings in the best 10 list, and (ii) (10a), a grammatical example, is not ranked highly enough to make an

[12] The Bikel re-implementation of the Collins Parser was used in this experiment (Bikel 2004). Relative rankings are based on the logprob score of the top parse. Part-of-speech tags were supplied to the trained parser and held constant across all permutations for the words in (10a). All training data were taken from the *Wall Street Journal* (WSJ) sub-corpus of the Penn Treebank (PTB) v3 (Marcus, *et al*. 1999).

[13] The training corpus, identified in note 12, contains no data related to Chomsky's example. For example, it contains no occurrence of *colorless* (though *colorlessness* occurs once) and only two of *furiously*, respectively.

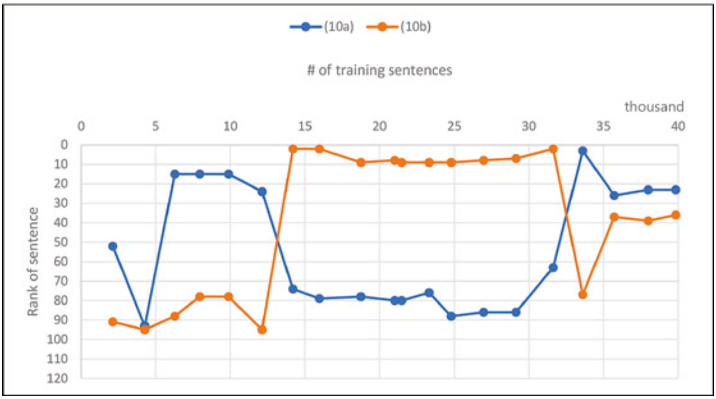
appearance in this list.[14] In fact, (10a) only ranks 23rd. As for (10b), the clearly ungrammatical variant, it ranks 36th out of 120. Although (10a) does outrank (10b) in this experiment (as well as in Pereira's experiment), the bigger picture shows that we have not distinguished grammatical from ungrammatical, and Chomsky's observation, reproduced in (21) below, still holds.

(21) In §2.4 we argued that there is no significant correlation between order of approximation and grammaticalness. If we order the strings of a given length in terms of order of approximation to English, **we shall find both grammatical and ungrammatical strings scattered throughout the list**, from top to bottom. Hence the notion of statistical approximation appears to be irrelevant to grammar. (Chomsky 1956)

We can ask the question, does grammatical (10a) consistently outrank ungrammatical (10b) when the amount of training data is varied? In other words, are the relative rankings stable? The chart in (22) below exhibits two lines, the blue one for (10a), and orange for (10b).

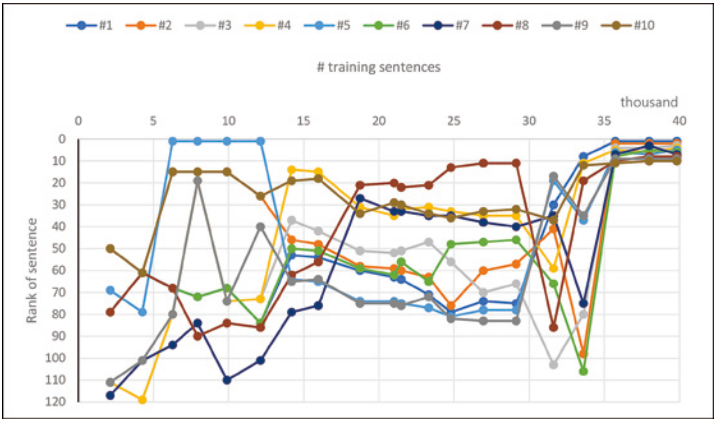
[14] No judgments are shown in the case of examples (20a-j). Chomsky (1956) offers an interesting diagnostic. He observes that (10a) is read with ordinary sentence intonation, but (10b) will be read with falling intonation on each word. The reader is invited to test examples (20a-j) in this fashion to confirm this holds.

(22)



Although (10a) ranks higher than (10b) when 34,000-40,000 treebank sentences are used in training, in the region from about 15,000-32,000 sentences, not only does the ungrammatical sentence rank higher, but it achieves a top-10 score for this interval, a score not achieved by grammatical (10a) at any stage of the experiment. This calls into question the stability of the statistical system. Further experimentation confirms the observed instability. The chart in (23) below shows how the top-10 ranked sentences in (20a-j) vary in ranking with respect to the amount of training data.

(23)



Phrase Structure (PS) Grammars are usually defined around Context-Free Grammar (CFG) rules of the form in (24a-b).[15, 16]

(24) a. $\text{Phrase}_0 \rightarrow \text{Phrase}_1, \ldots, \text{Phrase}_n$ for finite $n \geq 0$
b. Phrase $\rightarrow$ lex for lex $\in$ LEX, *i.e.* the lexicon

Our experiment involved hierarchical phrase structure encoded using CFG rules, the second formalism examined in (Chomsky 1956). In particular, we employed a series of CFGs of increasing size, augmented with probabilities inferred from a set of *gold-standard* phrase structure trees, termed a treebank. One ready advantage of CFGs is that perception is not a problem,

---

[15] Alternative PS grammar formalisms exist for natural languages, *e.g.* Tree Adjoining Grammars (Joshi 1985) and other work on mildly context-sensitive languages.

[16] The number of rules is of both theoretical and practical importance. In (24a-b), the total number of rules must be finite. (If an infinite number of rules is permitted, the expressive power of the resulting system obviously exceeds that of CFGs.) The number of rules is also of practical importance in the case of probabilistic CFGs for natural language, as the probability mass associated with unknown words, *i.e.* words not in the training set, and therefore, rules of the form (24b), must also be estimated.

practical algorithms exist for efficiently computing all possible syntactic structures from linear input.[17] However, from another viewpoint, CFGs are too unconstrained; in principle, all combinations of phrases, both exo- and endocentric, are possible in this framework. From the point of view of empirical coverage, CFGs are too broad. At the same time, CFGs are a poor choice of formalism for encoding many types of structurally-determined relations, *e.g.* displacement, control, long-distance agreement or pronominal binding. These cannot be succinctly represented.[18] CFGs also pose an acquisition problem that contrasts with the human experience. Unlike the case of the cognitively-unrealistic treebank containing already-parsed sentences, hierarchical structure is not explicitly represented in primary linguistic data. Even if we back off from cognitively-motivated conditions, and treat CFG training purely as an engineering exercise, there exists a serious number-of-degrees-of-freedom problem. As Bikel observes:

(25) … it may come as a surprise that the [parser] needs to access more than **219 million probabilities** during the course of parsing the 1,917 sentences of Section 00 [of the Penn Treebank: SF]. (Bikel 2004)

The probabilistic CFG system is also surprisingly sensitive to perturbation in the training data, even beyond what the charts in (22) and (23) have suggested. Another experiment with prepositional phrase (PP) attachment ambiguity, an important task for any syntactic parser, confirms this problem,

[17] For example, a fast and general method based on LR parsing is described in (Tomita 1987).

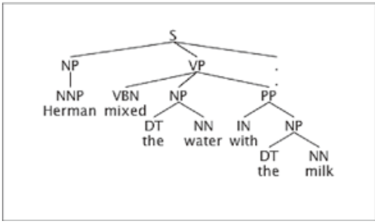
[18] While it is possible, in principle, to write specialized CFG rules on a case-by-case basis, this is rarely done, as “unrolling” just a few instances of a general structural relation, *e.g.* binding, would lead to massive rule proliferation. Instead, general-purpose descriptive devices, *e.g.* to facilitate formal feature structure passing and valuation, are typically grafted onto the framework for this purpose, as in the case of unification-based PS grammars, such as HPSG (Pollard & Sag 1994). In the statistical CFG realm, parsers do not address the computation of such relations, although the relations may actually be available in the treebank training corpus, *e.g.* indicated by coindexation.

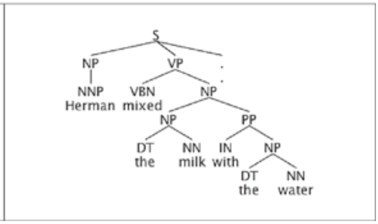
despite the many thousands of treebank sentences available for training.[19] Consider the similar sentences in (26a-b).

(26) a. Herman mixed the water with the milk
b. Herman mixed the milk with the water
c. Herman drank the water with the milk
d. Herman drank the milk with the water

It is reasonable to assume stability in the sense that a statistical parser will assign phrase structure consistently in these four examples. However, we can observe a difference between high and low prepositional phrase (PP) attachment, as illustrated in (27a-b) below.

(27) a. High PP attachment (to VP) b. Low PP attachment (to NP)

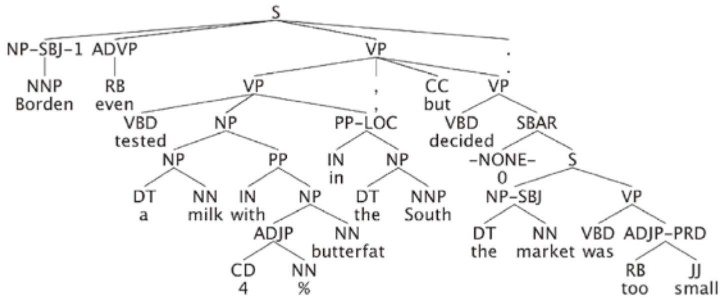




As it turns out, although there are 24 sentences in the treebank corpus with *milk* as a word, 21 as a noun, there is only one training example with PP attachment data for *milk*, shown in (28).

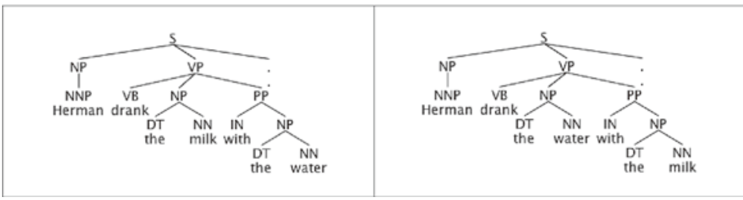
[19] For details of the experiment, see (Fong & Berwick 2008).

(28)



In (28), the PP *with 4% butterfat* exhibits low attachment to the noun phrase headed by *milk*. This single example is enough to account for the low attachment seen in (27b). To confirm this, it is enough to delete the relevant PP from (28) and retrain the parser. The result is high attachment for both cases, as shown in (29a–b).

(29) a. high PP attachment (to VP) b. high PP attachment (to VP)

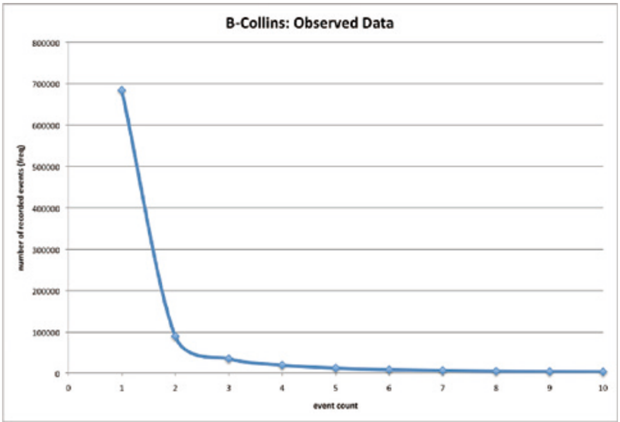


The reason for this extreme sensitivity to perturbation in the training data is that there are millions of parameters that need to be estimated, and this particular parser makes use of nearly every statistical event (recorded during training), even if those events occur only once.[20] In fact, as the graph in (30) indicates, 94% of the time, there are 5 or fewer occurrences of the same event, and in 77% of the time, events occur just once (as in the *milk* example), de-

[20] Of course, not all parsers need make the same PP attachment choices, and one could *tune* the system using some kind of threshold (to exclude such singular events), at the cost of ignoring much of the training data.

spite nearly 40,000 sentences of training data.

(30)



4. Current and Future Directions for Computational Language Modeling

In the previous section, we have seen how apparently *simple* models such as CFGs are actually much more complex than they appear to be. Recently, general-purpose deep neural networks have been adopted that contain vastly more parameters than the statistical CFG models. For example, the well-known GPT-2 neural net model has 1.5B parameters, and the next-generation GPT-3 model has 175B parameters (Brown, *et al*. 2020). GPT-2 can be used for various language tasks, including summarization and translation, but the next-word language modeling task is the principal one for its transformer-decoder architecture.[21] General-purpose systems are attractive to the engineering community, advantages include flexibility across problem sets and

[21] These systems likely are equally (or more) adept at modeling certain kinds of artificially-constructed languages than natural languages. On the other hand, humans are strongly biased towards natural languages, as experiments have revealed. See Moro (2016).

(non-language) domains. There is also an intuitive appeal in setup simplicity as nothing necessarily particular to language is hardcoded ahead of time.[22] One can regard these general systems as a continuation of the behaviorist conception that language is a "*matter of training and habit*."[23] However, with so many parameters, the chief downsides are that a lot of training data is required, much more than what seems to be cognitively plausible, and that there are burdensome requirements in terms of computational resources (for training).[24] The term *overparameterization* is used when a model has many more parameters than data points, like in GPT-2's case, potentially leading to *overfitting*, *i.e.* memorization of the training data, rather than true generalization.

To provide some sense of what these models are capable of, let us conduct an experiment to see how GPT-2 completes possible prefixes of (10a). In (31a-e) below, the hand-typed input is <u>underlined</u>, followed by GPT-2's sample continuation.[25]

(31) a. <u>colorless</u> green ideas came in different varieties.
b. <u>colorless green</u> ideas were a group of people who had different opinions
c. <u>colorless green ideas</u> were not the same as the other colors.
d. <u>colorless green ideas sleep</u>, and the blue ones dream.

---

[22] As these systems deal with floating point vector representations, words must be vectorized first. We can regard that as a preliminary step, along with tokenization.

[23] See, for example, (Chomsky 2021a) for discussion of the behaviorist view of language that predominated part of the 20^{th} century.

[24] Humans "learn" language in spite of being given relatively-little exposure to linguistic data, this is termed the *poverty of stimulus*. Deep neural network systems require much more data than available to us. For example, GPT-2 was trained on the WebText corpus, containing about 40 GB of text data. In the case of English, 40 GB is not particularly burdensome, but in the case of under-resourced languages, large amounts of training data may never become available.

[25] This experiment was conducted online via the interface at <https://transformer.huggingface.co/doc/gpt2-large>. Note that the response is randomized to some extent, *i.e.* GPT-2 may not come up with the same continuation on retries. A smaller GPT-2 model containing 345M parameters is also available online at <https://demo.allennlp.org/>.

e. colorless green ideas sleep furiously in the memory of their old life: and the green idea awoke at the sound of the new ones, and cried for joy as the first-born had done.

We observe that GPT-2 completed the prefixes grammatically in each case, and did not simply memorize the now-famous Chomsky example.[26] Furthermore, these examples seem to lend weight to the (much-hyped) claim that the system is not only grammatical, but also creative, being ultimately capable of generating "*stories, poems and (fake news) articles*" (Wakefield 2019). When given considerably more context, the suggestion is that GPT-2 may predict the intentions of the human writer, as the snippet in (32) below, taken from a *New Yorker* magazine article seems to indicate.

(32)

> And yet, sitting there at the keyboard, I could feel the uncanny valley prickling my neck. It wasn't that Smart Compose had guessed correctly where my thoughts were headed—in fact, it hadn't. The creepy thing was that the machine was more thoughtful than I was.
>
> Read Predicted Text
>
> Generated by GPT-2 (including any quotes)
>
> By that I mean, it seemed to want to distinguish my feelings from my thoughts. To put it another way, Smart Compose seemed to want to know me.

(Seabrook 2019)

The next-generation GPT-3 model has 175B parameters, two orders of magnitude greater than GPT-2, thus requiring substantial increases in both training data and computing time to converge. However, there is some evidence that the performance of these systems can scale with parameter size, assuming

[26] We ignore stylistic rules, *e.g.* the unusual use of the colon in (31e).

enough training data is available (in order to avoid the problem of overfitting).

(33) While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, **humans can generally perform a new language task from only a few examples** [...] — **something which current NLP systems still largely struggle to do**. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the art fine-tuning approaches. (Brown, *et al*. 2020)

However, it is not clear whether these systems do anything more with the upscaled parameter size other than simply memorize more. A substantial downside of these scaled-up systems is in terms of the computational resources required to perform the training. GPT-3 is reputed to have cost around $4.6 M to train.[27] This has resulted in a curious admission in the case of GPT-3 (by the authors).

(34) Unfortunately, a bug resulted in only partial removal of all detected overlaps from the training data. Due to the cost of training, it wasn't feasible to retrain the model. (Brown, *et al*. 2020)

Diminishing returns are another (expected) negative factor.

(35) To halve the error rate, you can expect to need more than 500 times the computational resources. (Thomson, *et al*. 2021)

The enormous resources required, both in terms of energy and exposure to large amount of data, means that these systems, independent of their potential

---

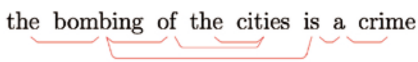
[27] According to <https://lambdalabs.com/blog/demystifying-gpt-3/>, training GPT-3 would cost over $ 4.6 M using a Tesla V100 cloud instance.

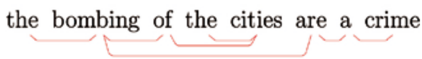
achievements or promise of their biologically-inspired architecture, cannot possibly meet the austere conditions for Chomsky's *Genuine Explanation*, conditions that nature has already met.

Artificial neural networks cannot operate directly on symbolic representations of words, *e.g.* either represented orthographically or as bundles of features. Instead, words must first be transformed into points in n-dimensional space, implemented as a vector of length n (containing floating point numbers), usually via a process called *contextual word embedding*, *e.g.* as in BERT from Google (Devlin, *et al*. 2019).[28] Artificial neurons, being numerical devices, compute directly on these *vectorized words*. Hewitt & Manning (2019), in their structural probe model, make an interesting claim about the information embedded in these vectors, summarized in (36).

(36) The syntax distance hypothesis:
there exists a linear transformation of the word representation space under which vector distance encodes parse trees.

In other words, it should be possible to recover (from vectorized words in the context of a sentence) diagrams similar to the dependency parses of the sort shown earlier in (2) and (11) (but without grammatical relation labels). Structural probe diagrams are given in (37a-b) below for Chomsky's examples (1a-b), respectively.[29]

(37) a.


b.


[28] GPT-2, mentioned earlier, also uses contextual word embedding, but does not use BERT.

[29] The results reported in this paper were obtained using the publicly available code in <https://github.com/john-hewitt/structural-probes> via the default BERTlarge model.

(37a-b) match well with the *Google Natural Language* dependency parse shown previously in (2).[30] In both cases, the NP head *bombing* is the closest preverbal noun to the verbal head *is/are* in terms of link count. Note that *Google Natural Language* requires extensive training on dependency treebanks. By comparison, these BERT contextual word embeddings are learnt from unlabeled large corpora, *i.e.* without visible syntactic structure, and therefore this structural probe model represents a considerable reduction in terms of what knowledge of language the neural architecture needs to be primed with. The tradeoff, of course, is that links remain unlabeled in the structural probe model.[31]

Hewitt (& Manning) go on further to suggest that subject-verb number agreement may also have been learnt: in the case of (38) below, their model will consistently compute *chef* as the closest linked noun to the singular verb form *is*, rather than the plural nouns, *viz.* *stores* or *parents*, despite *chef* being further away in terms of linear distance.

(38) The chef$_{[sg]}$ (who ran to the stores$_{[pl]}$ (and talked to the parents$_{[pl]}$)) is$_{[sg]}$ out of food[32]

However, as (37b) above suggests, this does not necessarily mean that number agreement is responsible for the correct linking. In fact, a hierarchy-respecting link model of number agreement oversimplifies attested data, as there is abundant literature on first or last conjunct agreement, varying across languages and pre- vs. post-verbal order. In the case of English, consider (39a-

30 The only (unlabeled) discrepancy between the Google parse in (2) and the structural probe graphs in (37a-b) concerns the direct link between verbal head *is/are* and the nominal head of the object NP *a crime* or its determiner *a*. Incidentally, in the case of (37a-b), this also means *bombing* is closer (in terms of link distance) to the verbal head than post-verbal *crime*.

31 It seems possible that the relations might be identified, *i.e.* labeled, via a bootstrapping approach.

32 We do not reproduce the diagrams for the examples (38) in this paper. See John Hewitt's blog entry (04 April, 2019) at <https://nlp.stanford.edu//~johnhew//structural-probe.html> instead.

f) below.

(39) a. The boy or his friends run every day

b. *The boy or his friends runs every day

c. The friends or the boy runs every day

d. *The friend or the boys runs every day

e. The friends or the boy runs every day

f. The friends or the boy run every day

English appears to exhibit last (alternatively, linear) conjunct agreement when *or* is the coordinating conjunction, as (39b) and (39d) indicates. This is inconsistently reflected in the structural probe diagrams: (39a) and (39b) have the same links, although (39d) sprouts an extra link from the verb form *runs* to singular *friend*, when compared with (39c). Similarly, (39f) has an extra link from *run* to plural *friends*, when compared with (39e).[33]

5. Simplicity and Efficient Design in Linguistic Models

In this section, we provide motivation for a different view of simplicity. In contrast to the engineering world, the world of Chomskyan linguistics, in pursuit of better explanation, has moved in quite the opposite direction, pruning unnecessary parameterization and fixing core syntax as unlearned, and therefore, part of our genetic endowment. The problem of acquisition of mechanisms of core syntax vanishes.[34] However, the problem of evolutionary plausibility

[33] Note that both (39c) and (39e) also have an extra (but erroneous) link between the two occurrences of *the*.

[34] This is assuming proposed mechanisms are empirically adequate and satisfy

must be taken seriously. We cannot move mechanisms from the to-do acquisition pile and simply stack them on the genetic endowment side. If we assume simple mechanisms are evolutionarily more plausible than complex ones, we should pursue theories containing only the simplest possible mechanisms. On the evolutionary time scale, nature optimizes design in the face of resource (and environmental) challenges. We should also pursue computational minimalism, adhering to general principles of efficiency, as nature itself has done. As a result, much in the way of prior formalisms has been discarded in the pursuit of these goals, including complex transformations, context-free grammars, X´-theory, theory-internal feature systems and category labels, all of which may turn out to be unnecessary machinery. (See Chomsky (2021b) for a summary of the minimalist program of research and a snapshot of the current theory.)

In what follows, we sketch one possible instantiation of recent theory and discuss some consequences. As mentioned earlier, there is evidence that there is no place for linear order in this framework, with all structure being hierarchical in nature, generated by the simplest possible combinatorial operation, *Binary Merge*. Structures produced by Merge are minimalist, represented in the theory as sets, leaving no room for category labels or feature systems (apart from those introduced as heads at the lexical interface). Ideally, Merge simply creates a single new object (and does nothing more), placing it in a *Workspace* (WS) for further possible Merges, and for *Interpretation* (INT) to read when sufficiently fleshed out.[35] An example of a derivation for (40a) is given in (40b). Beginning with a WS consisting only of lexical items drawn from LEX, objects are recursively introduced by application of Merge. (Each Merge step is indi-

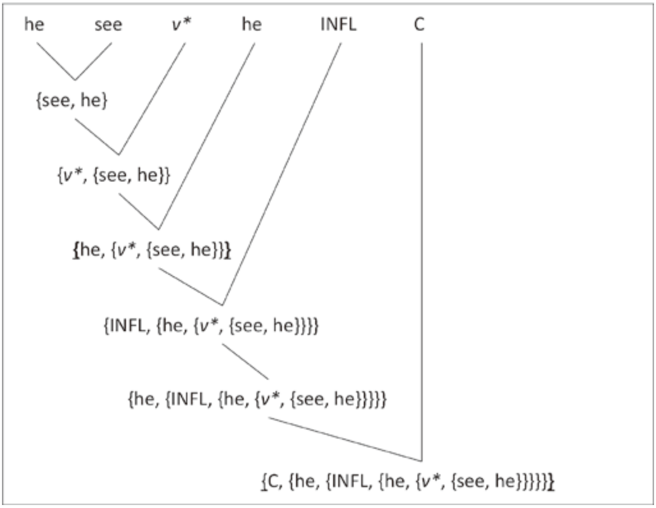
evolutionary plausibility, perhaps a tall order. Additionally, the problem of lexical acquisition still looms large.

[35] It may be the case that INT has only limited access to the results of Merge for general efficiency reasons, *e.g.* minimizing bandwidth between core language and external systems, and also for the reason that structural relations crucial for interpretation, *e.g.* copies, cannot be formed until certain stages in the derivation have been reached, in which predicates and their arguments have been arrayed appropriately, *i.e.* at *Phase* levels in Chomsky's terms. (In (40b), the Phases are specially marked using bold underlined curly braces **{ … }**.)

cated by the lines connecting inputs to a new object.)

(40) a. He saw him

b. WS: he, see, *v**, he, INFL, C[36]



In (40b), we assume Merge forms canonical predicate-argument configurations that can be read by INT. For transitive verbs *v**-Root, this is the Phase {EA, {*v**, {Root, IA}}}, realized as {he, {*v**, {see, he}}} in the case of (40b). Beyond this stage, inflectional and force elements are added to form a complete clause. In English, INFL participates in visible Agreement between the surface subject (in INFL-specifier position) and the verbal complex INFL-*v*-see*. As a result of Agreement, the verb inflects, spelling out as *sees* and *see* in *he sees him* and *I see him*, respectively. The prominent Force feature on C may take on dif-

[36] Initial contents of the WS in (40b) are sourced from LEX. Functional categories *v**, INFL and C represent heads with prominent features for (verbal) transitivity, ϕ (person, number)-Agreement and declarative Force, respectively. Note there are two independent occurrences of the pronoun *he*, as required by verbal *v*-see*.

ferent values, *e.g.* resulting in a declarative clause, as in (40a), or an interrogative clause, as in *what does he see* or *did he see him*.

A possible formalization of Merge is given in (41), with *Binary Search* (Σ_B), part of the toolkit of computationally minimalist devices available to language (and other cognitive systems), to be defined later.

(41) *Binary Merge* (on WS):
Binary Search (Σ_B) WS + LEX for distinct inputs, X and Y.
Add {X, Y} to WS.

In the most bare-bones conception of computation on this WS, selected inputs to Binary Merge are never transformed (or edited in any way) during Merge, they simply become members of a new set (and nothing more) through ***Add*** in (41). This means that Merge obeys a *Non-Tampering Condition* (NTC), as any form of modification would constitute an extra operation. Ideally, inputs are also subject to computational minimalism, *e.g.* perhaps there can be only one (visible) instance of that object. Suppose once an object has been Merged, thereafter, it can only be accessed from within the (new to the WS) larger object. Ambiguity (and complexity) in the selection of that object is eliminated, and the WS is kept to the smallest possible size (in terms of selectable objects). Put concretely, no WS object participates more than once in Merge for the derivation in (40b). In this sense, Binary Merge can be characterized as being Markovian, *i.e.* only the WS going forwards will be accessed by Σ_B, and nothing beforehand.

Unlike the phrase structure trees encoded by CFGs, in which order and hierarchy are intermingled, the sets produced through Merge are conceptually unordered. Encoding of word order differences, *e.g.* head-complement and modifier-modifee order, must be accomplished outside of core syntax, or *I-Language*. Linearization of I-Language structures for speech articulation happens during some process of *Externalization* (EXT) to the sensorimotor system, but need not happen, *i.e.* I-Language need not be externalized at all, an idea that is still radical to many linguists, but places I-Language centrally as the *language of thought*.

Consider the three identical inscriptions for pronominal *he* in (42a) below. (Subscripts are used for expository purposes only.) (42a) is the I-Language structure computed for example (40a) via derivation (40b). Since Merge only creates structure (and nothing else), Merge computes no relations between the inscriptions. However, at INT, different *he*'s may be related, *i.e.* paired up as copies. If paired, we interpret the identical inscriptions as referencing the same individual. This operation Chomsky calls FORMCOPY (FC), which operates at the Phase level.

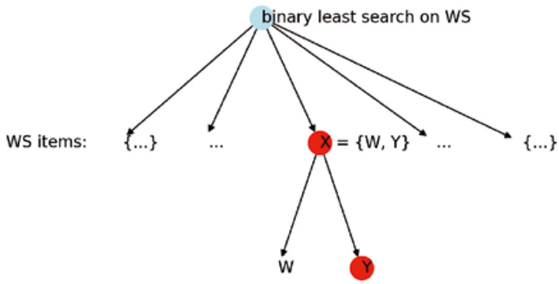
(42) a. $\{C, \{he_1, \{INFL, \{he_2, \{v^*, \{see, he_3\}\}\}\}\}\}$
b. $(he_1, he_2) \mapsto \{C, \{he_1, \{INFL, \{\cancel{he}_2, \{v^*, \{see, he_3\}\}\}\}\}\}$
$\mapsto$ *he saw him*
c. (he_2, he_3) violates *Θ-Theory*, no output
d. (he_1, he_3) unseen by *Binary Search*, no output

FC operates under computational minimalism as an instance of Σ_B, part of the primitive toolkit that underpins Merge, as defined in (41) above. One possibility is that binary search identifies he_1 and he_2 as identical inscriptions that could be paired. This is depicted in (42b). We assume only highest copies (in the case of languages like English) will be pronounced, a rule of economy in externalization; hence $\cancel{he}_2$ will be marked to remain unpronounced when linearized. Assuming nominative and accusative Case are assigned to subjects and objects, respectively, we obtain *he saw him*, pronouncing *he* twice. Suppose instead, FC related he_2 and he_3 as in (42c). Verbal *see* comes with two θ-roles (as part of its lexical entry). We assume these roles must be assigned to independent nouns, call this rule a part of *Θ-Theory*. As (42c) violates Θ-Theory, no output can be formed. Finally, consider the pairing (he_1, he_3) in (42d). This pairing is never considered by FC as search is a computationally minimalist operation, the search down inside (42a) for an inscription identical to he_1 will find he_2 first and stop, never reaching he_3.

Suppose also that Merge is the only operation that can create structure, *i.e.* there are no other set-formation operations in the theory. Then one possible story is that, about 200,000-300,000 years ago, Merge, the simplest combinatorial operation, appeared at the heart of the most recent inflexion point in the evolution of our species, giving rise to rich and complex language that we know is possessed only by modern humans. The strongest version of this story also poses a severe challenge for perception. If only Merge emerged, by what mechanism do we assemble the sequence of words that we hear? We surmise from psycholinguistics that the human brain (apparently) does not wait until the end of an utterance to begin building structure. Did another brand-new mechanism co-evolve along with Merge, or did we adapt something pre-existing to the task? (Contrast this situation with a theory based on CFGs, for which no corresponding mystery exists, and algorithmic parsing methods that require no explanation are straightforwardly available.) Similarly, in the case of output, what mechanisms permit the reading of these (new to the species) structures? And how are they mapped onto a sequence of sensorimotor instructions? The conditions for a *Genuine Explanation* in these cases raise difficult new questions.

Binary Search (Σ_B) is the minimalist procedure at the heart of *Binary Merge*. For efficiency, search will do the least amount of work, *i.e.* find the first item that meets a targeted object and stop. Suppose we define Σ_B for Merge as follows:

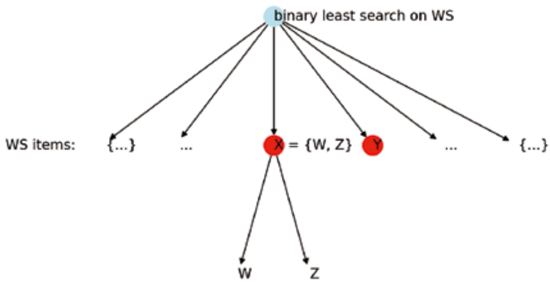
(43) Binary search (Σ_B) for Merge returns two objects, X and Y.
Σ_B on WS + LEX first finds an item X $\in$ LEX or X $\in$ WS.
(This choice point is represented by the top circle in the diagram below.)



Suppose $X \in$ WS is selected, to find Y, Σ_B **continues** the search within X, as illustrated above. This secondary search is the simplest way to find a distinct Y. Y will be a sub-term of X.[37] Call this case *Internal Merge* (IM). (This option is not available when $X \in$ LEX is selected, as a lexical item X has no internal structure.)

Alternatively, Σ_B **pops up** to the top choice point (originally, for X), and selects a different item from the WS, *viz*. Y, as illustrated below. Call this case *External Merge* (EM).

(44)



External Merge is conceptually necessary to form hierarchical objects in general. For language, EM computes predicate-argument structure. *Internal Merge*

[37] Recall that syntactic objects are sets. A sub-term of a syntactic object SO is a member of SO or a sub-term of a member of SO.

(IM) encodes displacement and is needed for discourse/information-related functions in language.

EM alone builds the Phase {EA, {*v**, {Root, IA}}}. In (40b), EA = *he*, Root = *see* + PAST tense feature, IA = *he*. IM promotes the closest argument to the surface subject position, *viz*. EA *he* in (40b), forming {he, {INFL, {he, {*v**, {see_{PAST}, he}}}}}. IM may also promote IA to surface subject, making the IA more prominent, as the case of simple passives, *e.g. he was seen*, or even arguments of an embedded clause, as in *he was thought to have won the race* or *the race was thought to be have been abandoned*. Promotion to surface subject need not always occur though, *e.g.* as in *there arrived a man* or *it was thought that he had won*. The definition of binary set Merge in (41) covers both functions, intended to suggest that these two are packaged together. With respect to the evolutionary emergence of Merge, an interesting possibility, suggested by the definition of Σ_B in (43), is that IM could be the result of a small optimization in search, *i.e.* nature optimized search by happening upon the search continuation (that permits IM).[38]

6. Efficient Computation in Linguistic Models

In the previous section, we have seen how theory selection can be guided by principles such as simplicity and computational minimalism. Merge, as an abstract recursive operation, is combinatorially explosive. For instance, given an initial WS with just two elements *a* and *b*, (45b) through (45d) lists the *different* structures that can be formed after 1 to 3 rounds of Set Merge, respectively.

[38] Chomsky (2008) observes that iterated application of a reduced form of IM over just one lexical item can yield the successor function, a primitive in the Peano axiomatization of arithmetic over the natural numbers. Perhaps then, arithmetic and language draw from the same toolkit perhaps available to all human cognitive systems.

(45) a. initial WS: a b
b. after 1 round: {a, b}
c. after 2 rounds: {a, {a, b}}, {b, {a, b}}
d. after 3 rounds: {{a, b}, {a, {a, b}}}, {b, {a, {a, b}}}, {a, {a, {a, b}}}, {{a, b}, {b, {a, b}}},{b, {b, {a, b}}}, {a, {b, {a, b}}} and so on

Note that the set {a, {a, {a, b}}} in (45d) (highlighted in red) can be formed from {a, {a, b}} in (45c) (also highlighted in red) in two different ways through IM. This is illustrated in (46a-c) below (indices supplied for expository reasons only).

(46) a. $\{a_1, \{a_2, b\}\}$
b. $\{a_1, \{a_1, \{a_2, b\}\}\}$
c. $\{a_2, \{a_1, \{a_2, b\}\}\}$

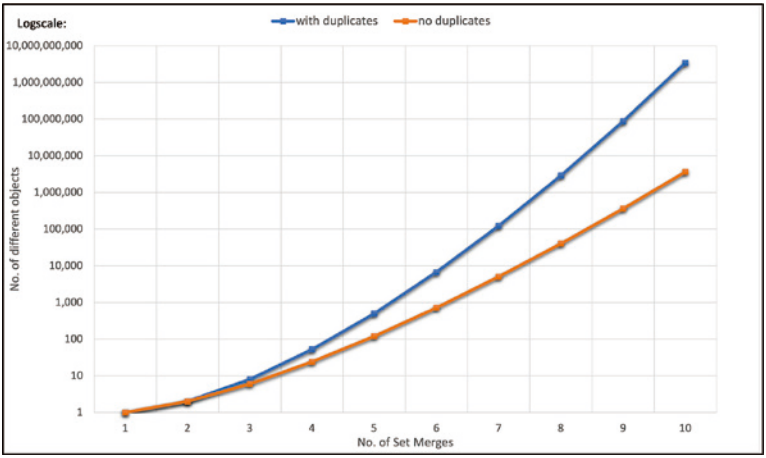
As the WS is merely a list of objects, not itself a set object, the list of structures counting duplicates grows even faster than shown in (45a-d). A possible candidate for a principle of efficient computation is a duplicate-free computational space. However, to produce (45d), which is duplicate-free, a theorem proving system would have to spot (and eliminate) already-seen structures through some sort of memory device.[39] But worrying about duplicates is not combinatorially fruitful because the number of possible structures grows faster than any exponential, whether we count duplicates separately or not, as the logscale graph in (47) indicates.[40]

---

39 For instance, a global table of unique structures could be maintained by the theorem prover, and be consulted by each (normally independent) thread of computation. The general form of this optimization technique is called *memoization*.

40 The number of possible structures grows as $n!$, n being the number of merge steps. The proof is simple. Suppose IM selects X and has a choice of k sub-terms (of X) to select as Y. In the next step, it will have $k + 1$ choices for Y, $+ 1$ because X will become a sub-term of the new object formed.

(47)



Not only is it not fruitful to worry about duplicates, Chomsky (p.c.) also observes that WS duplicates must occur naturally. Consider the separate occurrences of *saw many people* in (48) below.

(48) the man who saw many people saw many people

(48) cannot be derived from WS_1, (= 49a) below, but instead requires two (independent) instances of {v*, {see, many people}}, as in WS_2, (49b).

(49) a. WS_1: {v*, {see, many people}}, {who man}, INFL, C_{rel}, the, INFL, C
b. WS_2: {v*, {see, many people}}, {who man}, INFL, C_{rel}, the, {v*, {see, many people}}, INFL, C

The continuation of (49a), using a movement theory of relative clause formation, as shown in (50a) and onwards, runs into trouble at step (50e).[41]

[41] Strikethrough is used to indicate copies that will not be pronounced. There are many possible theories of relative clause formation, and this one was selected to illustrate how deep in the phrase (equivalently, how far back in history) the

(50) a. {who man, {v*, {see, many people}}}, INFL, C_{rel}, the, INFL, C
b. {who man, {INFL, {~~who man~~, {v*, {see, many people}}}}}, C_{rel}, the, INFL, C
d. {C_{rel}, {who man, {INFL, {~~who man~~, {v*, {see, many people}}}}}}, the, INFL, C
e. {who man, {C_{rel}, {~~who man~~, {INFL, {~~who man~~, {v*, {see, many people}}}}}}}, the, INFL, C
f. {the, {man, {who ~~man~~, {C_{rel}, {~~who man~~, {INFL, {~~who man~~, {v*, {see, many people}}}}}}}}}, INFL, C

The WS object in (50e) corresponding to *the man who saw many people* must be Merged to a θ-position using EM (by Duality), but {v*, {see, many people}} is not a WS object. Moreover, it is too deeply embedded to be even accessed by IM (even if Duality could be suspended).[42] A (special) Merge operation that could fish out the required phrase seems too implausible on several levels.

In any case, the system sketched above is clearly computationally implausible as it will quickly exceed all computational resources, biological or artificial. Let us turn then to consider two kinds of constraints on Merge, one sort is language organ-specific, Chomsky calls them *Language Specific Conditions* (LSCs), and the other is general to computation. We have already mentioned one example of the general sort, *i.e*. the elimination of duplicate structures such as (46b) and (46c) via a memory device. But no new device need be required (and justified) if we assume some form of minimal search. If a_1 and a_2 in (46a) are identical inscriptions, all that is required is for Σ_B, defined in (43), to find a_1 (when looking for *Y*) and stop, never to encounter a_2. This situation is only logically possible when Σ_B is searching for some feature that both a_1 and a_2

necessary Merge companion might be. (Details: assume C_{rel} is a relativizing C head that targets an argument with a relative pronoun through IM, and that the relative pronoun is part of the noun phrase *who man*.)

42 In Chomsky's Phase theory, the internals of a Phase, a phrase headed by a Phase head, *viz*. C or v*, are not accessible to IM.

happen to possess (and since they are identical, a_1 must hide a_2).[43] Whether minimal feature search is an example of a LSC or general now becomes an interesting question for future work.

Another candidate for a general principle of computational efficiency is one that blocks useless repetitive operations, perhaps justified by some sort of energy conservation principle for biological systems. For instance, we can continue to grow (46a), repeated as (51a), indefinitely, as illustrated in steps (51b-d).

(51) a. $\{\underline{a_1}, \{a_2, b\}\}$
b. $\{a_1, \{\underline{a_1}, \{a_2, b\}\}\}$
c. $\{a_1, \{\underline{a_1}, \{a_1, \{a_2, b\}\}\}\}$
d. $\{a_1, \{\underline{a_1}, \{a_1, \{a_1, \{a_2, b\}\}\}\}\}$ and so on

In (51a-d), IM selects the highest a to be raised (selected a is highlighted in red). Another kind of repetitive pattern is given in (52a-e), in which IM alternately selects the highest b then highest a, repeating indefinitely.

(52) a. $\{a, \underline{b}\}$
b. $\{b, \{\underline{a}, b\}\}$
c. $\{a, \{\underline{b}, \{a, b\}\}\}$
d. $\{b, \{\underline{a}, \{b, \{a, b\}\}\}\}$
e. $\{a, \{\underline{b}, \{a, \{b, \{a, b\}\}\}\}\}$ and so on

If we admit extra machinery to check the output of Merge, it is possible to block such repetitions by detecting (and blocking) a repeated series of operations as soon as it occurs, formalized in condition (53).

[43] There is a trap that we must avoid in thinking about duplicates a_1 and a_2. The trick is to not come up with a proposal that needs to compare them. If the search for Y is not a search for a feature, but just the sub-term-of operation, *i.e*. return a sub-term (any sub-term), we would need a table to check found items against previously-found ones.

(53) $*\pi\pi$, where $\pi = (\text{IM } O_1, \ldots, \text{IM } O_n)$, $n \geq 1$

The tuple $(\text{IM } O_1, \ldots, \text{IM } O_n)$ represents a contiguous sequence of IM operations, each time raising some object O_i, for i in range $1 \ldots n$. If a recorded pattern π occurs twice in a row, *i.e.* $\pi\pi$, it is blocked, therefore ruling out (51) and (52). However, there are several problems with this approach. One is conceptual: pattern recognition requires access to (and storage of) history of Merge operations, but that permits circumvention of the Markovian assumption for Merge. Another problem is empirical coverage, despite possessing a memory device, (53) cannot block repetitions of the sort illustrated in (54) below.[44]

(54) a. {a, b}
 b. {b, {a, b}}
 c. {{a, b}, {b, {a, b}}}
 d. {{b, {a, b}}, {{a, b}, {b, {a, b}}}}
 e. {{{a, b}, {b, {a, b}}}, {{b, {a, b}}, {{a, b}, {b, {a, b}}}}} and so on

In (54), IM always selects the member of the set representing the entire syntactic object in the previous round.[45] Perhaps this kind of *infinite looping* is not a problem in practice. It is certainly not a problem for formalisms in which each IM operation must be driven by feature checking, *e.g.* in the *Minimalist Grammar* (MG) framework of Stabler (1997).[46] Recall from (43-44) that there is also the idea of a strict division of labor between EM and IM, something Chomsky

44 In fact, depending on how one defines 'repetition,' generally there are infinitely many kinds of repetitions possible.

45 This statement is true, except, in the case of the very first round.

46 However, MGs, being feature-driven, generally require the invention of many formal features to drive IM, and, as a result, lexical entries can proliferate. The proliferation of formal features raises serious questions for acquisition in this framework. One of the goals of the Minimalist Program is to eliminate as many theory-internal features as possible. The MG formalism also shares some of the disadvantages of CFGs, including the intermingling of linear order and hierarchy.

terms the *Duality of Semantics*.[47] This will limit (to some degree) the possible application of IM: consider the possible Merge operations in (55) below. (Note that (55) partially repeats the general form of the initial steps in example (40b), using IA and EA to stand for the *Internal* and *External* argument of a transitive verb v*-R.)

(55) a. {R, IA$_{\theta}$} (EM)
b. {v*, {R, IA$_{\theta}$}} (EM)
c. i. {EA$_{\theta}$, {v*, {R, IA$_{\theta}$}}} (EM)
ii. *{IA$_{\theta}$, {v*, {R, IA$_{\theta}$}}} (IM)
d. {IA, {EA$_{\theta}$, {v*, {R, IA$_{\theta}$}}}} (IM)
e. {IA, {IA, {EA$_{\theta}$, {v*, {R, IA$_{\theta}$}}}}} (IM) and so on

In (55a), a verbal root *R* and *IA* are combined. In the initial step, only EM is a possible Merge, and a θ-configuration is formed, in which *IA* occupies a θ-position (indicated by the θ-subscript). According to Duality, only EM can Merge to θ-positions. From (55b), there are two possibilities under abstract Merge, (55c-i) and (55c-ii). In (55c-i), Duality is respected, as EM introduces *EA* to a θ-position (indicated by EA$_{\theta}$). In (55c-ii), IM raises IA to occupy a θ-position. This operation, although arguably simpler than (55c-i) from the definition of Σ_B in (43), is blocked under Duality, as only EM can Merge to a θ-position. Therefore, Θ-Theory filters the possibilities shown in (55c). Note that IM can raise IA to a non-θ-position (without violating Duality), as in (55d), a possible continuation of (55c-i). In fact, this displacement is required in the case of object *wh*-fronting, *e.g.* as in *what did John see*, as (55c-i) is a Phase, and only the edge of a Phase will be accessible to further operations.[48] However,

[47] Duality segregates language operations. It limits Θ-Theory to EM, *e.g.* only EM can introduce a syntactic object into a θ-position. It is also an optimization step, reducing the number of options. Consider, for example, {XP, {v*, {R, XP}}}, in which the higher *XP* could have been added either through EM of *XP* or IM of the lower *XP*. Duality limits this to EM, as the higher *XP* position is a θ-position.

[48] The *edge* of this Phase is any object higher than the Phase head v*, *i.e.* *EA* and *IA* in (55d), but before the introduction of a higher head, *e.g.* INFL in (40b). *IA* needs to be

what is to prevent an infinite loop being introduced (by IM) as in (55e)? One could also argue that (55a), repeated below as (56a), could be continued as in (56b-c).

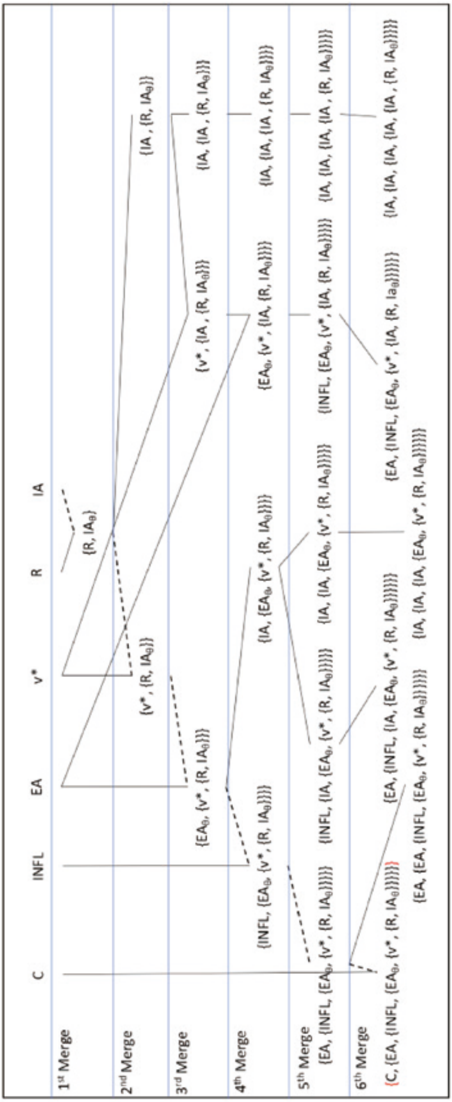
(56) a. $\{R, IA_\theta\}$ (EM)
b. $\{IA, \{R, IA_\theta\}\}$ (IM)
c. $\{IA, \{IA, \{R, IA_\theta\}\}\}$ (IM) and so on

(56b) is the case of *Object Shift* (OS), a phenomenon visible in North Germanic languages. However, (56c) explodes that into an infinite loop that gaps (55b) from (55a). Note that in both (55e) and (56c), Minimal Search is respected, as only the highest *IA* is accessed in each case. Duality is also respected, as IM is not introducing a θ-position either. As both infinite loops occur inside a Phase, that concept of a natural "chunk" of computation does no work here either.

Let us consider two possible scenarios to resolve this combinatorial problem. First, suppose we let all possible derivations operate in parallel. (Assume, for the moment, that sufficient resources exist for such a scenario.) Any computation unrolling an infinite loop (even partially) is at a significant disadvantage with respect to the race to Phase level to be interpreted. Suppose a derivation gets stuck in a loop, then INT simply never happens to that thread of computation.

at the edge as it must be targeted by C_Q, the interrogative complementizer, for displacement to the edge of the clause.

(57)


C
INFL
EA
v*
R
IA
1st Merge
{R, IAθ}
2nd Merge
{v*, {R, IAθ}}
{IA , {R, IAθ}}
3rd Merge
{EAθ, {v*, {R, IAθ}}}
{v*, {IA , {R, IAθ}}}
{IA, {IA , {R, IAθ}}}
4th Merge
{INFL, {EAθ, {v*, {R, IAθ}}}}
{IA, {EAθ, {v*, {R, IAθ}}}}
{EAθ, {v*, {IA, {R, IAθ}}}}
{IA, {IA, {IA , {R, IAθ}}}}
5th Merge
{EA, {INFL, {EAθ, {v*, {R, IAθ}}}}}
{INFL, {IA, {EAθ, {v*, {R, IAθ}}}}}
{IA, {IA, {EAθ, {v*, {R, IAθ}}}}}
{INFL, {EAθ, {v*, {IA, {R, IAθ}}}}}
{IA, {IA, {IA, {IA , {R, IAθ}}}}}
6th Merge
{C, {EA, {INFL, {EAθ, {v*, {R, IAθ}}}}}}
{EA, {INFL, {IA, {EAθ, {v*, {R, IAθ}}}}}}
{EA, {INFL, {EAθ, {v*, {IA, {R, Iaθ}}}}}}
{IA, {IA, {IA, {IA, {IA , {R, IAθ}}}}}}
{EA, {EA, {INFL, {EAθ, {v*, {R, IAθ}}}}}}
{IA, {IA, {IA, {EAθ, {v*, {R, IAθ}}}}}}

To make the comparison explicit, consider the derivational graph in (57).[49] By the 6th Merge, only one derivation, the intended one (indicated by the dotted lines), has reached the CP Phase level necessary for interpretation. The other derivations shown are incomplete and will be delayed, possibly indefinitely. However, this scenario is inefficient in terms of resources consumed, *i.e.* it's possible that the intended derivation contains only a small fraction of the number of Merges made overall. An alternative scenario is to apply the following condition, a candidate principle of efficient computation:

(58) Never consecutively repeat the same operation

Note that the implementation of (58) does not require unbounded memory, only the ability to recall a single step in the history of derivation, and therefore is considerably more resource-friendly than the condition in (53). In the case of the 6th step in (57), we obtain (59a-c) as the surviving derivations filtered by (58).[50]

(59) a. $\{$C, $\{$EA, $\{$INFL, $\{$EA$_\theta$, $\{v*$, $\{$R, IA$_\theta\}\}\}\}\}\}$
b. $\{$EA, $\{$INFL, $\{$IA, $\{$EA$_\theta$, $\{v*$, $\{$R, IA$_\theta\}\}\}\}\}\}$
c. $\{$EA, $\{$INFL, $\{$EA$_\theta$, $\{v*$, $\{$IA, $\{$R, IA$_\theta\}\}\}\}\}\}$

(59a) represents the converged syntactic object for a simple transitive sentence, as computed concretely in (40b) for the case of example (40a). (59b) leads to a valid derivation generally for *wh*-object questions, shown in (60a-c).

49 For ease of exposition, the graph in (57) represents a simplified all-possibilities model. IM only targets θ-relevant objects IA and EA in this model. We have also deliberately omitted some other possible threads of computation that do not converge, including ones involving *Pair Merge*, which we have not discussed in this paper. (See also note 52.)

50 Implicit in the sequence of operations leading to (59a-c) are some assumptions about Φ-feature computation and what objects may promote to surface subject position, *viz*. specifier of INFL, given Search. We have also not discussed Labeling, a restrictive operation at the Phase level that filters out uninterpretable structures.

(60) a. $\{EA, \{INFL, \{IA, \{EA_\theta, \{v*, \{R, IA_\theta\}\}\}\}\}\}$ (= 59b)

b. $\{C_Q, \{EA, \{INFL, \{IA, \{EA_\theta, \{v*, \{R, IA_\theta\}\}\}\}\}\}\}$

c. $\{IA, \{C_Q, \{EA, \{INFL, \{IA, \{EA_\theta, \{v*, \{R, IA_\theta\}\}\}\}\}\}\}\}$

Finally, (59c) represents the general OS case, converging as (61b).[51]

(61) a. $\{EA, \{INFL, \{EA_\theta, \{v*, \{IA, \{R, IA_\theta\}\}\}\}\}\}$ (= 59c)

b. $\{C, \{EA, \{INFL, \{EA_\theta, \{v*, \{IA, \{R, IA_\theta\}\}\}\}\}\}\}$

Apart from the three survivors in (60a-c), all shown to be attested in language, the other derivations in (57) all involve a violation of (58) at some stage, and thus are eliminated. We speculate that more complex cases of repetition, *e.g*. (54), may never occur in practice. If so, the complex mechanisms required to spot and eliminate these cases will not be needed. Nature has perhaps happened upon a simple solution in this particular case.

Much work remains in order to determine whether Merge is computationally plausible in a resource-limited setting. The simplified derivational graph in (57) illustrates several threads that must be explored in the search for a convergent derivation. However, in (57) there is the undocumented possibility of applying Pair-Merge, a variant of Merge that may also be conceptually necessary, at each step.[52] Despite being constrained by Θ-Theory, a LSC, and gener-

---

[51] Across Scandinavian languages, there is considerable variation in whether OS obtains, *e.g*. full noun phrases, as in the case of Icelandic, or (weak) pronouns, as in the case of most Mainland Scandinavian languages. Many factors appear to be in play, and the phenomenon seems unexplained to a large extent. Note that (59a) and (60b) are substantially the same syntactic object. Perhaps both I-Language structures are co-generated, but one or other may be eliminated during *Externalization* due to language-particular factors.

[52] Pair-Merge produces the asymmetric ordered-pair $<A, B>$, given syntactic objects A and B. In pair $<A, B>$, one of A or B is defined as being inaccessible to Σ_B, rendering extraction impossible. As Chomsky (2004) states, it is an empirical fact that there is also an asymmetric operation of adjunction. With respect to the (in)extractability of adjuncts (or parts of adjuncts), the facts are complicated, and various mechanisms have been proposed. See also relevant discussion of adjunction in Toyoshima (2022).

al computational minimalism, *e.g.* as in (53) or (58), to plug theory-internal gaps such as certain types of repetitive (or vacuous) movement, it is possible that Merge may still be under-constrained. In fact, it is possible to have a combinatorially explosive derivation graph despite only having one convergent outcome for a starting WS.[53] For instance, diagram (63) illustrates the derivation of a simple NP *the book*, (62a), beginning with WS (62b), and converging as (62c), based on the NP theory outlined in Oishi (2015).

(62) a. the book
b. WS: the, *d*, book, *n*
c. $<\{d, \text{the}\}, \{n, \text{book}\}>$[54]

(63)

LIs:[book,n!case,[d,the]] Derivation #1			
Step	**Branch**	**Op**	**SO**
1	-	-	book
2	1	esm	{book,n!case}
3	2	dws	d
4	1	esm	{d,the}
5	3	uws	{book,n!case}
6	1	epm	$<\{d,the\},\{book,n!case\}>$
Spellout heads: [the,book]			
Final output: [the,book]			

There is only one licit derivation in this model, taking 6 steps, as shown in (63).[55] However, the total number of steps explored is much larger, as the full

[53] Chomsky (p.c.) suggests that cases of non-A movement such (55e) and (56c) could be blocked by requiring non-A movement to occur only at/between Phase levels. This would be an example of another LSC.

[54] For this implementation, in $<A, B>$, we define *A* to be the inaccessible adjunct. See note 52.

[55] In (63-65), the primitive operations are identified as *esm* (External Set-Merge), *ism* (Internal Set-Merge), *epm* (External Pair-Merge), *ipm* (Internal Pair-Merge), *dws* (Down WS) and *uws* (Up WS). The latter two operations permit sub-WS computation and return. *SO* refers to a syntactic object in the WS. *Input* is also part of the WS and holds lexical items that have not yet undergone Merge. Each thread in the derivation is uniquely numbered. For example, the first three lines of the derivation in (64) are 1 *epm*, 2 *epm* and 3 *esm*, indicating that Pair-Merge has been

derivation, spread across (64) and (65), illustrates.[56] Therefore, this particular implementation of Oishi's theory can be considered to be computationally implausible.

---

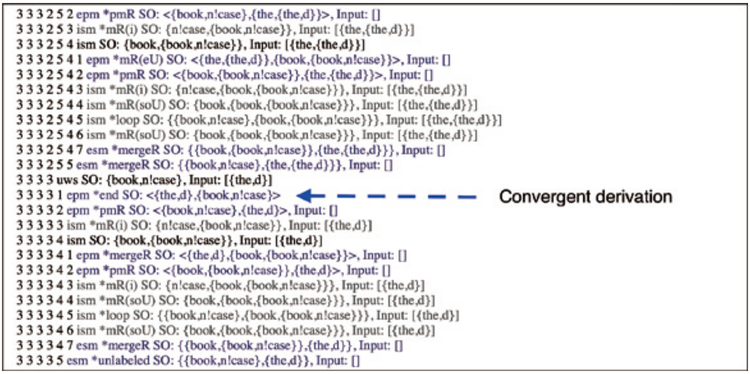
tried twice, as possibilities 1 and 2, and Set-Merge once, as possibility 3. This is because, given two lexical items *X* and *Y*, we can compute <*X*, *Y*>, <*Y*, *X*> or {*X*, *Y*}. Further operations are indicated by extending the sequences of operation numbers. For example, 3 2 in (64) identifies the sequence *esm* (3) followed by *ism* (3 2), resulting in SO {book, {book, *n*}} being formed. *end* indicates the derivation has converged. **mR*/**pmR* indicates that a particular thread of the derivation has been terminated due to violations of Set-Merge and Pair-Merge restrictions, respectively. An example of a *pmR* is adjoining a syntactic object (via *epm*) containing an unvalued feature. Since an adjunct is opaque to search, that feature can never be valued, and so we can rule this option out. **loop* indicates termination due to an identified consecutively repeated operation, *i.e*. the system directly implements constraint (53).

[56] In (64-65), the primitive *ipm* (Internal Pair-Merge) has been removed from the theory.

(64)

```
Example: [book,n!case,[the,d]]
SO: book, Input: [n!case,[the,d]]
1 epm *pmR SO: <n!case,book>, Input: [[the,d]]
2 epm *mR(e) SO: <book,n!case>, Input: [[the,d]]
3 esm SO: {book,n!case}, Input: [[the,d]]
3 1 ism *mR(i) SO: {n!case,{book,n!case}}, Input: [[the,d]]
3 2 ism SO: {book,{book,n!case}}, Input: [[the,d]]
3 2 1 ism *mR(i) SO: {n!case,{book,{book,n!case}}}, Input: [[the,d]]
3 2 2 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [[the,d]]
3 2 3 ism *loop SO: {{book,n!case},{book,{book,n!case}}}, Input: [[the,d]]
3 2 4 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [[the,d]]
3 2 5 dws SO: the, Input: [d]
3 2 5 1 epm *mR(e) SO: <d,the>, Input: []
3 2 5 2 epm *mR(e) SO: <the,d>, Input: []
3 2 5 3 esm SO: {the,d}, Input: []
3 2 5 3 1 ism *mR(i) SO: {d,{the,d}}, Input: []
3 2 5 3 2 ism SO: {the,{the,d}}, Input: []
3 2 5 3 2 1 ism *mR(i) SO: {d,{the,{the,d}}}, Input: []
3 2 5 3 2 2 ism *mR(soU) SO: {the,{the,{the,d}}}, Input: []
3 2 5 3 2 3 ism *loop SO: {{the,d},{the,{the,d}}}, Input: []
3 2 5 3 2 4 ism *mR(soU) SO: {the,{the,{the,d}}}, Input: []
3 2 5 3 2 5 uws SO: {book,{book,n!case}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 1 epm *mR(eU) SO: <{the,{the,d}},{book,{book,n!case}}>, Input: []
3 2 5 3 2 5 2 epm *pmR SO: <{book,{book,n!case}},{the,{the,d}}>, Input: []
3 2 5 3 2 5 3 ism *mR(i) SO: {n!case,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 4 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 ism SO: {{book,n!case},{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 1 epm *mR(eU) SO: <{the,{the,d}},{{book,n!case},{book,{book,n!case}}}>, Input: []
3 2 5 3 2 5 5 2 epm *pmR SO: <{{book,n!case},{book,{book,n!case}}},{the,{the,d}}>, Input: []
3 2 5 3 2 5 5 3 ism *mR(i) SO: {n!case,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 4 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 5 ism *dup SO: {{book,n!case},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 6 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 7 ism *loop SO: {{book,{book,n!case}},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 8 ism *mR(i) SO: {n!case,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 5 9 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 510 ism *loop SO: {{book,n!case},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 511 esm *mergeR SO: {{{book,n!case},{book,{book,n!case}}},{the,{the,d}}}, Input: []
3 2 5 3 2 5 6 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 2 5 3 2 5 7 esm *mergeR SO: {{book,{book,n!case}},{the,{the,d}}}, Input: []
3 2 5 3 3 uws SO: {book,{book,n!case}}, Input: [{the,d}]
3 2 5 3 3 1 epm *mergeR SO: <{the,d},{book,{book,n!case}}>, Input: []
3 2 5 3 3 2 epm *pmR SO: <{book,{book,n!case}},{the,d}>, Input: []
3 2 5 3 3 3 ism *mR(i) SO: {n!case,{book,{book,n!case}}}, Input: [{the,d}]
3 2 5 3 3 4 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,d}]
3 2 5 3 3 5 ism SO: {{book,n!case},{book,{book,n!case}}}, Input: [{the,d}]
3 2 5 3 3 5 1 epm *mergeR SO: <{the,d},{{book,n!case},{book,{book,n!case}}}>, Input: []
3 2 5 3 3 5 2 epm *pmR SO: <{{book,n!case},{book,{book,n!case}}},{the,d}>, Input: []
3 2 5 3 3 5 3 ism *mR(i) SO: {n!case,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 4 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 5 ism *dup SO: {{book,n!case},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 6 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 7 ism *loop SO: {{book,{book,n!case}},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 8 ism *mR(i) SO: {n!case,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 5 9 ism *mR(soU) SO: {book,{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 510 ism *loop SO: {{book,n!case},{{book,n!case},{book,{book,n!case}}}}, Input: [{the,d}]
3 2 5 3 3 511 esm *mergeR SO: {{{book,n!case},{book,{book,n!case}}},{the,d}}, Input: []
3 2 5 3 3 6 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,d}]
3 2 5 3 3 7 esm *mergeR SO: {{book,{book,n!case}},{the,d}}, Input: []
3 3 dws SO: the, Input: [d]
3 3 1 epm *mR(e) SO: <d,the>, Input: []
3 3 2 epm *mR(e) SO: <the,d>, Input: []
3 3 3 esm SO: {the,d}, Input: []
3 3 3 1 ism *mR(i) SO: {d,{the,d}}, Input: []
3 3 3 2 ism SO: {the,{the,d}}, Input: []
3 3 3 2 1 ism *mR(i) SO: {d,{the,{the,d}}}, Input: []
3 3 3 2 2 ism *mR(soU) SO: {the,{the,{the,d}}}, Input: []
3 3 3 2 3 ism *loop SO: {{the,d},{the,{the,d}}}, Input: []
3 3 3 2 4 ism *mR(soU) SO: {the,{the,{the,d}}}, Input: []
3 3 3 2 5 uws SO: {book,n!case}, Input: [{the,{the,d}}]
3 3 3 2 5 1 epm *mR(eU) SO: <{the,{the,d}},{book,n!case}>, Input: []
```

(65)



```
3 3 3 2 5 2 epm *pmR SO: <{book,n!case},{the,{the,d}}>, Input: []
3 3 3 2 5 3 ism *mR(i) SO: {n!case,{book,n!case}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 ism SO: {book,{book,n!case}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 1 epm *mR(eU) SO: <{the,{the,d}},{book,{book,n!case}}>, Input: []
3 3 3 2 5 4 2 epm *pmR SO: <{book,{book,n!case}},{the,{the,d}}>, Input: []
3 3 3 2 5 4 3 ism *mR(i) SO: {n!case,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 4 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 5 ism *loop SO: {{book,n!case},{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 6 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,{the,d}}]
3 3 3 2 5 4 7 esm *mergeR SO: {{book,{book,n!case}},{the,{the,d}}}, Input: []
3 3 3 2 5 5 esm *mergeR SO: {{book,n!case},{the,{the,d}}}, Input: []
3 3 3 3 uws SO: {book,n!case}, Input: [{the,d}]
3 3 3 3 1 epm *end SO: <{the,d},{book,n!case}>          Convergent derivation
3 3 3 3 2 epm *pmR SO: <{book,n!case},{the,d}>, Input: []
3 3 3 3 3 ism *mR(i) SO: {n!case,{book,n!case}}, Input: [{the,d}]
3 3 3 3 4 ism SO: {book,{book,n!case}}, Input: [{the,d}]
3 3 3 3 4 1 epm *mergeR SO: <{the,d},{book,{book,n!case}}>, Input: []
3 3 3 3 4 2 epm *pmR SO: <{book,{book,n!case}},{the,d}>, Input: []
3 3 3 3 4 3 ism *mR(i) SO: {n!case,{book,{book,n!case}}}, Input: [{the,d}]
3 3 3 3 4 4 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,d}]
3 3 3 3 4 5 ism *loop SO: {{book,n!case},{book,{book,n!case}}}, Input: [{the,d}]
3 3 3 3 4 6 ism *mR(soU) SO: {book,{book,{book,n!case}}}, Input: [{the,d}]
3 3 3 3 4 7 esm *mergeR SO: {{book,{book,n!case}},{the,d}}, Input: []
3 3 3 3 5 esm *unlabeled SO: {{book,n!case},{the,d}}, Input: []
```

References

Berwick, Robert C. (2018). "Revolutionary Ideas Appear Infrequently," pp. 177-194 in Norbert Hornstein, *et al*. (eds.) *Syntactic Structures After 60 Years*: *The Impact of the Chomskyan Revolution in Linguistics*. Boston: Mouton, De Gruyter.

Bikel, Daniel M. (2004). "Intricacies of Collins' Parsing Model." *Computational Linguistics* **30**: 479-511.

Brown, Tom B., *et al*. (2020). "Language Models are Few-Shot Learners," in H. Larochelle, *et al.* (eds.) *Advances in Neural Information Processing Systems* **33**. [arxiv.org/abs/2005.14165].

Chomsky, Noam (1956). "Three Models for the Description of Language." *IRE Transactions on Information Theory* **2**: 113-124.

Chomsky, Noam (2004). "Beyond Explanatory Adequacy," pp. 104-131 in Adriana Belletti (ed.) *Structures and Beyond*: *The Cartography of Syntactic Structures*, Vol. 3. Oxford: Oxford University Press.

Chomsky, Noam (2008). "On Phases," pp. 133-166 in Robert Freidin, *et al.* (eds.) *Foundational Issues in Linguistic Theory*: *Essays in Honor of Jean-Roger Vergnaud*. Cambridge, MA: MIT Press.

Chomsky, Noam (2021a). "Linguistics Then and Now: Some Personal Reflections." *Annual Review of Linguistics* **7**: 1-11.

Chomsky, Noam (2021b). "Minimalism: Where Are We Now, and Where Can We Hope to Go." *Gengo Kenkyu* **160**: 1-41.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding."

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: *Human Language Technologies* **1**: 4171–4186.

Fong, Sandiway & Robert C. Berwick (2008). "Treebank Parsing and Knowledge of Language: A Cognitive Perspective." *Proceedings of the Annual Conference of the Cognitive Science Society* **30**.

Hewitt, John & Christopher D. Manning (2019). "A Structural Probe for Finding Syntax in Word Representations." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies* **1**: 4129–4138.

Joshi, Aravind K. (1985). "Tree Adjoining Grammars: How Much Context-Sensitivity Is Required to Provide Reasonable Structural Descriptions?" pp. 206–250 in David R. Dowty, *et al.* (eds.) *Natural Language Parsing*: *Psychological, Computational, and Theoretical Perspectives*. Cambridge, UK: Cambridge University Press.

Marcus, Mitchell P., *et al*. (1999). *Treebank-3 LDC99T42.* Philadelphia: Linguistic Data Consortium.

Moro, Andrea (2016). *Impossible Languages*. Cambridge, MA: MIT Press.

Oishi, Masayuki (2015). "The Hunt for a Label," pp. 322–334 in Hiroki Egashira, *et al.* (eds.) *In Untiring Pursuit of Better Alternatives*. Tokyo: Kaitakusha.

Pereira, Fernando (2002). "Formal Grammar and Information Theory: Together Again?" pp. 13–32 in Bruce E. Nevin & Stephen B. Johnson (eds.) *The Legacy of Zellig Harris*: *Language and Information into the 21st Century. Volume 2*: *Mathematics and Computability of Language*. Amsterdam: John Benjamins.

Pollard, Carl & Ivan Sag (1994). *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.

Pratt, Fletcher (1939). *Secret and Urgent*: *The Story of Codes and Ciphers*. Indianapolis: Bobbs-Merrill.

Seabrook, John (2019). "The Next Word: Where Will Predictive Text Take Us?" *The New Yorker*, October 14 2019 issue. <https://www.newyorker.com/magazine/2019/10/14/can-a-machine-learn-to-write-for-the-new-yorker>.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal* **27**: 379–423, 623–656.

Sprouse, Jon, Beracah Yankama, Sagar Indurkhya, Sandiway Fong & Robert C. Berwick (2018). "Colorless Green Ideas Do Sleep Furiously: Gradient Acceptability and the Nature of the Grammar." *The Linguistic Review* **35**: 575–599.

Stabler, Edward (1997). "Derivational Minimalism," pp. 68–95 in Christian Retor (ed.) *Logical Aspects of Computational Linguistics*: *First International Conference, LACL '96, Nancy, France, September 23–25, 1996. Selected Papers*. LNAI 1328. Berlin: Springer.

Thomson, Neil C., Kristjan Greenwald, Keeheon Lee and Gabriel F. Manso (2021). "Deep Learning's Diminishing Returns: The Cost of Improvement is Becoming

Unsustainable." *IEEE Spectrum* September 24th 2021. <https://spectrum.ieee.org/deep-learning-computational-cost>.

Tomita, Masaru (1987). "An Efficient Augmented-Context-Free Parsing Algorithm." *Computational Linguistics* **13**: 31-46.

Toyoshima, Takashi (2022). "A Note on Adjunct(ion), Pair-Merge, and Sequence." *Tohoku Gakuin University Journal of Institute for Research in English Language and Literature* **46**: 49-67. (This volume).

Wakefield, Jane (2019). "'Dangerous' AI Offers to Write Fake News." *BBC News*. <https://www.bbc.com/news/technology-49446729>.

A Note on Adjunc(tion), Pair-Merge, and Sequence

Takashi Toyoshima

1. Introduction

In traditional grammars, 'modifiers' are standardly deemed optional, in contrast to 'subject' or 'object,' both of which are archetypally obligatory though the latter may as well be optional sometimes. Canonically subjects and objects are noun phrases or clauses, and sometimes prepositional phrases for the former, whereas modifiers are adjectives, adverbials, or prepositional phrases. Intransitive verbs are so defined as they do not allow a noun phrase object, and yet some intransitive verbs require an adverbial or a prepositional phrase.

(1) a. The boy behaved *(politely/like a gentleman/in a good manner).
    b. Our baby is lying *(peacefully/in comfort/there/on the couch).

Other than the reflexive middle or imperative usage, some kind of manner expression is obligatory for *behave*, and a manner or locative expression is necessary for *lie*.

Some transitive verbs also require an adverbial or a prepositional phrase.

(2) a. Jill put the key *(here/in the box).
    b. The company placed profit *(uppermost/above safety).

Given the potential optionality of objects and the obligatoriness of modifiers for some verbs, the distinction between objects and modifiers is not so clear-cut as has generally been thought. Yet, in the current framework of the Minimalist Program in generative grammar (Chomsky 2000, *et seqq.*), adjunct(ion) structures are generated by an operation distinct from the one for the head-complement structure.

This squib is just meant to be a personal memorandum for organizing my current thoughts, questions, their backgrounds, and ideas for possible directions in future research, so no solution or novel analysis is intended to be developed here. We begin by surveying the developmental history of adjunct(ion) structures and how they are created, noting, along with it, obscurities and issues that deserve but had escaped attention in the mainstream literature. The final section offers an outlook for problems to be solved.

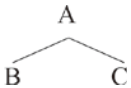
2. A Brief Selective History of Adjunct(ion)

2.1. Before Minimalism

In the early days of generative grammar, subjects and objects are regarded as arguments of a predicate head, coded in the subcategorization frame of the head in its lexical entry whereas modifiers are called adjuncts, specified with regard to their potential categories and positions in the phrase structure rules. Elementary transformations that move or reorder constituents are classified into two sorts: substitution and adjunction. In the former, phrase structure rules generate a structure with a dummy symbol, which some transformational operation replaces with some other constituent in the structure. The latter adjoins a constituent to some other node in the structure, creating a new branch. Three types of adjunction are recognized: daughter-adjunction, sister-adjunction, and ancestor-adjunction or better known as Chomsky-adjunction.

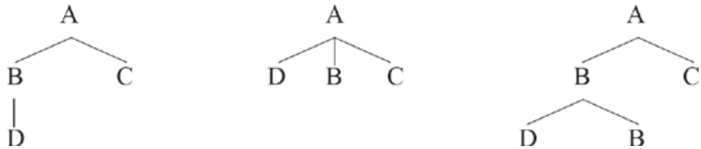
Given the structure as in (3):

(3)



daughter-adjunction of D to B derives the structure in (4a), sister-adjunction of D to the left of B yields (4b), and Chomsky-adjunction of D to the left of B, (4c).

(4) a. b. c.



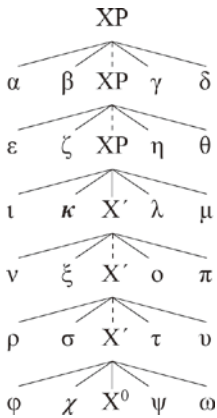
Empirical motivation for daughter-adjunction was scarce (Ross 1967), and sister-adjunction of D to the left of B is indistinguishable from daughter-adjunction of D to A and later considered as a sub-case of substitution (Bach 1974). Only Chomsky-adjunction has survived after X'-Theory supplanted the phrase structure rules. Chomsky-adjunction 'creates' or 'splits' a node with the same label of the host; the newly created node inherits the category information and projection level of the host, and the host and the newly created nodes together were later regarded as 'segments' of a single 'categorial projection' (May 1985, Chomsky 1986, *inter alia*).[1]

The general X'-schema standardly accepted recognizes three levels of projection: X^0, X', and XP, and the latter two levels can potentially recur (5a, c) to host adjunct(ion)s.

[1] See Chametzky (1994) for the problem of labeling for the newly created node by Chomsky-adjunction, and also Hornstein & Nunes (2008) for its exploitation in the theory of Bare Phrase Structure (BPS) of Chomsky (1995a, b).

(5) a. $XP \rightarrow YP^{*}\ XP\ ZP^{*}$
b. $XP \rightarrow YP^{*}\ X'\ ZP^{*}$
c. $X' \rightarrow YP^{*}\ X'\ ZP^{*}$
d. $X' \rightarrow YP^{*}\ X^{0}\ ZP^{*}$

(6)



In the X´-tree diagram (6), all the sisters to X^0 (φ, χ, ψ, ω) are complements, the left innermost daughter to the lowest XP, ***κ***, is the specifier, and all others (α, β, γ, δ, ε, ζ, η, θ, ι, λ, μ, ν, ξ, ο, π, ρ, σ, τ, υ) are adjuncts. Depending on the language and the category, complements may be restricted to appear either only to the left or right of the head X^0 (head parameter), and the specifier may be thought of as some kind of special adjunct that has a special relationship with the head X^0 (spec-head agreement), as suggested in Kuroda (1988).[2]

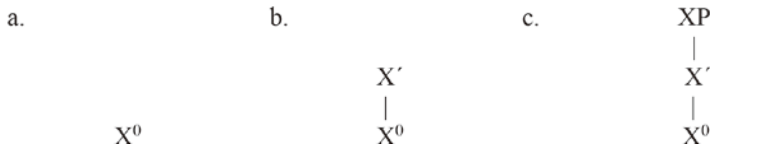
More important is the fact that whereas the recurred XPs are considered as ‘segments’ of a single XP ‘category’ projection, it has never, to the best of my knowledge, been talked about whether or not the recurred X´s are also ‘segments’ of a single X´ level of an intermediate projection, and if so, what the difference is between the lowest ‘segment’ of XP and the highest ‘segment’ of X´.

2.2. From the Inception of Minimalism until Recently

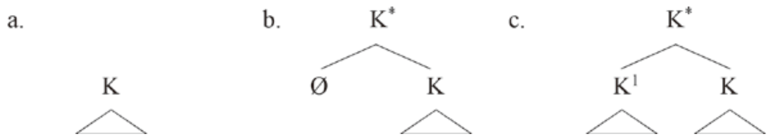
In the initial exposition of the Minimalist Program, Chomsky (1993) aban-

[2] Object vs. subject/adjunct asymmetry subsumed under the Condition on Extraction Domain (CED: Huang 1982) renders further support for Kuroda’s view that the specifier is an adjunct with a special relation to the head.

doned non-interface levels of structural representations, such as D-Structure or S-Structure, and instead proposed that syntactic structures are built up incrementally in the course of derivation by recursive application of a single transformation, Generalized Transformation (GT), revived with revision from Chomsky (1955, 1957). The computational system of human language C_{HL} first selects a lexical item X^0 and projects it to an X´-kernel as the derivation proceeds (*cf*. Project α of Speas 1990).

(7)


GT targets a phrase marker K (8a), 'adjoins' an empty symbol Ø to K, forming K^* (8b), and substitutes another phrase marker K^1 for Ø (8c), conforming to X´-schemata.

(8)


When K^1 was built separately from K and K^*, it is equivalent to base-generation; a unary application of GT to K, copying K^1 from within K, results in movement. In either case, the empty symbol Ø is 'adjoined' to the host K, 'extending' the latter to K^*, which is called the Extension Requirement. Thus, each step of structure-building involved 'adjunction' of Ø and 'substitution' to Ø. Yet, only when K and K^* are the same level of a projection, they are 'segments' of the same category, constituting an adjunct(ion) structure.

Dispensing with X´-schemata of any kind, including the notion of three-leveled X´-kernel, Chomsky (1995a, b) proposed the theory of Bare Phrase Structure (BPS), in which levels of projections are not marked but derivatively read out of structural configurations, following Muysken (1982).

(9) Given a phrase marker, a category that does not project any further is a maximal projection XP, and one that is not a projection at all is a minimal projection X^0; any other is an X´, invisible at the interface and for computation. (Chomsky 1995b, 61f.)

There are no spurious non-branching projections, and a minimal projection, X^0 in the traditional X´-theory, can be maximal at the same time, equivalent to the X´-theoretic non-branching XP. Unlike in X´-Theory, multiple specifiers are allowed in principle, and no linear order is implied among constituents.

Structures are built up step-by-step in the course of derivation by *Merge* or *Move*, two structure-building operations decomposed of GT, and there is no 'specifier' in the traditional X´-theoretic sense as one does not exist before some constituent is merged or moved to a maximal projection which projects further, turning itself into non-maximal. Thus, no movement can be 'substitution' in the traditional sense, and both *Merge* and *Move* are 'adjunction' operations to a host (*cf.* Kayne 1994). The distinction between 'substitution' and 'adjunct(ion)' is made in terms of how the host projects: if the host projects to a higher level, it is equivalent to 'substitution' whereas if a 'segment' is projected, it is equivalent to 'adjunct(ion).' A maximal projection becomes non-maximal when it projects further by merging or moving another constituent to it in the course of derivation.

In the incipient BPS, structures are represented with a labeled set: *Merge* or *Move* takes two syntactic objects α and β, and forms $\gamma = \{\delta, \{\alpha, \beta\}\}$, where δ is the label of γ, indicating the head of γ. The label δ is either the head of α or of β, whichever projects. Syntactic objects are lexical items or phrasal constituents already built recursively out of them by *Merge* and/or *Move*.

(10) $\gamma = \{\delta, \{\alpha, \beta\}\} = \{H(\alpha), \{\alpha, \beta\}\}$ if α projects, or
$\{H(\beta), \{\alpha, \beta\}\}$ if β projects

α β

Adjunct(ion) structures are distinguished in terms of the label of the set: the label is taken to be an ordered-pair of the same head <H, H>.

(11) $\gamma = \{\delta, \{\alpha, \beta\}\} = \{<H(\alpha), H(\alpha)>, \{\alpha, \beta\}\}$ if β adjoins to α, or
$\{<H(\beta), H(\beta)>, \{\alpha, \beta\}\}$ if α adjoins to β

α β

Here, α or β is the host of the adjunct(ion), and presumably retains its original label as α or β; only the newly created node γ obtains the label of an ordered-pair of the same head, <H(α), H(α)> or <H(β), H(β)>.

Nevertheless, the relational read-out of the projection levels as [± maximal, ± minimal] does not quite capture the maximality of each 'segment' in such adjunct(ion) structures. Hornstein & Nunes (2008) demonstrate the conservation of the maximality of the host of the adjunct(ion).

(12) a. John could [eat the cake] and [eat the cake] he did.
b. John could [[eat the cake][in the yard]] and [eat the cake] he did [in the yard].

(13) a. … and [[eat the cake][in the yard]] he did [with a fork].
b. … and [[[eat the cake][in the yard]][with a fork]] he did.

Presumably, PP [in the yard] is adjoined to VP [eat the cake] in (12b) so that [[eat the cake][in the yard]] constitutes (the upper 'segment' of) VP, and yet (the lower 'segment' of) VP [eat the cake] can undergo VP-Preposing, leaving the upper 'segment' of VP with the remnant PP. That is, the lower 'segment,' the host VP of the adjunction of PP, retains its maximality. Yet, as can been seen in (13), the entire VP, the upper(most) 'segment' of VP, can undergo VP-

Preposing as well. Such maximality of each 'segment' can be observed in other grammatical phenomena, such as VP-ellipsis, *do-so* anaphora, and *one*-substitution in NP. This state of affairs demands more explicit elaborations in (9), as to what kind of 'projections' are "invisible at the interface and for computation."

Decomposing *Move* into *Merge* + *Agree* (+ *Generalized Pied-Piping*) and discerning two subtypes of *Merge*, Chomsky (2000) reinterpreted 'substitution' structures as ones produced by Set-Merge and 'adjunct(ion)' structures by Pair-Merge, and suggested elimination of labels since they are predictable in BPS representations (*cf*. Collins 2002). Directly encoding the difference in their projection statuses into their respective derived structures, 'substitution' by Set-Merge as a symmetric operation yields a binary set whereas 'adjunct(ion)' by Pair-Merge as an asymmetric operation yields an ordered-pair.[3]

(14) a. Set-Merge(α, β) = {α, β}
b. Pair-Merge(α, β) = <α, β>

Reconceptualizing *Merge* as a free operation for C_{HL} and converting to the view that the property of 'movement/displacement' need not be triggered either, Chomsky (2004) identified two types of *Merge*: External Merge (EM) and Internal Merge (IM). EM corresponds to 'base-generation' and IM to 'movement/displacement' in the traditional sense. Together with the distinction in (14) above, *Merge* is cross-classified into four subclasses: External Set-Merge, Internal Set-Merge, External Pair-Merge, and Internal Pair-Merge, roughly corresponding to 'base-generation' of argument structures, 'substitution' movement, 'base-generated' adjuncts, and 'adjunction' movement, respectively (*cf*. Richards 2009).

Maintaining the assumption that *Merge* operates on two syntactic objects,

---

[3] Labeling Algorithm was later proposed in Chomsky (2013, 2015) to determine the category/headedness information, independent of structure-building operations.

its cross-classification has prompted proliferations of various strains of *Merge* in the literature: Interarboreal Movement (Bobaljik 1995, Bobaljik & Brown 1997, *inter alia*), Remerge (Bobaljik 1995, Epstein, *et al*. 1998, *inter alia*), Tucking-in Movement (Richards 1997, *et seqq*., *inter alia*), Late Merge (Bošković & Lasnik 1999, Fox 1999, *inter alia*), Sideward Movement (Nunes 1995, *et seqq*., *inter alia*), Parallel Merge (Citko 2005, *inter alia*), Grafting (van Riemsdijk 2006, *inter alia*), and Multidominance by External Remerge (de Vries 2009, *inter alia*), among many others.

2.3. Up to the Minute

Dissatisfied with the disarrayed propagation of unprincipled variants of *Merge*, Chomsky (2019a, *et seqq*.) reconceives it as MERGE that operates on the workspace (WS), in which *Merge* has previously been thought to be operating on two syntactic objects. MERGE maps the current workspace (WS) to a new workspace (WS´) at the next stage of the derivation, replacing two syntactic objects in WS with a binary set of them in WS´. Chomsky's (2019b) informal rendition is as follows:

(15) MERGE(P, Q, WS) = [{P, Q}, …] = WS´

Yet, in (15), MERGE looks like a ternary operation that takes P and Q as its direct operands, along with WS. Thus, let us reformulate it as below, sharpening its intention.

(16) Let a workspace WS be a set of syntactic objects SOs, where SOs are either lexical items LIs selected from the lexicon, or sets formed by MERGE.

(17) X is a term of Y iff X $\neq$ Y, X $\in$ Y, or X $\in$ Z, Z a term of Y.

(18) For any accessible terms P and Q, $P \neq Q$ in WS,

MERGE(WS) = WS´, where

a. $WS = [SO_1, SO_2, \dots, SO_n]$

b. $WS´ = [\{P, Q\}, SO_1, SO_2, \dots, SO_{n-1/2}]$

c. $((R \in WS) \wedge (R \notin \{P, Q\})) \rightarrow (R \in WS´)$

Terms are recursively defined SOs in (17), and the accessibility is determined by Minimal Search (MS), in which any SOs c-commanded by their copies are not accessible, and the Phase Impenetrability Condition is respected. By (18), $P \neq Q$, so that self-merger as {P, P} or {Q, Q} is excluded,[4] and by (18b) if the cardinality of WS´ is equal to WS, either P or Q is not a member of WS and one is a term of the other, instantiating Internal MERGE; if the cardinality of WS´ is smaller than WS by 1, both P and Q are members of WS, instantiating External MERGE. As the set {P, Q} is newly created in WS´, the cardinality of WS´ remains the same as the one of WS (Internal MERGE) or reduced only by 1 (External MERGE). That is, the cardinality of the workspace is monotone-decreasing (monotone non-increasing) whereas the number of accessible terms is strictly monotone-increasing by 1, adding the newly created set {P, Q}.[5] By (18c), any SOs other than P or Q remain in WS´, ensuring recoverability.

In addition to MERGE, which is a symmetric binary set-formation operation, Chomsky (*op. cit.*) discusses the necessity for an asymmetric analog for adjunct(ion), Pair-Merge. If we build on (Set-)MERGE, it would look something like the following:

---

4 If self-merger is allowed, {P, P} = {P}, recursively {{P}, {P}} = {{P}}, {{{P}}, {{P}}} = {{{P}}}, and so on, yielding an equivalent of non-branching projection in X´-Theory. This is reminiscent of Zermelo's set-theoretic construction of ordinal numbers. *Cf.* von Neumann (1928).

5 P and Q in WS are accessible by definition, and their copies are rendered inaccessible or removed from WS´.

(19) For any accessible terms P and Q, $P \neq Q$ in WS,

Pair-MERGE(WS) = WS´, where

a. $WS = [SO_1, SO_2, \ldots, SO_n]$

b. $WS' = [<P, Q>, SO_1, SO_2, \ldots, SO_{n-1/2}]$

c. $((R \in WS) \wedge (R \notin \{P, Q\})) \rightarrow (R \in WS')$

Instead of the set {P, Q}, the pair <P, Q> is created in WS´, and if both P and Q are members of WS, it is External Pair-MERGE, instantiating a 'base-generated' adjunct; if either P or Q is not a member of WS and one is a term of the other, it is Internal Pair-MERGE, instantiating 'adjunction' movement. As the newly created ordered-pair <P, Q> should be able to undergo further manipulation by (Pair/Set-)MERGE, it must also be counted as an SO, and hence a term, to be included in (16) and (17).

Attributing to Hisa Kitahara's proposal for head-to-head adjunction as Interarboreal Movement *à la* Bobaljik (1995), Bobaljik & Brown (1997), *inter alia*, Chomsky (*op. cit.*) suggests that both P and Q in <P, Q> are rendered inaccessible so that the number of accessible terms remains the same in WS´ and the problem of indeterminacy does not arise.

But such an inaccessibility, if any, seems to be restricted to head-to-head adjunction, since for phrasal adjunct(ion), the host of the adjunct(ion) can retain its (maximality and hence) accessibility, as we have seen in Hornstein & Nunes' (*op. cit.*) demonstration (12-13). Furthermore, the adjunct itself is accessible so that it is subject to movement/displacement.

(20) a. How$_i$ did John $[_{VP}$ $[_{VP}$ fix the car] $\boldsymbol{t}_i$]?

b. [Without any professional tools]$_i$, John $[_{VP}$ $[_{VP}$ fixed the car] $\boldsymbol{t}_i$].

What needs to be blocked is extraction out of an adjunct, known as the Adjunct (Island) Condition, subsumed under Huang's (*op. cit.*) CED.

(21) a. *Who$_i$ did [$_{IP}$ [$_{IP}$ you have lunch] [$_{CP}$ after you met $\boldsymbol{t}_i$]]?[6]

b. *Mt. Fuji$_i$, I used to [$_{VP}$ [$_{VP}$ live] [$_{PP}$ near $\boldsymbol{t}_i$]].[7]

It remains to be seen whether the Adjunct (Island) Condition is reducible to the inaccessibility of the constituents inside the adjunct.[8] Even in the head-to-head adjunction cases, it is not clear why the both coordinates of an ordered-pair become inaccessible.[9]

3. Sequence

Chomsky (*op. cit.*) brings out Pair-MERGE in the discussion of unbounded unstructured coordination, which he argues requires a sequence and its limiting case is a pair, produced by Pair-MERGE for adjunct(ion).

Chomsky (2021; 31ff.) proposes the operation FORMSEQUENCE, generalizing Merge to be the combinatorial *n*-ary set-formation operation (*ibid*.: 20f. [D]) and building on it.

(22) Merge(X_1, … , X_n, WS) = WS′ = {{X_1, … , X_n}, W, Y}, satisfying SMT (Strong Minimalist Thesis: TT) and LSCs (Language-Specific Conditions: TT).

---

6 Assuming the temporal clause is adjoined to IP.

7 Assuming the circumlocative PP is adjoined to VP.

8 Nakashima (to appear) makes an interesting attempt to account for the Adjunct (Island) Condition in terms of indeterminacy in the framework of MERGE. Hornstein & Nunes (*ibid*.: 77f., fn.23.) suggest that adjunct(ion) structures may lack labels (*cf*. Chametzky 1994) and the label-less constituents interfere with the path calculation, blocking movement out of the adjunct(ion) structures.

9 Most likely, it has to do with some version of lexical integrity, but it is difficult to grapple with, as in the recent adoption of versions of Distributed Morphology (Halle & Marantz 1993, *inter alia*) and Exo-Skeletal syntax (Borer 2003, *inter alia*) into Minimalism (*cf*. Den Dikken 2002, Bruining 2018, *inter alia*). For lexical integrity, see Siegel (1974), Di Sciullo & Williams (1987), Roberts (1991), Lieber (1992), Bresnan & Mchombo (1995), Ackema & Neeleman (2002), Lieber & Scalise (2005), Booij (2008), Haspelmath & Sims (2010), among many others.

MERGE is just the limiting case of n=2 and Y is null.

(23) FORMSEQUENCE($\{X_1, \ldots, X_n\}$, WS) = WS´
$= \{<(\text{Conj},)\ X_1, \ldots, X_n>, W, Y\}$

Taking stock, there are two general n-ary structure-building operations, (Generalized set-formation) Merge and FORMSEQUENCE, and (Set-)MERGE and Pair-MERGE are their respective limiting cases of binary applications.[10] This state of affairs does not seem, at least to me, to be the optimal scenario for SMT, but a recipe for potential mutations. I think that the best scenario would be just one single structure-building operation for C_{HL}.

4. Outlook

As we have been seeing from the outset, the argument/adjunct distinction is not so clear-cut, and since the abandonment of X´-schemata, 'adjunct(ion)' in a sense had been unwittingly insinuated into any incremental structure-building operation as 'node-creation.'

Yet, intuitively, there seems to be a need for some kind of distinction to be drawn, fuzzy though it may be. The question is where such a distinction emanates from. My hunch is that the distinction stems from the difference in the structures themselves, not from the operations that build such structures or from any special marking on representation of such structures.

In fact, there is an ingenious proposal of reducing Pair-Merge to Set-Merge in Omune (2018a, b, 2019), employing set-theoretic reduction of an ordered-pair to a set. Exploiting Kuratowski's (1921) short definition of an ordered-pair as a set:

[10] Chomsky (*ibid.*: 33f., fn.51) claims that the operation FORMSEQUENCE can apply non-cyclically. If so, Pair-MERGE, as its limiting case, should also be able to apply non-cyclically, which, if so, raises the same problem as Late Merge, which MERGE is supposed to solve in the first place.

(24) $<\alpha, \beta> := \{\alpha, \{\alpha, \beta\}\}$

Omune reformulates Pair-Merge(α, β) as Set-Merge(α, $\{\alpha, \beta\}$), yielding $\{\alpha, \{\alpha, \beta\}\} = <\alpha, \beta>$. That is, to obtain the result of Pair-Merge(α, β), Set-Merge operations have been executed twice in succession: Set-Merge(α, β) $= \{\alpha, \beta\}$ immediately followed by Internal Set-Merge(α, $\{\alpha, \beta\}$), yielding $\{\alpha, \{\alpha, \beta\}\}$. This two-step process may be depicted as follows:

(25) Pair-Merge(α, β) $\Leftrightarrow$ Set-Merge(α, Set-Merge(α, β))

Or it may better be thought that Set-Merge does not have to apply successively in such a fashion, but has the option of applying in such a way. If it happens to apply this way, the resulting structure is *interpreted* as an 'adjunct(ion)' structure; otherwise, the structure is not an 'adjunct(ion)' structure. This is more befitting to the strongly Markovian nature of the system envisaged. And this idea can be straightforwardly carried over to Pair-MERGE, built on the two-step successive immediate application of Set-MERGE.[11]

Mathematically, a sequence is a totally ordered multiset (potentially containing multiple instances of each member[12]), and it can be reduced to a nested ordered-pair:

(26) $<\alpha, \beta, \gamma, \alpha, \gamma, \delta, \ldots> = <\alpha, <\beta, <\gamma, <\alpha, <\gamma, <\delta, \ldots>>>>>>$

and thus further reducible to a nested set, for instance, in Kuratowski's short definition as follows:

(27) $= \{\alpha,\{\alpha, \{\beta, \{\beta, \{\gamma, \{\gamma, \{\alpha, \{\alpha, \{\gamma, \{\gamma, \{\delta, \{\delta, \ldots\}\}\}\}\}\}\}\}\}\}\}\}$

Goodman (1941) raised a concern about such reduction with the progres-

[11] *Cf*. Fong's (2022) 36ff., *e.g*. (51-61) for the discussion of problems with repetitive operations.

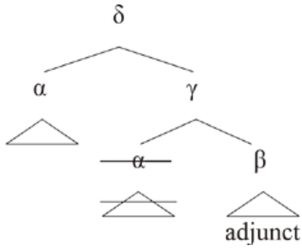
[12] *Cf*. Chomsky (*op. cit.*) 31f. fn.48.

sive raising of logical types as the length of the sequence increases and offered a solution for it. However, for syntactic structures, it does not seem to matter; a phrase is a higher type than a word and a clause is a higher type than a phrase. In BPS, a complete sentence is a nested set after all. Iterated applications of (Set-)MERGE recursively produces nested sets, part of which might be interpreted as an ordered-pair or a sequence.

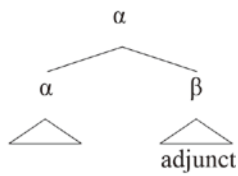
A potential problem may emerge from the set representation of the adjunct(ion) structure {α, {α, β}} as in (24); 'movement' of α is extremely local whether by 'substitution' or 'adjunction,' and it is the proto-typical configuration where some kind of anti-locality conditions may be invoked.[13] Yet, the entire structure must be of the category α, with β being the adjunct.

The two α's may appear reminiscent of the two 'segments' of the single category projection of α, but the appearance is misleading: the upper α of the set {α, {α, β}} is a copy of the lower α of the contained set {α, β} and the former c-commands the latter. If we were to represent it in an X´-tree diagram, it would look like (28a) whereas the X´-theoretic 'adjunct(ion)' structure would be something like (28b):

(28) a.



b.



In (28a = 24), the lower ~~α~~ of the contained set {α, β}= γ is inaccessible by MS, but the entire set {α, {α, β}} = δ (= the upper 'segment' of α in (28b)), its upper α, which is an identical copy of the original host of 'adjunc(tion)' ~~α~~ (=

[13] For phenomena of anti-locality, see Abels (2003), Grohmann (2011), and reference cited therein, among others.

the lower ‘segment’ of α in (28b)), and the adjunct β itself are accessible, accounting for Hornstein & Nunes’ (*op. cit.*) observation (12-13) and the fact that the adjunct β itself can undergo ‘movement/displacement’ (20).[14]

The structural representation (28a) brings to mind Larson’s (1988, 1990) rightward downward branching analysis,[15] where elements on the right are generally lower in the phrase structure, which is taken up in Kaynes’ (*op. cit.*) theory of antisymmetry. Yet, in BPS, no linear order is entailed in the structure, and it is not clear how it makes out with premodifiers.

It remains to be seen how the Adjunct (Island) Condition and the Coordinate Structure Constraint can be accounted for, and whether other empirical phenomena vindicated by Pair-Merge or FormSequence can be analyzed in terms of nested set structures produced by the single structure-building operation (Set-)MERGE alone.

References

Abels, Klaus (2003) *Successive Cyclicity, Anti-Locality, and Adoposition Stranding*. Ph.D. dissertation, University of Connecticut, Stors.

Ackema, Peter & Ad Neeleman (2002) “Syntactic Atomicity.” *Journal of Comparative Germanic Linguistics* **6**: 93-128.

Bach, Emmon (1974) *Syntactic Theory*. New York: Holt, Rinehart & Winston.

Bobaljik, Jonathan David (1995) “In Terms of Merge: Copy and Head Movement.” *MIT Working Papers in Linguistics* **27**: 41-64.

Bobaljik, Jonathan David & Samuel Brown (1997) “Interarboreal Operations: Head Movement and the Extension Requirement.” *Linguistic Inquiry* **28**: 345-356.

Borer, Hagit (2003) “Exo-Skeletal vs. Endo-Skeletal Explanations: Syntactic Projections and the Lexicon,” pp. 31-67 in John Moore & Maria Polinsky (eds.) *The Nature of Explanation in Linguistic Theory*. Stanford CA: CSLI Publications.

Bošković, Žejliko & Howard Lasnik (1999) “How Strict Is Cycle?” *Linguistic Inquiry*

---

[14] Omune (2019) argues that adjuncts (β in 28a, b) are inaccessible and in the cases where adjuncts have moved as in (20), they are in fact complements.

[15] Larson (1988: 345ff. fn.11) notes that in his rightward downward branching analysis, adverbs are not the outermost adjuncts of verbs but rather its innermost complements, assuming the semantic analysis of adverbs by McConnell-Ginet (1982). See fn.14 above.

30: 691-703.

Booij, Geert (2008) "Lexical Integrity as a Formal Universal: A Constructionist View," pp. 83-100 in Sergio Scalise, Elisabeth Magni & Antonietta Bisetto (eds.) *Universals of Language Today*. Rotterdam: Springer.

Bresnan, Joan & Sam M. Mchombo (1995) "The Lexical Integrity Principle: Evidence from Bantu." *Natural Language and Linguistic Theory* **13**: 181-254.

Bruening, Benjamin (2018) "The Lexicalist Hypothesis: Both Wrong and Superfluous." *Language* **94**: 1-42.

Chametzky, Robert (1994) "Chomsky-Adjunction." *Lingua* **93**: 245-264.

Chomsky, Noam (1955) *The Logical Structure of Linguistic Theory*. Ms. Harvard University. Published in part in 1975, New York: Plenum.

Chomsky, Noam (1957) *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam (1993) "A Minimalist Program for Linguistic Theory," pp. 1-52 in Kenneth Hale and Samuel J. Keyser (eds.) *The View from Building 20*: *Essays in Linguistics in Honor of Sylvain Bromberger.* Cambridge, MA: MIT Press.

Chomsky, Noam (1986) *Barriers*. Cambridge, MA: MIT Press.

Chomsky, Noam (1995a) "Categories and Transformation," pp. 219-394 in Noam Chomsky, *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam (1995b) "Bare Phrase Structure," pp. 51-109 in Héctor Campos & Paula Kempchinsky (eds.) *Evolution and Revolution in Linguistic Theory*. Washington D.C.: Georgetown University Press.

Chomsky, Noam (2000) "Minimalist Inquiries: The Framework," pp. 89-155 in Roger Martin, *et al.* (eds.) *Step by Step*: *Essays on Minimalist Syntax in Honor of Howard Lasnik*. Cambridge, MA: MIT Press.

Chomsky, Noam (2004) "Beyond Explanatory Adequacy," pp. 104-131 in Adriana Belletti (ed.) *Structures and Beyond*: *The Cartography of Syntactic Structures*, *Volume* **3**. Oxford: Oxford University Press.

Chomsky, Noam (2013) "Problems of Projection." *Lingua* **130**: 33-49.

Chomsky, Noam (2015) "Problems of Projection: Extensions," pp. 3-16 in Elisa Di Domenico, Cornelia Hamann & Simona Matteini (eds.) *Structures, Strategies and Beyond*: *Studies in Honour of Adriana Belletti*. Amsterdam: John Benjamins.

Chomsky, Noam (2019a) "Some puzzling foundational issues: The Reading program." *Catalan Journal of Linguistics*. *Special Issue, Generative Syntax*: *Questions, Crossroads, and Challenges* **2019**: 263-285.

Chomsky, Noam (2019b) UCLA lectures. Invited lectures at the Department of Linguistics, University of California, Los Angeles, April 29-May 2, 2019. https://linguistics.ucla.edu/noam-chomsky/ [accessed June 2021]. (manuscript available with some changes and an introduction by Robert Freidin. https://ling.auf.net/lingbuzz/005485 [accessed June 2021].)

Chomsky, Noam (2021) "Minimalism: Where We Are Now, and Where Can We Hope to

Go." *Gengo Kenkyu* **160**: 1-41.

Citko, Barbara (2005) "On the Nature of Merge: External Merge, Internal Merge, and Parallel Merge." *Linguistic Inquiry* **36**: 475-496.

Collins, Chris (2002) "Eliminating Labels," pp. 42-64 in Samuel David Epstein & T. Daniel Seely (eds.) *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell.

Den Dikken, Marcel (2002) "Lexical Integrity, Checking, and the Mirror: A Checking Approach to Syntactic Word Formation." *Journal of Comparative Germanic Linguistics* **6**: 169-225.

Di Sciullo, Anna-Maria & Edwin Willimas (1987) *On the Definition of Word*. Cambridge, MA: MIT Press.

Epstein, Samuel D., *et al.* (1998) *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Fong, Sandiway (2022) "Simple Models: Computational and Linguistic Perspective." *Tohoku Gakuin University Journal of Institute for Research in English Language and Literature* **46**: 1-48. (This volume).

Fox, Danny (1999) "Reconstruction, Binding Theory, and the Interpretation of Chains." *Linguistic Inquiry* **30**: 157-196.

Goodman, Nelson (1941) "Sequences." *The Journal of Symbolic Logic* **6**: 150-153.

Grohman, Kleanthes K. (2011) "Anti-Locality: Too-Close Relations in Grammar," pp. 260-290 in Cedric Boeckx (ed.) *The Oxford Handbook of Linguistic Minimalism*. Oxford: Oxford University Press.

Halle, Morris & Alec Marantz (1993) "Distributed Morphology and the Pieces of Inflection," pp. 111-176 in Kenneth Hale & Samuel J. Keyser (eds.) *The View from Building 20*: *Essays in Linguistics in Honor of Sylvain Bromberger.* Cambridge, MA: MIT Press.

Haspelmath, Martin & Andrea D. Sims (2010) *Understanding Morphology*. London: Hodder Education.

Hornstein, Norbert & Jairo Nunes (2008) "Adjunction, Labeling, and Bare Phrase Structure." *Biolinguistics* **2**: 57-86.

Huang, C.-T. James (1982) *Logical Relations in Chinese and the Theory of Grammar.* Ph.D. dissertation, MIT.

Kayne, Richard (1994) *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.

Kuratowski, Casimir (1921) "Sur la notion de l'ordre dans la Théorie des Ensembles." *Fundamenta Mathematicae* **2**: 161-171.

Kuroda, Shige-Yuki (1988) "Whether We Agree or Not: A Comparative Syntax of English and Japanese." *Lingvisticæ Investigationes* **12**: 1-47.

Larson, Richard K. (1988) "On the Double Object Construction." *Linguistic Inquiry* **19**: 335-391.

Larson, Richard K. (1990) "Double Objects Revisited: Reply to Jackendoff." *Linguistic*

Inquiry **21**: 589-632.

Lieber, Rochelle (1992) *Deconstructing Morphology*. Chicago: University of Chicago Press.

Lieber, Rochelle & Sergio Scalise (2005) "The Lexical Integrity Hypothesis in a New Theoretical Universe." *Lingue e Linguaggio* **5**: 7-32.

May, Robert (1985) Logical Form. Cambridge, MA: MIT Press.

McConnell-Ginet, Sally (1982) "Adverbs and Logical Form: A Linguistically Realistic Theory." *Language* **58**: 144-184.

Muysken, Pieter (1982) "Parametrizing the Notion "Head"." *Journal of Linguistic Researc*h **2**: 57-75.

Nakashima, Takanori (to appear) "How to Generate Adjuncts by MERGE." *Proceedings of the North East Linguistic Society* **52**.

von Neumann, John (1928) "Über die Definition durch transfinite Induktion und verwandte Fragen der allgemeinen Mengenlehre." *Mathematische Annalen* **99**: 373-391.

Nunes, Jairo (1995) *The Copy Theory of Movement and Linearization of Chains in the Minimalist Program.* Ph.D. dissertation, University of Maryland, College Park.

Nunes, Jairo (2001) "Sideward Movement." *Linguistic Inquiry* **31**: 303-344.

Omune, Jun (2018a) "Reformulating Pair-Merge of Heads." *English Linguistics* **34**: 266-301.

Omune, Jun (2018b) *Reformulating Pair-Merge, Inheritance and Valuation*. Ph.D. dissertation, Kansai Gaidai University.

Omune, Jun (2019) "Immediate-Local MERGE as Pair-Merge," *Coyote Papers* **22**: 12-21.

Richards, Marc (2009) "Internal Pair-Merge: the Missing Mode of Movement." *Catalan Journal of Linguistics* **8**: 55-73.

Richards, Norvin (1997) *What Moves Where When in Which Language?* Ph.D. dissertation, MIT.

Richards, Norvin (1998) "Shortest Moves to (Anti-)Superiority." *Proceedings of the West Coast Conference on Formal Linguistics* **XVI**: 335-349.

van Riemsdijk, Henk (2006) "Free Relatives," pp. 338-382 in Martin Everaert & Henk van Riemsdijk (eds.) *The Blackwell Companion to Syntax*, vol. II. Oxford: Blackwell.

Roberts, Ian (1991) "Excorporation and minimality." *Linguistic Inquiry* **22**: 209-218.

Ross, John R. (1967) *Constraints on Variables in Syntax.* Ph.D. dissertation, MIT.

Siegel, Dorothy (1974) *Topics in English Morphology*. New York: Garland Publishing.

Speas, Margaret (1990) *Phrase Structure in Natural Language*. Dordrecht: Kluwer Academic Publishers.

de Vries, Mark (2009) "On Multidominance and Linearization." *Biolinguistics* **3**: 344-403.

令和３年度　研究所行事報告

◇定時総会

5 月 27 日（木）（オンライン開催）

◇学術講演会

10 月 30 日（土）13 : 00〜14 : 30
（オンライン開催）
Sandiway Fong 氏（Associate Professor; Department of Linguistics, University of Arizona）
“On Modeling Human Language : Computational and Linguistic Perspectives”

◇定例公開講演会

3 月 19 日（土）13 : 00〜14 : 30
（オンライン開催）
清水　　遥 氏 （本学文学部教育学科准教授）
「英語リーディングにおける推論生成―深い読みに関わる読解プロセス―」

『東北学院英学史年報』第 43 号（令和 4 年 3 月 2 日）

『英語英文学研究所紀要』第 46 号（令和 4 年 3 月 14 日）

東北学院大学英語英文学研究所紀要第 **45** 号

（2020 年 3 月）所載論文

東北学院大学英語英文学研究所規程

昭和32年4月1日制定第1号
改正
昭和54年5月24日
平成19年4月1日
平成28年10月5日改正第119号

（設置）

第 1 条　東北学院大学（以下「本学」という）に東北学院大学学則第65条第2号オの規定に基づき、東北学院大学英語英文学研究所（以下「本研究所」という。）を置く。

（目的）

第 2 条　本研究所は、英語英文学及び関連諸分野の研究を行い、その発展に寄与することを目的とする。

（事業）

第 3 条　本研究所は、前条の目的遂行のために次の事業を行う。

(1)　英語英文学、英語教育学、英学史の研究及び資料の蒐集、整理、所蔵図書の点検

(2)　関連学会との連絡、他大学の関連研究室との共同研究

(3)　定期刊行物の発行

(4)　講演会、公開講座等の開催

(5)　学生の関連領域における研究への指導、助言

(6)　その他本研究所の目的遂行に必要な事業

（組織）

第 4 条　本研究所は、次に掲げる者をもって組織する。

(1)　所長　1名

(2)　主事　1名

(3)　所員　本学の英語英文学、英語教育学及び関連分野の専任教員

(4)　客員　若干名

（所長）

第 5 条　所長は、学長が委嘱する。

2　所長の任期は 2 年とし、再任を妨げない。

3　所長は、本研究所を代表し、事業を統括する。

（主事）

第 6 条　主事は、所長が推薦し、学長が委嘱する。

2　主事の任期は 2 年とし、再任を妨げない。

3　主事は、本研究所の庶務一般をつかさどる。

（客員）

第 7 条　客員は、学外者で適任と認められる者を、委員会の議を経て主事が推薦し、所長が委嘱する。

（運営委員会）

第 8 条　本研究所に東北学院大学英語英文学研究所運営委員会（以下「委員会」という。）を置く。

2　委員会は、所員の互選による 4 名の委員及び主事をもって構成する。

3　委員の任期は 1 年とし、再任を妨げない。

4　委員会は、本研究所の運営について所長の諮問に応える。

5　委員会の議長は、主事をもって充てる。

（専門委員会）

第 9 条　本研究所は、事業の遂行のために専門委員会を設けることができる。

（経費）

第10条　本研究所の経費は、本学の補助金、基金、寄附金、事業収入等

をもって充てる。

（総会）

第11条　本研究所は、毎年1回定時総会を開き、必要に応じて臨時総会を開く。

2　総会は、本研究所に関する必要事項を審議する。

（改廃）

第12条　この規程の改廃は、総会の議を経て学長が行い、理事会の承認を得るものとする。

附　則

この規則の改正は、総会における3分の2以上の決議による。

附　則（昭和54年5月24日）

この規則は、昭和54年5月24日から施行する。

附　則（平成19年4月1日）

この規則は、平成19年4月1日から施行する。

附　則（平成28年10月5日改正第119号）

この規程は、平成28年10月5日から施行する。

英語英文学研究所「紀要」規程

I.　投稿資格

英語英文学研究所員及び英語英文学研究所委員会が特に依頼した者。

II.　投稿内規

募集原稿の種類：英米文学・英米文化、英語学、英語教育学に関する論文。未発表のもの（口頭発表及び口頭発表に基づくプロシーディングは除く）

原稿作成要領：1）和論文の場合、本文は横書きで 16,000 字程度。必ず英文概要（A4 1 枚程度）を添えること。

2）英論文の場合、本文は 6,000 語程度。

3）原則として *The MLA Handbook* に基づくこと。

4）注の番号は右上に 1, 2, 3, …と表示すること。尚、注は和文、英文ともに脚注にすること。

原稿の採否：英語英文学研究所内の審査機関を経て決定する。

原稿の締切日：毎年 10 月 20 日

（昭和 54 年 5 月 24 日より施行）
（平成元年 5 月 11 日改正）
（平成 9 年 7 月 3 日改正）
（平成 25 年 7 月 18 日改正）

東北学院大学英語英文学研究所（令和3年度）

所　長	教　授	豊	島	孝	之
主　事	教　授	福	士		航
委　員	教　授	下	館	和	巳
委　員	准教授	井	出	達	郎
委　員	准教授	岸		浩	介

編　集　後　記

『英語英文学研究所紀要』第 46 号をお届けします。今号には，英語学分野の論文が 2 本掲載されることとなりました。ご投稿に心より感謝申し上げます。

査読をお引き受け下さった先生方，編集作業に携わって下さった研究機関事務課の方々，お骨折りに感謝申し上げます。

今後も，所員の皆様の積極的なご投稿を心よりお待ち申し上げております。

（編集担当　福士）

令和 4 年 3 月 14 日 発行

東北学院大学英語英文学研究所紀要 第 46 号

編集発行人 豊 島 孝 之

発 行 所 東北学院大学英語英文学研究所
980-8511 仙台市青葉区土樋一丁目 3 番 1 号
TEL. 022(264)6401
FAX. 022(264)6530

印 刷 所 笹 氣 出 版 印 刷 株 式 会 社
984-0011 仙台市若林区六丁の目西町 8 番 45 号

Journal of Institute for Research in English Language and Literature

46

2022

Tohoku Gakuin University